

Multi-agent Integer Programming Games with Non-cooperation and Cooperation

Hyunwoo Lee

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Industrial and Systems Engineering

İ. Esra Büyüктаhtakın, Chair

Robert Hildebrand, Co-chair

Wenbo (Selina) Cai

Rohit Kannan

August 5, 2026

Blacksburg, Virginia

Keywords: Integer Programming Games, Pure Nash Equilibrium, Best-Response Dynamics,
Cutting-Plane Algorithms, Cooperative Games and the Core, Aquatic Invasive Species

Copyright 2026, Hyunwoo Lee

Multi-agent Integer Programming Games with Non-cooperation and Cooperation

Hyunwoo Lee

(ABSTRACT)

This dissertation studies *integer programming games* (IPGs), in which each of several self-interested players solves an integer program whose payoff is coupled to the others' discrete decisions. Because feasible sets are combinatorial and best responses require integer optimization, even deciding whether a *pure Nash equilibrium* (PNE) exists is computationally intractable in general. The unifying thesis is that the equilibrium of such a game is a *computable and leverageable* object around which better system-level decisions can be built, and this idea is developed across the non-cooperative, generalized, and cooperative regimes on a common branch-and-cut foundation of globally valid inequalities and best-response oracles. First, to make equilibrium computation scale, *round-based random-restart best-response dynamics* (RRR-BRD) are introduced with a Las Vegas convergence guarantee and hybridized with the zero-regret cutting-plane method (BZR), lifting tractable PNE computation and sampling from a handful of players to dozens; the methods are instantiated on *edge-weighted budgeted maximum coverage* games, including a Minnesota aquatic invasive species (AIS) inspection game with up to 84 counties. Second, *Nash-informed planning* converts computed equilibria into individual-rationality floors and optimizes social welfare above them, improving system outcomes without making any player worse off; the accompanying *Price of Individual Rationality* is proven to be bounded by the Price of Stability, and the welfare cost of guaranteeing participation is empirically negligible. Third, for generalized IPGs with shared constraints, *conditional equilibrium inequalities* and a generalized zero-regret algorithm optimize over the equilibrium set, while α -*approximate* equilibrium certificates and structural (VEST) cuts handle cost-sharing games such as integer splittable bin packing.

Fourth, *cooperative* IPGs define coalition values through pooled integer programs and compute optimal, Core-stable coalition structures via stability inequalities and separation. Computational studies, anchored by the AIS application, demonstrate scalability, interpretability, and concrete policy improvements.

Multi-agent Integer Programming Games with Non-cooperation and Cooperation

Hyunwoo Lee

(GENERAL AUDIENCE ABSTRACT)

Many important public decisions are made by many separate groups, each with its own limited budget and a set of concrete, all-or-nothing choices—such as which sites to staff with inspectors. When each group plans on its own, the outcome for the whole region can fall well short of what could be achieved together, yet no central authority can simply order everyone what to do. This dissertation develops mathematical tools for such *discrete*, multi-decision-maker problems. It first shows how to compute *stable* plans—arrangements in which no group can do better by changing its own choices alone—for problems far larger than previous methods could handle. It then uses those stable plans as a starting point to design region-wide policies that improve the overall result while guaranteeing that no group ends up worse off than before, and with no money changing hands. It further addresses settings where groups must share limited resources, and settings where groups can cooperate by pooling budgets, showing which partnerships are worth forming and how to keep them stable. The methods are demonstrated on the real problem of slowing the spread of aquatic invasive species across Minnesota’s lakes, where independent counties run watercraft inspections: the approach coordinates their efforts to protect substantially more at-risk lakes without any county sacrificing its own outcome. The same ideas apply broadly to problems such as facility siting, emergency-response planning, and shared infrastructure investment.

Dedication

To my wife Juhae, and to my parents.

Acknowledgments

I have been extraordinarily fortunate to be guided by two advisors, İ. Esra Büyüктаhtakın and Robert Hildebrand, who invested in me generously and without reserve, and I am deeply grateful to both. Professor Büyüктаhtakın has a gift for identifying problems that matter—real societal and operational challenges—and for framing them so that rigorous methods can make a genuine difference. Her constant encouragement and steadfast support carried me through the most difficult stretches of this journey. Professor Hildebrand shaped me as a researcher through his deep expertise in the theory and algorithms of discrete optimization. He was never a manager but a true mentor, patiently and generously raising the level of my thinking through hands-on, thoughtful guidance and a sincere care for my growth. I have always been indebted to his sharp mathematical instincts; I cannot hope to match the depth of his insight, but I will keep striving toward it. To both of them I owe the questions I have learned to ask and the researcher I have become.

I would also like to thank the members of my committee. I thank Professor Wenbo (Selina) Cai for bringing compelling environmental problems into a game-theoretic modeling framework, which helped motivate the direction of my research, and Professor Rohit Kannan for his careful reading, thoughtful questions, and constructive feedback, which improved this work in many ways.

I have been fortunate to work with wonderful collaborators, and I thank Bainian Hao, Carla Michini, and Jian Yang for the insight, effort, and many discussions that shaped the results in this dissertation. I also gratefully acknowledge the Minnesota Aquatic Invasive Species Research Center (MAISRC) for its financial support and for providing the data that made the case studies concrete.

I owe special thanks to Professor Taesu Cheong and Professor Young Woong Park, who first opened the path for me to study abroad. Beyond that initial opportunity, they have continued to offer me sincere and thoughtful advice on my studies, my research, and my life to this day, and I am deeply grateful for their enduring mentorship.

My heartfelt thanks go to the Korean Baptist Church of Blacksburg for welcoming me as one of their own and for supporting me as I served the young adult group, the Peniel Choir, and the campus ministry. I also thank the many Korean friends in the Grado Department of Industrial and Systems Engineering at Virginia Tech, whose support meant a great deal to me, and my labmates in the SysOptiMaL Lab for the many enjoyable and constructive discussions we shared.

Finally, my deepest gratitude belongs to my family. To my parents, whose generous and unwavering love and support—their constant help and encouragement—made it possible for me to pursue my studies this far: thank you, with all my heart. And to my wife, Juhae: we began our marriage as newlyweds far from home, and you faced the hardships of life abroad together with me, thinking through every difficulty and enduring them at my side. You have been my most trusted advisor and counselor and my most patient listener, and your steady emotional support carried me throughout. A significant part of this degree belongs to you.

Contents

List of Figures	xvii
List of Tables	xix
1 Introduction	1
2 Random-Restart Best-Response Dynamics for Large-Scale Integer Programming Games and Their Applications	9
2.1 Introduction	10
2.1.1 Literature Review	12
2.1.2 Contribution	14
2.2 Preliminaries	17
2.2.1 Integer Programming Games and Pure Nash Equilibria	17
2.2.2 Normal-Form Games, BR State Graphs, and Markov Reduction	19
2.3 Convergence of Best-Response Dynamics in Finite Normal-Form Games . . .	20
2.3.1 SCCs, the Condensation DAG, and Recurrent Classes	21
2.3.2 Deterministic Convergence in Structured Subclasses	22
2.3.3 Round Random-Restart Best-Response Dynamics (RRR-BRD)	23
2.3.4 Complexity Analysis for RRR-BRD	28

2.4	Best-response Dynamics Incorporated Zero-regret Algorithm	30
2.4.1	Lifted Space and Perfect Equilibrium Formulation	31
2.4.2	Algorithm Details	33
2.4.3	RRR-BRD as a Randomized Local-Search Subroutine for Other Al- gorithms	36
2.4.4	Approximate PNE Certificates upon Early Termination	39
2.5	EBMC Game Models	39
2.5.1	The Edge-weighted Budgeted Maximum Coverage (EBMC) Problem	40
2.5.2	A Non-cooperative EBMC Game Among N Players	41
2.5.3	Types and Network Structure in EBMC	46
2.5.4	A Resource Allocation Game among Minnesota Counties	46
2.6	Pure Nash Equilibrium for EBMC Games	48
2.6.1	Existence of a PNE in the Locally Altruistic Games	49
2.6.2	Existence of a PNE in the Selfish Games	49
2.7	Computational Results	51
2.7.1	Computational Setup and Metrics	52
2.7.2	EBMC Games with Random Dataset	54
2.7.3	EBMC Games with Minnesota Dataset	56
2.7.4	Large-scale Knapsack Problem Games Datasets and Results	59
2.8	Conclusion	63

Appendix	64
2.A Acronyms and Notations	65
2.B Proofs	68
2.C Random Feasible-Start Generators for KPG and EBMC	72
2.D Extremely Small $q_\mu[R]$	76
2.E Full Simulation Results	79
2.F Example for the BZR Algorithm	82
2.G Exponential Separation Between ZR and BZR	84
2.H Illustration: A Two-Player Selfish EBMC Game	85
2.I EBMC as an IPG: Reformulation and Complexity Implications	87
2.J Counterexamples for the Existence of PNE	89
2.K RRR-BRD Results	91
2.L BZR Full Results	96
2.M Approximate-PNE	98
2.N County-level Utility Values	100
3 Nash-informed Planning for Public-Resource Coordination: An Application to Aquatic Invasive Species Prevention	101
3.1 Introduction	101
3.1.1 A new direction of analysis	102
3.1.2 Related literature	104

3.1.3	Contributions and plan	106
3.2	Nash-informed Planning Framework	108
3.2.1	General IPG setting and notation	109
3.2.2	Equilibrium analysis phase	110
3.2.3	Construction of reference utility values	111
3.2.4	Bounded social welfare (BSW) optimization	113
3.2.5	Structural guarantees	115
3.3	EBMC Games and Application Models	117
3.3.1	Operational context and modeling assumptions	117
3.3.2	EBMC formulation	118
3.3.3	Three Nash-informed optimization models for EBMC	121
3.3.4	Instantiation	122
3.3.5	Policy-constrained BC variants	123
3.4	Structural Sources of the Realized PIR	123
3.4.1	General IPG argument	124
3.4.2	EBMC arc structure and externality multiplier	125
3.5	Computational Study	128
3.5.1	Data and calibration	128
3.5.2	Computing the range of social welfare	130
3.5.3	Baseline results (Minnesota, 100% budget)	131

3.5.4	Tailored BC variants	133
3.5.5	Budget sensitivity and PIR across budget scenarios	134
3.6	Generalization Across IPG Classes	136
3.7	Managerial and Policy Implications	139
3.8	Conclusion	142
	Appendix	143
3.A	Acronyms and Notation	143
3.B	Proofs	143
3.C	Stylized comparison of BSW with correlated equilibria	146
3.D	Analytical bounds on PIR for EBMC games	148
3.D.1	Instance-parameterized bound	148
3.D.2	Degenerate single-player construction with unbounded PIR	150
3.E	Supplementary computational figures	152
3.F	EBMC random instances: detailed results	159
3.G	Knapsack Problem Game: detailed results	161
3.H	Kidney Exchange Game: detailed results	163
4	Conditional Equilibrium Inequalities for Generalized Nash Equilibrium Problems with Pure-Integer Variables	168
4.1	Introduction	168
4.1.1	Problem Setting	168

4.1.2	Literature Review	170
4.1.3	Contributions	172
4.1.4	Paper Organization	174
4.2	Conditional Equilibrium Inequalities	174
4.2.1	Lifted Formulation	179
4.2.2	CEI Validity and Closure	181
4.2.3	Generalized Zero-Regret Algorithm	184
4.3	VEST Inequalities	186
4.4	Complementary Solution Algorithms	191
4.4.1	Random-Restart Best-Response Dynamics for GNEPs	191
4.4.2	α -Bisection Search	193
4.5	Applications	195
4.5.1	Integer Splittable Bin Packing (ISBP)	195
4.5.2	Network Formation Games (NFG)	207
4.5.3	Generalized Knapsack Games (GKG and GKGB)	213
4.6	Computational Study	215
4.6.1	Experimental Design	215
4.6.2	ISBP: Integer Splittable Bin Packing Games	218
4.6.3	NFG: Network Formation Games	221
4.6.4	GKG: Generalized Knapsack Games	222

4.6.5	GKGB: Generalized Knapsack Games with Reciprocally Bilinear Co-efficients	224
4.6.6	Impact of VEST Cuts	227
4.7	Conclusion	230
	Appendix	230
4.A	Symmetry-Breaking Techniques for ISBP	231
4.B	Comprehensive Computational Results	236
5	Cooperative Integer Programming Games: Core Stability and Optimal Coalition Structures	244
5.1	Introduction	244
5.1.1	Literature Review	247
5.1.2	Contributions and Outline	249
5.2	Cooperative Integer Programming Games	251
5.2.1	A resource allocation game	252
5.2.2	Structural properties of the characteristic function V	255
5.3	A Mixed-Integer Formulation	257
5.3.1	OCS formulations: aggregated and disaggregated	257
5.3.2	Stability inequality for SCS	262
5.3.3	Lifted stability cuts	265
5.4	Computing an OCS	267

5.4.1	Separation strategies	267
5.4.2	Iterative re-solving versus lazy constraints	272
5.4.3	An SCS-feasible primal heuristic	272
5.4.4	Per-coalition solution concepts and payoff refinement	274
5.4.5	Algorithm summary	275
5.5	Experimental design and results	276
5.5.1	Does disaggregation pay?	277
5.5.2	Solution strategies for OSCS	281
5.5.3	Effect of the coalition policy	282
5.5.4	Cost of the framework	284
5.5.5	Implementation recommendations	285
5.6	Conclusion	286
	Appendix	286
5.A	Notation	286
5.B	Omitted Proofs and Examples	286
5.C	Extension to General MILP-Based CIPGs	291
5.D	Complexity of the Stability Problems	293
5.E	A Global Bound on the Objective	298
5.F	Additional Computational Results	299
5.G	Coalition Structure and Payoff Details	307

5.G.1 Per-coalition solution concepts: details	307
6 Conclusion	310
Bibliography	317

List of Figures

2.1	Illustration of a connected condensation DAG with two terminal sinks: a singleton <i>PNE sink</i> P and a multi-node <i>non-PNE sink</i> M . Boxes and circles are SCCs; internal dots depict the number of profiles each SCC contains (singletons vs. multi-node SCCs). Arrows are edges of the acyclic condensation graph. SCC labels (A, B, C, D, E, P, M) are arbitrary and used solely for illustration. In potential or BR-weakly acyclic games, non-PNE sinks do not occur; all sink SCCs are singleton PNEs, so the branch labeled M is absent.	21
2.2	Illustration of the BZR algorithm in a two-player game.	36
2.3	A condensation DAG showing that $q_\mu[R]$ can be extremely small.	76
2.4	The interactions in the selfish game with two players.	86
2.5	No PNE for $\mathcal{G}_{ \mathcal{L} =1}^{\text{Self}}$	89
2.6	No PNE for $\mathcal{G}_{ \mathcal{L} \geq 2}^{\text{Self}}$	90
3.1	The three steps of Nash-informed planning. The dashed box encloses the contribution of this chapter: the reference-utility construction (the second arrow) and the bounded social welfare optimization (the third box). The first two boxes assemble the game and compute its pure Nash equilibria (PNEs) using existing methods.	103
3.2	Degenerate EBMC instance used in the unbounded-PIR proof.	151

3.3	County-level utility improvements relative to RPNE under the tailored coalition models.	153
3.4	Coalition configurations across the tailored models.	154
3.5	Social welfare and selected infested lakes across models (Minnesota, 100% budget).	156
3.6	County-level utility improvements relative to RPNE under BSW and BC. . .	157
3.7	Budget reallocations across coalition models for counties with a high baseline budget per infested lake.	157
3.8	Social welfare and selected infested lakes across models under the four budget scenarios.	158
4.1	The joint feasible region of the running CEI example.	178
5.1	Availability and the coalition-dependent split of the item set. Circles are players, squares are items, and an edge means availability. Members of c are drawn in black, and shaded squares are the common items, reachable by two or more members. Item B is common in (b) but individual-specific in (c), and item C the other way round: the split depends on the coalition, not on the item. Note also that D is out of reach until player 3 joins.	253
5.2	Coalition sizes in the incumbent structure, and stability of the Shapley value by coalition size.	283

List of Tables

2.1	Normal-form reductions: empirical success within $R = 10$ rounds under uniform restarts.	30
2.2	KPGs: empirical success within $R = 20$ rounds under random restarts. . . .	30
2.3	Summary of algorithms for computing Nash equilibria in IPGs.	37
2.4	Acronyms and shorthands	53
2.5	Summarized RRR-BRD results with different initial strategy profiles for $\mathcal{G}_{ \mathcal{L} =1}^{\text{Self}}$ and $\mathcal{G}_{ \mathcal{L} \geq 2}^{\text{Self}}$	55
2.6	Summarized comparison of BZR and ZR for $\mathcal{G}_{ \mathcal{L} =1}^{\text{Self}}$ and $\mathcal{G}_{ \mathcal{L} \geq 2}^{\text{Self}}$	56
2.7	Comparison of BZR and ZR for selfish EBMC games $\mathcal{G}_{ \mathcal{L} =1}^{\text{Self}}$	57
2.8	RRR-BRD results for selfish EBMC games $\mathcal{G}_{ \mathcal{L} \geq 2}^{\text{Self}}$ in the Minnesota dataset.	58
2.9	Comparison of BZR and ZR for selfish EBMC games $\mathcal{G}_{ \mathcal{L} \geq 2}^{\text{Self}}$ in the Minnesota dataset.	58
2.10	County-level utility values using different strategy profiles.	59
2.11	Summarized RRR-BRD results with different initial strategy profiles for Type A, B, and C KPG.	61
2.12	Summarized comparison of BZR and ZR for Type A, B, and C KPG.	61
2.13	Comparison of BZR and ZR for Type B KPG.	62
2.14	List of acronyms used in this chapter and its appendix	65

2.15	Summary of notation used in this chapter and its appendix (core notation)	66
2.16	Summary of notation used in this chapter and its appendix (applications and experiments)	67
2.17	Full simulation results for normal-form reductions	80
2.18	Full simulation results for KPGs	81
2.19	Per-coordinate payoffs for the Prisoner's Dilemma component game in $\mathcal{P}_n$.	84
2.20	Worst-case complexity comparison on $\mathcal{P}_n$: ZR versus BZR.	85
2.21	Strategy Profiles and Selfish Deviations.	89
2.22	Strategy Profiles and Selfish Deviations.	90
2.23	RRR-BRD results with different initial strategy profiles for selfish EBMC games $\mathcal{G}_{ \mathcal{L} =1}^{\text{Self}}$.	91
2.24	RRR-BRD results with different initial strategy profiles for selfish EBMC games $\mathcal{G}_{ \mathcal{L} \geq 2}^{\text{Self}}$.	92
2.25	RRR-BRD results with different initial strategy profiles for Type A KPG.	93
2.26	RRR-BRD results with different initial strategy profiles for Type B KPG.	94
2.27	RRR-BRD results with different initial strategy profiles for Type C KPG.	95
2.28	Comparison of BZR and ZR for selfish EBMC games $\mathcal{G}_{ \mathcal{L} \geq 2}^{\text{Self}}$.	96
2.29	Comparison of BZR and ZR for Type A KPG.	97
2.30	Comparison of BZR and ZR for Type C KPG.	97
2.31	Approximate PNE for the EBMC game instances in which no exact PNE was found within the time limit	99

2.32	Approximate PNE for the KPG instances in which no exact PNE was found within the time limit	99
2.33	County-level utility values using different strategy profiles.	100
3.1	Notation for EBMC games	119
3.2	Welfare and selected lakes across models (Minnesota, 100% budget)	131
3.3	Headline metrics for Nash-informed and coalition models	132
3.4	Realized PIR across budget scenarios	135
3.5	Counties left below their reference utility by the SW solution	136
3.6	Headline PIR/POS/POA across three IPG classes	137
3.7	Acronyms used throughout this chapter	144
3.8	County-level participation deficits under the SW solution	152
3.9	Realized PIR under the three reference rules (Minnesota)	154
3.10	Tailored BC models (Minnesota, 100% budget)	155
3.11	Welfare and selected infested lakes under varying budget scenarios	155
3.12	Descriptive statistics for the MN dataset	155
3.13	Welfare range of the EBMC game (Minnesota, 100% budget)	155
3.14	Selected county-level utilities and budget reallocation (Minnesota, 100% budget)	158
3.15	EBMC random instances: full per-instance results	160
3.16	KPG (Types B and C) at $n_{\text{items}} = 100$, ratio = 0.5: full per-instance results	162
3.17	KEG results aggregated by graph size	165

3.18	KEG non-trivial instances ($\text{POS} > 1$)	166
4.1	ISBP Experiment 2: ABS results for non-PNE instances.	221
4.2	GKG benchmark: instances solved by CEI-GZR within 10 minutes (out of 60 per cell).	224
4.3	GKGB summary.	227
4.4	GKGB Experiment 2: ABS results for non-PNE instances.	227
4.5	Impact of VEST cuts: CEI-GZR with and without VEST, averaged over matched instances.	228
4.6	Comprehensive results for ISBP regular instances by (n, m)	238
4.7	Comprehensive results for ISBP non-regular instances by (n, m)	239
4.8	Comprehensive GKG scalability results for $n = 2$ to $n = 20$	240
4.9	Comprehensive GKG scalability results for $n = 25$ to $n = 50$	241
4.10	Comprehensive GKGB results.	242
4.11	Comprehensive NFG results by capacity, number of players, and layers. . . .	243
5.1	The four OCS configurations tested.	267
5.2	OCS formulation comparison, aggregated (AGG) vs. disaggregated (DIS): integer performance and root LP bounds.	278
5.3	OCS separation strategy comparison (AGG formulation).	280
5.4	Notation for the OCS formulation.	287

5.5	Coalition-value evaluations: proposed pipeline (lazy_MIP_ws, AGG) versus the exhaustive baseline. Measured V -enumeration times differ by α —at $n = 18$, 3,628 seconds at $\alpha = 0.5$ against 2,127 at $\alpha = 1.0$ —and the brute-force columns report the $\alpha = 1.0$ series, which understates the baseline’s cost at $\alpha = 0.5$	302
5.6	Ablation: lifted versus basic stability cuts within lazy_MIP (AGG, no warm start).	303
5.7	Payoff refinement summary over the 146 formed OCS coalitions of size ≥ 2	303
5.8	OCS formulation effect: AGG vs. DIS within lazy_MIP and lazy_MIP_ws.	304
5.9	The cutplane and lazy configurations under the disaggregated formulation (DIS), $n \leq 20$	305
5.10	Diagnostics of the SCS-feasible heuristic (lazy_MIP_ws, AGG).	306
5.11	Coalition structures for small instances ($n \leq 12$).	308
5.12	Per-coalition payoff refinement for small instances ($n \leq 12$).	309

Chapter 1

Introduction

Many consequential decisions in the public and private sectors are, at their core, *integer programs (IPs)* [47, 109, 142]: a decision maker selects among discrete, indivisible options— which sites to staff, which projects to fund, which items to pack—subject to budgets and combinatorial constraints. These problems become substantially richer, and substantially harder, when the decisions are made not by one authority but by many self-interested agents who act simultaneously and whose payoffs are coupled. Each agent then solves an IP that reflects its own objectives and constraints, while the value of its decision depends on what the others choose. This is the setting of *integer programming games (IPGs)*, a class of games in which every player’s strategy set is described implicitly by an integer program rather than enumerated as a finite list of actions [36, 37, 96].

IPGs bridge two mature disciplines—integer programming and algorithmic game theory [127]—and in doing so inherit the central difficulties of both. A Nash equilibrium in mixed strategies always exists in a finite game [108], but the discrete structure of an IPG makes the pure-strategy question far more delicate: a *pure Nash equilibrium* (PNE) need not exist, and deciding whether one does is Σ_2^P -complete for general IPGs [35]. Even in the classes where existence is guaranteed—potential and congestion games, for instance [105, 126]—computing an equilibrium can be intractable, and outside those classes best-response play may cycle indefinitely rather than settle, a phenomenon formalized through sink equilibria and the dynamics-based view of computation [73, 115]. Two questions therefore dominate the

study of IPGs: *when* do stable outcomes exist, and *how* can they be computed, selected, or improved upon at the scale that real applications demand. This dissertation is organized around those two questions and pursues them across three major forms of strategic interaction—purely non-cooperative play, generalized games with shared resources, and cooperative coalition formation.

Where the computational literature stands. A decade of work has produced a toolkit for computing equilibria of IPGs. Carvalho et al. [37] survey the field and organize the methods into six families. Mixed-equilibrium methods include the sample generation method and its enumerative variant [36, 49] and the Cut-and-Play algorithm [38]; pure-equilibrium methods include branching schemes for quadratic IPGs [128, 135] and the zero-regret cutting-plane algorithm, which optimizes an objective *over* the set of PNEs by separating globally valid equilibrium inequalities [58]. What these methods share is that equilibrium computation is ultimately reduced to one or more joint optimization problems coupling all players’ strategy sets, either explicitly in normal form or implicitly through extended mixed-integer or complementarity formulations. As a consequence, the reported computational studies concentrate on instances with few players and few integer variables per player—typically with $n \cdot m \leq 300$, where n is the number of players and m the number of integer variables per player (Chapter 2 documents the largest reported instances method by method). That is a natural starting point, but it leaves open how far tractable equilibrium computation can be pushed when $n \cdot m$ runs into the thousands, as it does in the statewide application studied here.

Two extensions of the basic setting matter for the chapters that follow. In *generalized* games the players’ feasible sets are coupled by shared constraints, so a deviation that is privately feasible may be jointly infeasible; the concept dates to Rosen [125], and mixed-integer in-

stances of it have only recently become computationally accessible [61, 62, 84, 85, 130]. In *cooperative* games the object of study is not a strategy profile but a coalition and a division of its value, with the Core, the Shapley value, and the nucleolus as the classical solution concepts [14, 134, 136], and with coalition-structure generation and its computational complexity as the algorithmic counterpart [40, 54, 123, 132]. Between these poles lies a question that runs through the entire dissertation: how much system-level performance is lost by respecting the agents’ autonomy? Algorithmic game theory answers it with the price of anarchy and the price of stability [3, 110, 127], and the operations literature poses a closely related question about the efficiency cost of fairness constraints [16]. Chapter 3 adds a third measure of the same kind, the price of individual rationality, which prices the guarantee that no participant is made worse off.

A methodological through-line. The unifying idea of this dissertation is that the equilibrium of an integer programming game is not merely an object to be certified, but a *computable and leverageable* artifact around which better system-level decisions can be built. Four methodological movements develop this idea. First, we make equilibrium computation *scale*: we introduce random-restart best-response dynamics and embed them in a cutting-plane solver so that pure equilibria of large IPGs—previously reported only for games far smaller than our application—can be found and sampled. Second, we *leverage* the equilibria so computed: rather than accept the best equilibrium as the final answer, we convert equilibrium payoffs into individual-rationality floors and optimize social welfare above them, improving the system without making any player worse off. Third, we extend the machinery to *coupled* settings, where players share scarce resources and exact equilibria may fail to exist, by designing conditional equilibrium inequalities and approximate-equilibrium certificates. Fourth, we cross from competition to *cooperation*, defining coalition values through pooled integer programs and computing coalition structures that are simultaneously optimal

and stable. Throughout, the technical backbone is the same: globally valid inequalities and best-response oracles wired into branch-and-cut, so that one can optimize an objective *over* an equilibrium or stability set rather than merely locate a single feasible point in it.

A running application. The methods are general, but the dissertation is anchored by a concrete public-sector problem: aquatic invasive species (AIS) prevention in Minnesota. Invasive species impose large and persistent economic and ecological costs, and the operations research literature has developed a substantial body of models for allocating scarce prevention, surveillance, and control resources [24]. That literature has itself progressed from continuous control toward increasingly detailed discrete and stochastic formulations: early dynamic-control models of invasion and its economic cost [25] were refined into age-structured bioeconomic models that resolve the population dynamics driving the optimal policy [26], and from there into deterministic mixed-integer formulations for surveillance and control [91], simulation-optimization for search and treatment planning [111, 112], and multistage stochastic and risk-averse formulations that hedge against uncertain spread [21, 92]. Optimization has in fact been applied broadly across environmental and public-health systems in which a limited budget must be spread over a spreading process, as the parallel literature on epidemic logistics and resource allocation for infectious disease shows [27, 144, 145]. What these share with the setting studied here is a scarce budget spent on discrete interventions against a process that propagates through a network; what distinguishes the present problem is that the budget is not held by a single planner. In the Great Lakes region specifically, aquatic invaders have become a defining management problem [65]. Minnesota is a natural setting for a decentralized formulation. The state has 87 counties and 9,182 lakes, of which 506 were infested with at least one AIS type as of 2018; roughly \$10 million per year was appropriated to county AIS prevention programs during the study period [57, 102], and the resulting watercraft inspection effort is planned and executed independently by the counties

rather than by the state. Because recreational boat traffic links lakes across county lines, an inspection paid for by one county protects lakes in another. Kao et al. [90] estimate this statewide boat-movement network, and it is what turns a collection of local budgeting problems into a game: county decisions generate positive externalities and are naturally modeled as an *edge-weighted budgeted maximum coverage* (EBMC) game over roughly 9,000 lakes. Chapter 2 solves an instance with 84 county players; Chapter 3 focuses on the 38 counties that are strategically active at the baseline budget.

This application also has a well-developed centralized counterpart, which sharpens what the game-theoretic formulation adds. Haight et al. [81] optimize the location of watercraft inspection stations to slow AIS spread, Haight et al. [82] pose a bi-level state-county prevention model, and Angell et al. [2], Kinsley et al. [93, 94] develop decision-support tools that help counties prioritize inspections and plan collaboratively. These models assume a planner who can direct effort. A parallel line of work relaxes that assumption using game theory: Büyüktaktın et al. [29] allocate the gains from coordinated invasive-species control through cooperative game theory, and Chen et al. [44] design reimbursement incentives that lead private landowners to undertake socially beneficial treatment. The formulation studied here is different again: county budgets, jurisdictional boundaries, and statutory authority are taken as fixed, no transfers or contracts are introduced, and the counties' discrete inspection decisions are modeled explicitly as the strategies of an integer programming game. Under those constraints the outcome improves markedly. Relative to the reference equilibrium the counties would otherwise reach, the recommendation produced in Chapter 3 raises the number of protected infested lakes from 220 to 302—an increase of roughly 37%—while guaranteeing every county at least its equilibrium payoff and requiring no inter-county transfers, and the welfare cost of that guarantee stays within 0.15% of the unconstrained optimum across budget scenarios from 50% to 125% of current funding (Table 3.11).

Contributions. The dissertation makes four principal contributions, developed in Chapters 2 to 5, which can be read along three dimensions.

Theoretical. We establish a Las Vegas convergence guarantee for round-based random-restart best-response dynamics in finite games, giving conditions under which the probability of reaching a PNE tends to one as restarts accumulate. We define the *price of individual rationality* (PIR) and prove that it is bounded above by the price of stability when reference utilities are drawn from the welfare-best equilibrium. We introduce *conditional equilibrium inequalities* (CEIs), which are, to the best of our knowledge, the first globally valid equilibrium inequalities for generalized IPGs, and we prove the *vertex equilibrium structure theorem* characterizing when equilibrium strategies must lie at vertices of the conditional strategy polytope. For cooperative IPGs we derive a stability inequality that keeps every formed coalition in its own Core, characterize when the resulting mixed-integer formulations are integer-equivalent, and show that the disaggregated formulation yields a strictly tighter linear relaxation.

Algorithmic. We develop RRR-BRD and its hybridization with the zero-regret cutting-plane method (BZR), which together substantially advance the scale at which pure equilibria of IPGs can be computed and sampled. We give a generalized zero-regret algorithm that separates CEIs as lazy constraints and optimizes over the generalized equilibrium set, together with an α -bisection search returning certified approximate equilibria when exact ones may not exist, and structural (VEST) cuts for cost-sharing games such as integer splittable bin packing. For cooperative IPGs we give lifted stability cuts, separation strategies, a stability-feasible primal heuristic, and a payoff-refinement step computing the Shapley value and the nucleolus of every formed coalition.

Application. We formulate the Minnesota AIS inspection problem as a large-scale EBMC game and compute its equilibria at statewide scale; we translate the equilibria into imple-

mentable, voluntarily acceptable statewide recommendations and quantify their ecological benefit and their welfare cost; and we deliver managerial guidance on where inspection capacity should be directed, how coalitions among counties should be formed, and how robust these recommendations are to changes in the funding level.

Organization. Chapter 2 develops scalable computation and sampling of pure Nash equilibria for non-cooperative IPGs and applies it to the Minnesota AIS EBMC game. Chapter 3 shows how to leverage those equilibria as guarantees to improve social welfare without harming any player, and quantifies the resulting welfare gains under a range of budget scenarios. Chapter 4 treats generalized IPGs with coupled feasibility, providing exact and approximate equilibrium certificates via conditional inequalities, separation, and structural cuts. Chapter 5 turns to cooperative IPGs, defining coalition values through pooled integer programs and computing optimal, stable coalition structures. Chapter 6 concludes by drawing together the non-cooperative, generalized, and cooperative threads and outlining directions for future research.

Underlying manuscripts and reproducibility. Chapters 2 to 5 are each based on a separate manuscript, adapted and expanded for this dissertation; notation, exposition, and organization have been revised for consistency across chapters. Each chapter repeats this provenance in a manuscript note at its opening; the four manuscripts and the repositories holding their code and data are collected here for reference.

- Chapter 2 is based on H. Lee, R. Hildebrand, W. Cai, and İ. E. Büyüktaktakın, “Random-Restart Best-Response Dynamics for Large-Scale Integer Programming Games and Their Applications,” arXiv:2409.04078 [98], invited for resubmission at *Management Science*. Code and data: <https://github.com/HyunwooLee0429/Best-respo>

[nse-dynamics-IPG](#).

- Chapter 3 is based on H. Lee, R. Hildebrand, W. Cai, A. Kinsley, N. Phelps, and İ. E. Büyüктаhtakın, “Nash-informed Planning for Public-Resource Coordination: An Application to Aquatic Invasive Species Prevention,” working paper. Code and data: <https://github.com/HyunwooLee0429/nash-informed-planning>.
- Chapter 4 is based on B. Hao, H. Lee, R. Hildebrand, and C. Michini, “Conditional Equilibrium Inequalities for Generalized Nash Equilibrium Problems with Pure-Integer Variables,” working paper. Code and data: <https://github.com/HyunwooLee0429/GIPG>.
- Chapter 5 is based on H. Lee, R. Hildebrand, İ. E. Büyüктаhtakın, and J. Yang, “Cooperative Integer Programming Games: Core Stability and Optimal Coalition Structures,” working paper. The paper was peer reviewed and accepted for presentation at the 2026 INFORMS Optimization Society Conference, with the camera-ready version available at <https://zenodo.org/records/19380553>, and has been invited for expedited review at *INFORMS Journal on Optimization*. Code and data: <https://github.com/HyunwooLee0429/CIPG>.

Chapter 2

Random-Restart Best-Response Dynamics for Large-Scale Integer Programming Games and Their Applications

Manuscript Note. This chapter is based in part on the following manuscript:

H. Lee, R. Hildebrand, W. Cai, and İ. E. Büyüктаhtakın, “Random-Restart Best-Response Dynamics for Large-Scale Integer Programming Games and Their Applications,” arXiv:2409.04078 [98]. Invited for resubmission at *Management Science*.

The manuscript has been adapted and expanded for inclusion in this dissertation; notation, exposition, and organization have been revised for consistency with the remaining chapters. The preprint is available at <https://arxiv.org/abs/2409.04078>, and code and data at <https://github.com/HyunwooLee0429/Best-response-dynamics-IPG>.

Keywords: Integer Programming Games, Pure Nash Equilibrium, Socially Optimal PNE, Best-Response Dynamics, Sink Equilibria, Aquatic Invasive Species

Funding: This work was supported in part by the Air Force Office of Scientific Research (AFOSR) and the Minnesota Aquatic Invasive Species Research Center (MAISRC).

2.1 Introduction

Integer programming games (IPGs) stand at the crossroads of integer programming and non-cooperative game theory. They study settings in which each player solves an optimization model with discrete decisions, and these decisions influence one another’s utilities. IPGs are non-cooperative, complete-information games: players are self-interested and have complete information about each other’s utilities and strategy sets [36]. A pure Nash equilibrium (PNE), as defined by Nash [108], is a strategy profile in which no player can improve their utility by unilaterally changing their strategy. Social welfare is the sum of all players’ utilities. In this chapter, we develop efficient algorithms for large-scale IPGs to (i) compute a PNE, (ii) compute multiple PNEs, and (iii) identify the PNE with the highest social welfare.

Computing multiple PNEs is an important task in non-cooperative games, as some equilibria may be more desirable than others. In particular, a socially optimal PNE—a PNE that maximizes total social welfare—is a natural target in settings with mediators or central planners who care about system-wide performance. This notion is also central in algorithmic game theory [127], where it underpins standard measures of equilibrium inefficiency relative to socially optimal outcomes. In the IPG literature, two recent lines of work address (i) equilibrium computation and enumeration and (ii) selecting a welfare-maximizing equilibrium: Dragotto and Scatamacchia [58] for PNEs, and Crönert and Minner [49] for mixed Nash equilibria (MNEs). However, both approaches remain limited to small-scale instances.

Although recent advances in IPG algorithms have improved our ability to compute PNEs, most methods remain limited to small-scale games with only two or three players ($n \leq 3$) [58] or with a smaller number of variables ($m \leq 5$) [128, 135]. Given the maturity of the field, it is timely to bridge this gap by addressing the large-scale, multi-stakeholder instances required for real-world decision-making. Compelling examples arise in resource allocation

decisions across decentralized administrative structures, such as the many states or counties in the United States, naturally forming large-scale IPG settings. Motivated by Minnesota’s AIS prevention efforts, which are independently managed by 84 counties, we introduce a new class of non-cooperative IPGs: *edge-weighted budgeted maximum coverage* (EBMC) games. These games model county-level decision-making under budget constraints and inter-county externalities. Solving such problems requires scalable and efficient algorithmic approaches.

To address this challenge, we focus on *random-restart best-response dynamics* (RR-BRD), a randomized variant of best-response dynamics (BRD) [101]. We adopt a Markov chain framework and model RR-BRD as a random walk on the best-response (BR) state graph of pure strategy profiles, as introduced by Goemans et al. [73]. A basic observation is that the induced random walk eventually enters a recurrent class of the BR state graph; these *sink equilibria* include singleton PNEs as well as non-PNE sink components. RR-BRD mitigates such traps via a restart strategy with a fixed cap on the number of steps per attempt. We show that, whenever a PNE exists and the restart distribution assigns positive probability to its basin of attraction, RR-BRD finds a PNE almost surely, with an instance-dependent bound on the expected number of best-response solves. In practice, we incorporate RR-BRD as a randomized local-search subroutine within the zero-regret (ZR) framework of Dragotto and Scatamacchia [58]. This hybrid approach dramatically improves the scalability of PNE computation, increasing the number of solvable players from small-scale cases ($n = 2, 3$) to up to 30 players in random datasets and 84 players in real-world datasets.

As a motivating application for large-scale IPGs, we build on optimization-based approaches to invasive species control [24, 44, 92] by introducing EBMC games, which arise naturally in decentralized, county-level AIS prevention across U.S. lakes. Previous studies have applied the EBMC formulation [39] to model AIS inspection at the county [81] and state levels [82], while others have examined AIS spread across network graphs [65]. In EBMC games, each

player controls a subset of vertices in a network graph, and each player maximizes their edge coverage under budget constraints. We define two variants of the utility functions—selfish and locally altruistic—and establish theoretical conditions for the existence of a PNE in these models. Additionally, we validate our IPG framework through experimental results on EBMC games using random and real-world datasets, as well as on knapsack problem games (KPGs) with random datasets. Our findings reveal that IPGs, coupled with the enhanced solvability of the BZR algorithm presented in this chapter, offer a powerful and scalable framework for modeling and solving large-scale multi-agent problems.

2.1.1 Literature Review

Integer programming games (IPGs) are gaining attention for their ability to model strategic interactions beyond traditional normal-form games. Various applications of IPGs have been studied, including facility location and design games [49], fixed-charge transportation problems [131], kidney exchange games [34], quadratic IPGs (qIPGs) [128, 135], and knapsack problem games (KPGs) [38, 58]. In these models, each player solves a player-specific integer program (IP), and strategic interactions are typically captured through reciprocally bilinear terms that couple players' decision variables, with either linear objectives (e.g., KPG) or quadratic objectives (e.g., qIPG).

Carvalho et al. [37] provide a comprehensive overview of the IPG solution methods, categorizing them into six approaches. Among these, sample generation method (SGM) [36], enumerative SGM (eSGM) [49], and Cut-and-Play (CnP) [38] compute MNEs, while branching method (BM) [128], zero-regret (ZR) [58], and branch-and-prune (BnP) [135] focus on PNEs. Notably, Carvalho et al. [36] and Crönert and Minner [49] use normal-form game methods to compute MNEs in sampled games before verifying them in the original setting.

Among exact methods to compute PNEs, the zero-regret (ZR) algorithm [58] is state-of-the-art for computing socially optimal PNEs via a cutting-plane approach. For approximate NEs, Sankaranarayanan [133] presents a best-response algorithm to compute approximate PNE and Duguet et al. [63] show a piecewise linear approximation to compute approximate MNE in the mixed-integer nonlinear programming cases based on SGM.

In most approaches, equilibrium computation is ultimately reduced to one or more joint optimization problems that couple all (or a subset of) players' strategy sets, either explicitly in normal form or implicitly via extended MIP or complementarity formulations. Consequently, reported computational studies focus on instances with small numbers of players and integer variables per player (typically with $n \cdot m \leq 300$, where n is the number of players and m is the number of integer variables per player; see also Section 2.4.3). This provides a useful baseline but leaves open the question of how far one can push the tractability of equilibrium computation in IPGs when the number of players is much larger—for example, when $n \cdot m$ is in the thousands.

Best-response dynamics (BRD) [101] is a foundational and commonly used method for computing PNEs in normal-form games, where players take turns solving their best-response problems while keeping the strategies of the others fixed. In well-structured classes such as potential games [105] and, more generally, best-response-weakly acyclic games [5], the convergence of BRD to a PNE is well understood. However, for arbitrary normal-form games the best-response state graph may contain directed cycles and non-singleton sink strongly connected components. Goemans et al. [73] interpret the long-run behavior of such dynamics through *sink equilibria*, which capture recurrent behavior of best-response walks; see also the broader dynamics-based perspective of Papadimitriou and Piliouras [115]. Relatedly, Hakim et al. [83] study *limit* (infinite-horizon) absorption over sink SCCs of a (better-or-equal) response graph under a different noisy dynamics; our analysis uses a Markov chain induced by

(improving) best-response updates and focuses on finite-horizon PNE-hitting probabilities.

Recent work in algorithmic game theory (AGT) has studied randomized variants of BRD that still follow a single trajectory but randomize either the initial profile or the updating player at each step. For example, Amiet et al. [1] and Heinrich et al. [88] consider games with random payoffs: Amiet et al. [1] study BRD in games with binary strategy profiles, and Heinrich et al. [88] analyze BRD with randomized playing sequences and randomly drawn initial strategy profiles. These works introduce randomness through the play sequence and initial draws from a known or assumed distribution, but they do not employ restart mechanisms. In the IPG context, Sankaranarayanan [133] analyzes a best-response algorithm with fixed playing sequence for lattice convex-quadratic simultaneous games and shows that, when the sequence of strategy profiles enters a finite trap (detected by a repeated profile), one can extract an approximate MNE from the strategies in that trap. To our knowledge, prior to this work, BRD-type procedures had not been developed as a scalable, restart-based algorithm for computing PNEs in IPGs with finite strategy sets.

2.1.2 Contribution

Our detailed contributions are summarized as follows:

- We introduce *random-restart BRD* (RR-BRD) into the IPG literature and analyze it as a randomized search framework for computing PNEs in large-scale games. For IPGs with finite strategy sets, we adopt the best-response (BR) state-graph viewpoint of Goemans et al. [73] and model RR-BRD as a Markov chain on this graph. We show that, whenever a PNE exists and the restart distribution assigns positive probability to states from which a PNE is reachable within the round cap, RR-BRD finds a PNE with probability one and admits an instance-dependent bound on the expected

number of BR solves. We further contribute to the normal-form game literature by providing a convergence guarantee for BRD under restarts stated directly in terms of BR-state-graph reachability, without assuming random payoffs or restrictive structural conditions (e.g., potential games or BR-weakly acyclic games).

- We propose a Monte Carlo sampling-and-simulation procedure, in the spirit of Hakim et al. [83], to estimate per-attempt success under a fixed round cap and to inform an instance-dependent performance characterization of RR-BRD. We also discuss best- and worst-case behavior on general BR state graphs and give a knapsack-style IPG instance illustrating worst-case behavior.
- Building on this framework, we propose *BRD-incorporated zero-regret* (BZR), which embeds RR-BRD as a randomized local-search subroutine within the ZR [58]. Using solver callbacks, BZR invokes RR-BRD to search for and supply PNEs, while ZR adds equilibrium inequalities to tighten the formulation. In BZR, integer-feasible incumbents produced by the mixed-integer program (MIP) solver provide randomized initial strategy profiles for RR-BRD, and equilibrium inequalities are generated from the best-response profiles encountered along RR-BRD trajectories. We also discuss how RR-BRD can serve as a modular local-search subroutine within other IPG algorithms.
- We introduce edge-weighted budgeted maximum coverage (EBMC) games, a novel class of non-cooperative integer programming games where players maximize edge coverage under budget constraints on a network. We examine two variants of the players' utility functions—locally altruistic and selfish—and prove the existence of a PNE in locally altruistic EBMC games via potential game arguments, as well as provide sufficient conditions for the existence of a PNE in selfish EBMC games.
- We conduct extensive experiments to assess the scalability and effectiveness of RR-

BRD and BZR on synthetic EBMC and KPG instances with up to 30 players. Across 135 test cases (27 per variant for the two EBMC and three KPG types), RR-BRD finds a PNE in 128 instances. When embedded in ZR, BZR substantially improves performance: it finds at least one PNE in 128 instances (vs. 33 for ZR) and identifies 179 PNEs for EBMC (vs. 35) and 3160 for KPGs (vs. 38). Overall, RR-BRD is effective both as a standalone algorithm and as a local-search subroutine that greatly improves ZR’s scalability.

- We implement our algorithms on a real-world EBMC instance with 84 county planners in Minnesota’s AIS inspection programs. For the selfish EBMC game, RR-BRD finds a socially optimal PNE; without this algorithm, it would be unclear whether a PNE exists, making PNE-based benchmarks difficult to use. For the locally altruistic EBMC game, we identify a socially optimal solution that is also a PNE. Comparing these equilibria, we show how the behavioral specification (selfish vs. locally altruistic) changes both system-wide performance and the distribution of individual utilities, with locally altruistic equilibria often yielding higher individual utilities.

This chapter is organized as follows. Section 2.2 provides preliminaries on IPGs, normal-form games, and the best-response state graph. Section 2.3 presents our RR-BRD framework and its probabilistic convergence guarantees. Section 2.4 shows how RR-BRD can be incorporated into the ZR framework and other algorithms. Section 2.5 formally defines EBMC games as an illustrative IPG, highlighting variations in utility functions across players. Section 2.6 establishes conditions for the existence of PNEs in EBMC games. Section 2.7 reports computational experiments that demonstrate the effectiveness and generalizability of RR-BRD and BZR on both random and real-world datasets, including random knapsack game datasets representing broader classes of resource-constrained IPGs. Finally, Section 2.8 offers concluding remarks. For quick reference on acronyms and notation used throughout this

chapter, see Appendix [2.A](#).

2.2 Preliminaries

This section provides the modeling and state-graph foundations used throughout this chapter. Section [2.2.1](#) defines IPGs and PNE, and Section [2.2.2](#) introduces the normal-form best-response state graph (and its Markov-chain reduction) used to describe and analyze our RR-BRD in Section [2.3](#).

2.2.1 Integer Programming Games and Pure Nash Equilibria

We introduce the fundamental definitions and concepts related to IPGs and PNEs. Following Köppe et al. [[96](#)], an IPG is defined as a tuple $\mathcal{G} = (N, (\mathcal{X}_i)_{i \in N}, (u_i)_{i \in N})$, where $N := \{1, \dots, n\}$ denotes the set of players, $\mathcal{X}_i := \{\mathbf{x}^i \in \mathbb{Z}^{m_i} : A^i \mathbf{x}^i \leq \mathbf{b}^i\}$ represents the pure-integer strategy set for player i with rational matrix A^i , vector $\mathbf{b}^i$, and dimension m_i , and $u_i(\mathbf{x}^i, \mathbf{x}^{-i}) : \mathcal{X} \rightarrow \mathbb{R}$ is the utility function for player i , parametrized by the strategies $\mathbf{x}^{-i}$ of all other players. Here, $\mathcal{X} := \mathcal{X}_1 \times \dots \times \mathcal{X}_n$ denotes the joint strategy space.

Throughout, we assume that each $\mathcal{X}_i$ is nonempty and *polyhedrally bounded*, so that each player has a finite number of pure strategies and the joint strategy space $\mathcal{X}$ is finite. This assumption is used when relating IPGs to finite normal-form games in the sequel.

The utility function of player i is often expressed as

$$u_i(\mathbf{x}^i, \mathbf{x}^{-i}) = (\mathbf{x}^i)^\top Q^i \mathbf{x}^i + (\mathbf{d}^i)^\top \mathbf{x}^i + \sum_{p \in N \setminus \{i\}} (\mathbf{x}^p)^\top Q_p^i \mathbf{x}^i, \quad (2.1)$$

where Q^i is a symmetric matrix (not necessarily positive semidefinite), $\mathbf{d}^i$ is a linear coeffi-

cient vector, and the terms $(\mathbf{x}^p)^\top Q_p^i \mathbf{x}^i$ capture inter-player interactions through *reciprocally bilinear* terms.

In an IPG, players act non-cooperatively, each solving an individual integer program (IP) to maximize their utility. The best-response problem for player i is

$$\max_{\mathbf{x}^i} \{u_i(\mathbf{x}^i, \mathbf{x}^{-i}) : \mathbf{x}^i \in \mathcal{X}_i\}. \quad (2.2)$$

We assume that the best-response problems (2.2) are tractable and solved to optimality.

A *strategy profile* $\mathbf{x} = (\mathbf{x}^1, \dots, \mathbf{x}^n)$ is a collection of all players' strategies. A profile $\hat{\mathbf{x}}$ is a *pure Nash equilibrium (PNE)* if, for each player $i \in N$, the strategy $\hat{\mathbf{x}}^i$ is a best response (BR) to the others' strategies $\hat{\mathbf{x}}^{-i}$, i.e., $u_i(\hat{\mathbf{x}}^i, \hat{\mathbf{x}}^{-i}) \geq u_i(\mathbf{x}^i, \hat{\mathbf{x}}^{-i})$, $\forall \mathbf{x}^i \in \mathcal{X}_i$. Carvalho et al. [35] show that deciding whether a general IPG admits a PNE is a Σ_2^P -complete problem and that PNEs may fail to exist even in simple two-player IPGs.

We define the social welfare function as $\phi(\mathbf{x}) := \sum_{i \in N} u_i(\mathbf{x}^i, \mathbf{x}^{-i})$, and refer to a *socially optimal* strategy profile (or *optimal social-welfare* (OSW) solution) as $\bar{\mathbf{x}}_{\text{sw}}^* \in \arg \max_{\mathbf{x} \in \mathcal{X}} \phi(\mathbf{x})$. Let $\mathcal{S}_{\text{pne}}(\mathcal{G})$ denote the set of all PNEs of the IPG instance $\mathcal{G}$. The *best PNE* is then defined as $\hat{\mathbf{x}}_{\text{pne}}^* \in \arg \max_{\mathbf{x} \in \mathcal{X} \cap \mathcal{S}_{\text{pne}}(\mathcal{G})} \phi(\mathbf{x})$. The *price of stability* (POS) [127], a common measure of inefficiency in game theory, is defined as the ratio $\text{POS} = \frac{\phi(\bar{\mathbf{x}}_{\text{sw}}^*)}{\phi(\hat{\mathbf{x}}_{\text{pne}}^*)}$. A larger POS indicates a larger gap in social welfare between the best PNE and the socially optimal solution.

IPGs as finite normal-form games. When each player's pure-integer feasible set $\mathcal{X}_i$ is finite and polyhedrally bounded, an IPG can be viewed as a finite normal-form game by setting $S_i := \mathcal{X}_i$ as the pure strategy set of player i and defining utilities u_i at each joint profile to equal the IP objective values. This representation preserves payoffs and best responses and induces the same best-response state graph and Markov chain as those defined

below. We emphasize that this is a conceptual viewpoint—explicit enumeration of $\prod_{i \in N} \mathcal{X}_i$ is generally intractable—and our methods avoid it by computing BRs via player-specific IPs.

2.2.2 Normal-Form Games, BR State Graphs, and Markov Reduction

Let $N = \{1, \dots, n\}$ denote the players. For each $i \in N$, let S_i be a *finite* set of pure strategies and set $S := \prod_{i \in N} S_i$. Utilities $u_i : S \rightarrow \mathbb{R}$ are arbitrary but deterministic. For each $i \in N$ and profile $s = (s_i)_{i \in N} \in S$, define the set of strictly improving best responses (BRs) as $\mathcal{BR}_i^+(s) := \{t_i \in \arg \max_{r_i \in S_i} u_i(r_i, s_{-i}) : u_i(t_i, s_{-i}) > u_i(s_i, s_{-i})\}$. We say that player i has an *improving BR move* at s if $\mathcal{BR}_i^+(s) \neq \emptyset$; otherwise, player i is already best-responding at s (possibly with ties), and we regard this as a *no-move* situation for player i .

Best-response state graph and a time-homogeneous Markov chain. The *best-response (BR) state graph* is $\mathcal{U} = (S, E)$ with a directed edge $s \rightarrow (t_i, s_{-i})$ for each $s \in S$, $i \in N$, and $t_i \in \mathcal{BR}_i^+(s)$; that is, edges correspond to strictly improving BR moves. We also interpret no-move updates as self-loops $s \rightarrow s$. A profile s^* is a PNE iff $\mathcal{BR}_i^+(s^*) = \emptyset$ for all $i \in N$ —equivalently, s^* has no outgoing strictly improving BR edges, so all BR updates reduce to self-loops.

Fix a scheduler π with $\pi_i > 0$ for all $i \in N$ (e.g., uniform $\pi_i = 1/n$). For each player $i \in N$ and state $s \in S$, we specify a *tie-breaking kernel* $\lambda_i(\cdot \mid s)$, a probability distribution on S , such that:

- if $\mathcal{BR}_i^+(s) \neq \emptyset$, then $\lambda_i(\cdot \mid s)$ is supported on the improving BR successors $\{(t_i, s_{-i}) : t_i \in \mathcal{BR}_i^+(s)\}$, and each such successor has positive probability (for instance, uniform choice over $\mathcal{BR}_i^+(s)$);

- if $\mathcal{BR}_i^+(s) = \emptyset$, then $\lambda_i(s \mid s) = 1$, i.e., player i does not move when already best-responding (even if there are other tied best responses).

We then define transition probabilities

$$P(s, s') := \sum_{i \in N} \pi_i \lambda_i(s' \mid s), \quad s, s' \in S. \quad (2.3)$$

Thus $(X_t)_{t \geq 0}$ with $X_{t+1} \sim P(X_t, \cdot)$ is a finite, time-homogeneous Markov chain whose positive-probability transitions are exactly the edges of $\mathcal{U}$ (including self-loops). By the Markov property, the next state depends on the current state and a fresh draw $I_t \sim \pi$ together with the tie-breaking kernels. At a PNE s^* , $\mathcal{BR}_i^+(s^*) = \emptyset$ for all i , hence $\lambda_i(s^* \mid s^*) = 1$ and $P(s^*, s^*) = 1$, so $\{s^*\}$ is an absorbing class. Standard notions such as communicating classes, recurrent and transient states, hitting probabilities, and expected hitting times apply.

2.3 Convergence of Best-Response Dynamics in Finite Normal-Form Games

We establish convergence guarantees for random-restart best-response dynamics (RR-BRD) in IPGs via the following steps: (i) cast an IPG with finite pure-integer feasible sets as a finite normal-form game; (ii) construct the best-response (BR) state graph; (iii) compute its strongly connected components (SCCs) and the associated condensation directed acyclic graph (DAG); and (iv) identify sink SCCs, distinguishing singleton *PNE sinks* from multi-node *non-PNE sinks*. Steps (i)–(ii) are described in Section 2.2. The baseline fact is that a random walk on the BR state graph eventually enters a sink SCC and remains there; thus it converges either to a singleton PNE sink or to a non-PNE sink (trap) [73]. Our addition is to

introduce *random restarts* aimed at reaching singleton PNE sinks, yielding a Las Vegas-style guarantee: if a PNE exists and the restart distribution places positive mass on any state from which some PNE sink is reachable, then RR-BRD finds (and certifies) a PNE almost surely.

2.3.1 SCCs, the Condensation DAG, and Recurrent Classes

Let $\text{SCC}(\mathcal{U})$ denote the set of strongly connected components of $\mathcal{U}$, and let $\mathcal{D}$ be the *condensation DAG* of $\mathcal{U}$, obtained by contracting each SCC (so $\mathcal{D}$ is finite and acyclic). An SCC S is a *sink SCC* if and only if no BR edge leaves S (equivalently, the node corresponding to S is a sink in $\mathcal{D}$). We distinguish: (i) a *PNE sink*, i.e., a singleton sink SCC $\{s^*\}$ (necessarily a PNE), and (ii) a *non-PNE sink* (or *trap*), i.e., a sink SCC S with $|S| > 1$. This structure is illustrated in Figure 2.1.

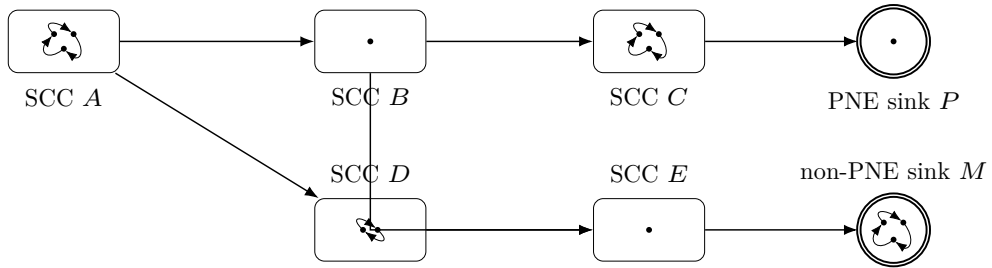


Figure 2.1: Illustration of a connected condensation DAG with two terminal sinks: a singleton *PNE sink* P and a multi-node *non-PNE sink* M . Boxes and circles are SCCs; internal dots depict the number of profiles each SCC contains (singletons vs. multi-node SCCs). Arrows are edges of the acyclic condensation graph. SCC labels (A, B, C, D, E, P, M) are arbitrary and used solely for illustration. In potential or BR-weakly acyclic games, non-PNE sinks do not occur; all sink SCCs are singleton PNEs, so the branch labeled M is absent.

Figure 2.1 illustrates that best-response dynamics may converge either to a singleton PNE sink or to a non-PNE sink, from which escape is impossible—motivating the use of random restarts developed in Section 2.3.3. Because (2.3) assigns positive probability exactly to BR edges, the Markov chain’s communicating classes coincide with the SCCs of $\mathcal{U}$; in particular,

its *recurrent classes* are precisely the *sink SCCs*. Thus, every PNE is a singleton recurrent (absorbing) class, while any non-PNE sink is a closed, non-singleton trap. BRD as a random walk (i.e., equivalently, with randomized playing sequences) samples $I_t \in N$ i.i.d. with $\Pr(I_t = i) = \pi_i > 0$ and, conditional on $I_t = i$, updates s_{t+1} by applying one improving best-response move for player i according to $\lambda_i(\cdot \mid s_t)$. This produces exactly the Markov chain with transition kernel P in (2.3).

Randomized playing sequences [88] can help escape *non-sink* cycles (for example, cycles in an SCC with at least one outgoing edge, such as the SCCs labeled A , C and D in Figure 2.1), whereas a fixed update order can get stuck there (i.e., the fixed update order can prevent the escaping BR move). However, using a randomized playing sequence is not enough; once the chain enters a non-PNE trap, it cannot leave. In particular, when non-PNE traps exist, a random playing sequence alone does *not* ensure reaching a PNE, as emphasized in the sink-equilibrium framework of Goemans et al. [73].

2.3.2 Deterministic Convergence in Structured Subclasses

Potential games and best-response weak acyclicity. (compare Figure 2.1: non-PNE sink is absent in these subclasses) A finite game is a (weighted) *potential game* if some $\psi : S \rightarrow \mathbb{R}$ tracks unilateral improvements: for all i and $t_i \in S_i$, $u_i(t_i, s_{-i}) - u_i(s_i, s_{-i}) = w_i(\psi(t_i, s_{-i}) - \psi(s_i, s_{-i}))$ with $w_i > 0$. By the finite improvement property, every BR path strictly increases ψ and therefore reaches a PNE in finitely many steps [105]; classic subclasses include congestion games [126] and player-specific congestion games [103]. A game is *BR-weakly acyclic* if every sink SCC in the BR state graph is a singleton PNE (equivalently, from every start, there exists a finite BR path to a PNE). In such games, *every* BR trajectory (under any scheduler) reaches a PNE in finite time because infinite trajectories would have

to reside in a non-PNE sink, which is excluded by definition; see, e.g., Apt and Simon [5] for systematic treatments and related characterizations.

Bridge to the general case. Outside these structured subclasses, the BR state graph may contain non-PNE trap components, which can trap BRD even under randomized playing sequences (see the sink-equilibrium viewpoint of Goemans et al. 73). The next subsection shows that introducing *random restarts* yields Las Vegas-style guarantees and instance-dependent bounds in arbitrary finite games.

2.3.3 Round Random-Restart Best-Response Dynamics (RRR-BRD)

RRR-BRD is a round-based variant of BRD that incorporates *random restarts*. A *restart* means that if the dynamics fails to reach a PNE within a prescribed cap, we abandon the current trajectory, reinitialize the strategy profile (according to a restart distribution), and repeat. Instead of updating one player at a time, RRR-BRD *batches* updates into *rounds*: in each round, every player plays one improving BR move, in a uniformly random order. This makes local PNE certification simple: if no player changes in a round (i.e., no improving BR exists when each player is selected), then the current profile is a PNE. Hereafter, we use RRR-BRD to denote the implemented algorithm; we retain RR-BRD as a conceptual umbrella term.

The guarantees we develop are *Las Vegas* rather than deterministic: randomness affects the trajectory and runtime, but the algorithm never returns an incorrect solution—whenever it returns, it outputs a *certified* PNE. If a PNE exists and a capped attempt succeeds with positive probability, then independent unbounded restarts imply almost-sure convergence to a PNE. We now present RRR-BRD (Algorithm 2.1).

Algorithm 2.1 Round Random-Restart BRD (RRR-BRD)**Require:** restart law μ on S ; round cap $R \in \mathbb{N}$

```

1: while true do
2:   sample  $s \sim \mu$  ▷ fixed-cap restart: if no PNE within  $R$  rounds, restart
3:   for  $r = 1, 2, \dots, R$  do
4:     draw a random permutation  $\Pi$  of  $\{1, \dots, n\}$  ▷ uniform
5:      $s^{\text{start}} \leftarrow s$ 
6:     for  $i$  in order  $\Pi$  do apply one improving best-response update for player  $i$ 
7:     if  $s = s^{\text{start}}$  then return  $s$  ▷ PNE certified by end-of-round no-change
8:
9:

```

Non-PNE sinks and the avoidance of trap detection. The BR state graph may contain non-singleton SCCs and, in particular, non-PNE traps. A natural question is whether Algorithm 2.1 could implement a “stop-on-trap” rule—e.g., restart immediately (Line 2) once the trajectory enters a non-PNE trap. While appealing in principle, this requires detecting sink membership, which is computationally intractable in general for succinctly represented games. To the best of our knowledge, there is no general complexity classification of the **IN-A-SINK** problem for arbitrary succinct normal-form representations; existing hardness results are established for specific classes. For example, **IN-A-SINK** is PSPACE-complete for graphical games/BGP [67], and across several natural game classes (e.g., anonymous, player-specific or weighted congestion, valid-utility, and two-sided markets) sink membership and the existence of non-singleton sinks are PSPACE-complete, while even PNE existence can be NP-hard [104]. Accordingly, our algorithm avoids trap detection entirely: Algorithm 2.1 uses only inexpensive *local* checks for PNE certification and fixed-cap restarts, and never requires constructing or reasoning over the full BR state graph.

Round-based (macro) Markov chain and hitting times. For a permutation π of $\{1, \dots, n\}$, let F_π denote the (random) operator that applies exactly one improving BR update for each player in the order π , using the tie-breaking kernels $(\lambda_i)_{i \in N}$ from Section 2.2.2. If $(\Pi_r)_{r \geq 0}$ are i.i.d. uniform permutations, then the round-boundary process $S_{r+1} = F_{\Pi_r}(S_r)$, $r = 0, 1, 2, \dots$, is a time-homogeneous Markov chain on the finite state space S . PNEs are still absorbing states. Moreover, any sink SCC of the stepwise BR state graph remains closed under round transitions, so the recurrent classes are unchanged. For any $C \subseteq S$, define the (round) hitting time $\tau_C := \inf\{r \geq 0 : S_r \in C\}$ and write τ_{PNE} for the hitting time of the set of PNE.

Restarts and per-attempt success probability. A restart law μ is any probability distribution on S . For a fixed round cap $R \geq 1$, define the per-attempt success probability

$$q_\mu[R] := \mathbb{E}_{s \sim \mu} [\Pr_s(\tau_{\text{PNE}} \leq R)], \quad (2.4)$$

i.e., the probability that a single RRR-BRD attempt started from $s \sim \mu$ reaches (and certifies) a PNE within at most R rounds. Note that $q_\mu[R] > 0$ whenever μ assigns positive probability to at least one state from which a PNE is reachable via improving BR moves within R rounds. (In contrast, Hakim et al. [83] study *limit* (infinite-horizon) absorption probabilities into *sink SCCs* under a different noisy dynamics.)

We now establish the following result; proof is given in Appendix 2.B.

Theorem 2.2 (RRR-BRD: almost surely success and expected complexity). *Assume a PNE exists and the restart law μ assigns positive mass to states from which a PNE is reachable via improving best-response moves. If $q_\mu[R] > 0$, then: (i) RRR-BRD finds a PNE with probability 1 and (ii) The expected total number of best-response solves is bounded by*

$$\mathbb{E}[\#BR \text{ solves}] \leq \frac{nR}{q_\mu[R]}.$$

Remark 2.3. If a PNE exists and μ has full support on S , then $q_\mu[R] > 0$ for all $R \geq 1$ because μ assigns positive probability to every state, including each PNE state (and thus a PNE can be reached within one capped attempt). Therefore, the condition $q_\mu[R] > 0$ in Theorem 2.2 holds under any full-support restart law and more generally whenever μ assigns positive mass to the attraction basin of some PNE.

Theorem 2.2 concerns the idealized unbounded-restart variant; in computation we cap the number of restarts at L and report failure if no PNE is reached. If each BR solve takes time at most $T_{\mathcal{BR}}$, then the expected time until discovery $\mathbb{E}[\text{time}] \leq \mathbb{E}[\#BR \text{ solves}] T_{\mathcal{BR}} \leq \frac{nR}{q_\mu[R]} T_{\mathcal{BR}}$. Here $T_{\mathcal{BR}}$ depends on the BR implementation and may be difficult to characterize.

RRR-BRD in IPGs. We now specialize RRR-BRD to IPGs. The scheme is analogous to Algorithm 2.1, but states are represented by integer-feasible strategy profiles $\mathbf{x} = (\mathbf{x}^1, \dots, \mathbf{x}^n)$, and BR updates are implemented by solving player-specific IPs. The round-based random permutation is encoded in Line 4, and random restarts are encoded in Line 11.

We now present the following corollary; proof is given in Appendix 2.B.

Corollary 2.5 (RRR-BRD for IPGs). *Consider an IPG $\mathcal{G} = (N, (\mathcal{X}_i)_{i \in N}, (u_i)_{i \in N})$ with polyhedrally bounded feasible sets $\mathcal{X}_i = \{\mathbf{x}^i \in \mathbb{Z}^{m_i} : A^i \mathbf{x}^i \leq b^i\}$, so that $\mathcal{X} = \prod_{i \in N} \mathcal{X}_i$ is finite and can be viewed as the state space S of a finite normal-form game. Let μ be the restart law induced by `RANDOM_GENERATION`($\mathcal{X}$) in Algorithm 2.4. If $\mathcal{G}$ admits a PNE and $q_\mu[R] > 0$, then Algorithm 2.4 with unbounded restarts ($L \rightarrow \infty$) finds a PNE with probability 1 and satisfies $\mathbb{E}[\#BR \text{ solves}] \leq \frac{nR}{q_\mu[R]}$. In particular, if `RANDOM_GENERATION` has full support on $\mathcal{X}$ and a PNE exists, then $q_\mu[R] > 0$ for every $R \geq 1$ (Remark 2.3).*

Algorithm 2.4 RRR-BRD algorithm for integer programming games

Require: IPG instance, initial strategy profile $\bar{\mathbf{x}}_{\text{init}}$, max rounds R , max restarts L

```

1: Initialize current profile  $\bar{\mathbf{x}} \leftarrow \bar{\mathbf{x}}_{\text{init}}$  (optional) or  $\bar{\mathbf{x}} \leftarrow \text{RANDOM\_GENERATION}(\mathcal{X})$ 
2: for  $l = 1$  to  $L$  do
3:   for  $r = 1$  to  $R$  do
4:     draw a random permutation of players  $N \leftarrow \text{RANDOMPERMUTATION}(N)$ 
5:     for each player  $i$  in  $N$  do
6:       If an improving best response exists, compute  $\hat{\mathbf{x}}^i \in \arg \max_{\mathbf{x}^i} u_i(\mathbf{x}^i, \bar{\mathbf{x}}^{-i})$  and update
7:        $\bar{\mathbf{x}} \leftarrow (\hat{\mathbf{x}}^i, \bar{\mathbf{x}}^{-i})$ .
8:       if consecutive profiles are identical then return  $\bar{\mathbf{x}}$   $\triangleright$  PNE certified by end-of-round
9:       no-change
10:
11:    $\bar{\mathbf{x}} \leftarrow \text{RANDOM\_GENERATION}(\mathcal{X})$   $\triangleright$  restart: generate a new initial strategy profile
12:
13: return None  $\triangleright$  no PNE found within caps  $R, L$ 

```

Any feasible-solution generator with broad coverage over $\mathcal{X}$ can be used to implement μ . We provide explicit random feasible-solution generators—both maximal and full-support variants—for KPG and EBMC in Appendix 2.C. We alternate these variants across restarts; the resulting mixture retains full support on $\mathcal{X}$. We also cap R and L , since we do not know *a priori* whether a PNE exists: if no PNE exists, the conceptual RRR-BRD would never terminate. Moreover, when integrating with a MIP solver, the solver supplies natural initial profiles, so imposing finite caps is practical.

In terms of runtime, if each BR solve takes at most $T_{\mathcal{BR}}$, then the expected runtime bound follows immediately by multiplying the BR-solve bound by $T_{\mathcal{BR}}$. In IPGs, each BR solve is implemented by solving a player-specific IP to optimality, so $T_{\mathcal{BR}}$ can be difficult to characterize theoretically. In practice, one may treat $T_{\mathcal{BR}}$ as an empirical per-solve runtime estimate under the chosen solver settings.

2.3.4 Complexity Analysis for RRR-BRD

The practical question is how to select the round cap R in RRR-BRD. As R increases, $q_\mu[R]$ (2.4) is nondecreasing, creating a trade-off in expected time. Taking R to be very large is impractical in a failed attempt, and the idealized “stop-on-trap” behavior is unavailable. Throughout this chapter, we therefore fix a modest cap $R \in [10, 20]$, which empirically balances per-attempt cost and $q_\mu[R]$.

Because the BR state graph can contain SCCs of size > 1 and non-PNE traps, performance can be poor even when a PNE exists: if $q_\mu[R]$ is extremely small, then after r restarts the total work nRr can exceed crude enumeration scales with non-negligible probability. In particular, as a *worst-case event*, more than r restarts are required (equivalently, the algorithm fails for r consecutive attempts), which occurs with probability $\Pr(R^* > r) = (1 - q_\mu[R])^r$. See Appendix 2.D for a schematic and a knapsack-style example where most restarts are absorbed by a non-PNE trap and $q_\mu[R]$ is exponentially small. As a *best-case event*, the first attempt can succeed, which occurs with probability $q_\mu[R]$. Very roughly, $q_\mu[R]$ captures instance difficulty for RRR-BRD under μ and R . Our guarantees are Las Vegas-style: success occurs almost surely when $q_\mu[R] > 0$, while efficiency is governed by these instance-dependent success probabilities.

Since $q_\mu[R]$ is not directly computable in large IPGs, we estimate it empirically via Monte Carlo simulation over restart states. This aligns with the sampling-and-simulation approach used to estimate absorption behavior on response graphs [83]. While their work defines transition via weakly improving deviations and studies infinite-horizon absorption into sink SCCs under noisy dynamics, we estimate a finite-horizon probability of hitting the PNE set within $\leq R$ rounds under improving best-response updates.

Given a restart law μ , we estimate $\tilde{q}_\mu[R] := \mathbb{E}_{s \sim \mu}[\mathbf{1}\{\text{RRR-BRD from } s \text{ reaches a PNE within } R \text{ rounds}\}]$

using Monte Carlo by averaging the indicator over a finite set of starts. We report `avg_rounds`, i.e., the mean number of rounds over all attempts (using R for failures and the realized round count for successes). Furthermore, based on $\tilde{q}_\mu[R]$, we report the expected step bound (ESB) for each instance.

We examine RRR-BRD in two settings:

- (i) **Normal-form reductions.** We consider instances of the form $\max_{\mathbf{x}^i \in \{0,1\}^m} \{\mathbf{d}_i^\top \mathbf{x}^i + \sum_{p \in N: p \neq i} (\mathbf{x}^p)^\top Q_p^i \mathbf{x}^i\}$, where $\mathbf{d}_i \in \mathbb{Z}^p$, and $Q_p^i \in \mathbb{Z}^{m \times m}$, with entries in $[-100, 100]$. For $n \in \{2, 3\}$ and $m \in \{3, 4\}$, we sample 10 random games per (n, m) , retaining only instances that admit at least one PNE. For each instance, μ is taken from a uniform distribution over all pure profiles S . Since $|S|$ is modest, we set $R = 10$ and estimate $\tilde{q}_\mu[R]$ by testing every starting state.
- (ii) **Knapsack problem games (KPG).** (see Section 2.7.4 for the full definition). We implement RRR-BRD for IPGs (Algorithm 2.4) with a round cap $R = 20$ and a restart cap $L = 100$. μ is taken from the empirical distribution over 100 randomly generated feasible starts per instance using a feasible-solution generator (Appendix 2.C); $\tilde{q}_\mu[R]$ is then estimated from these starts. We use the KPG instances from Dragotto and Scatamacchia [58] for which at least one PNE has been verified in their experiments.

The summaries for (i) and (ii) appear in Tables 2.1 and 2.2. Full results are in Appendix 2.E. In all simulated cases $\tilde{q}_\mu[R] > 0$, and the average $\tilde{q}_\mu[R]$ value often exceeds 0.5, implying high success probability with a modest number of restarts. While $|S|$ can be counted exactly for (i), the number of integer-feasible profiles for (ii) cannot be counted exactly; nevertheless, it is expected to be substantially larger than in (i) due to the many binary decisions, even under knapsack constraints. Note that the average ESB is a conservative upper bound; the realized number of best-response solves is often much smaller.

Table 2.1: Normal-form reductions: empirical success within $R = 10$ rounds under uniform restarts.

n	k	$ S $	# inst. with $\tilde{q}_\mu[R] > 0$	avg. $\tilde{q}_\mu[R]$	avg_rounds	avg_ESB ($\frac{nR}{q_\mu[R]}$)
2	3	64	10/10	1.00	2.6	20
2	4	256	10/10	0.925	3.2	26
3	3	512	10/10	0.58	6.4	94.4
3	4	4096	10/10	0.72	5.7	62.4

Table 2.2: KPGs: empirical success within $R = 20$ rounds under random restarts.

type	# inst. with $\tilde{q}_\mu[R] > 0$	avg. $\tilde{q}_\mu[R]$	avg_rounds	avg_ESB ($\frac{nR}{q_\mu[R]}$)
A	23/23	1	3.6	49.6
B	20/20	1	3.4	48
C	7/7	0.60	10.1	2135.2

Types (A) and (B) draw the interaction coefficients equally and uniformly from the set in $[1, 100]$ and independently and uniformly from the set in $[1, 100]$, respectively. Type (C) draws them from a uniform distribution over $[-100, 100]$.

This Monte Carlo exercise is used only to contextualize complexity, not as a prerequisite for implementation. Note that the estimate $\tilde{q}_\mu[R]$ is inherently *algorithm- and parameter-dependent* outside structured classes (e.g., potential games or BR-weakly acyclic games). Accordingly, we do not propose estimating $\tilde{q}_\mu[R]$ in practice; rather, we run RRR-BRD directly and treat $\tilde{q}_\mu[R]$ as an *ex post* diagnostic. In particular, even a *single* successful run that returns a PNE immediately certifies $q_\mu[R] > 0$ for that (μ, R) (and yields a PNE to report), whereas failure to hit a PNE within a finite budget does not rule out $q_\mu[R] > 0$.

2.4 Best-response Dynamics Incorporated Zero-regret Algorithm

This section develops our algorithmic framework for computing PNEs in large-scale IPGs. We first review the lifted formulation and equilibrium inequalities (EIs) used in ZR (Sec-

tion 2.4.1), and then present our best-response-incorporated variant and implementation details (Section 2.4.2). We also show how RRR-BRD can be used as a randomized local-search subroutine (Section 2.4.3) and, if it terminates without identifying any PNE, how it can be adapted to provide an approximate-equilibrium certificate (Section 2.4.4).

2.4.1 Lifted Space and Perfect Equilibrium Formulation

Following Dragotto and Scatamacchia [58], we work in a lifted space in which all players' utilities are linearized. Let $\mathbf{x} := (\mathbf{x}^1, \dots, \mathbf{x}^n)$ collect the players' decision vectors and $\mathbf{z}$ denote auxiliary variables used to linearize nonlinear terms (e.g., bilinear products). Let $\mathcal{K}$ be the lifted feasible set, $\mathcal{K} := \{(\mathbf{x}, \mathbf{z}) : \mathbf{x}^i \in \mathcal{X}_i \ \forall i \in N, (\mathbf{x}, \mathbf{z}) \text{ satisfies all linearization constraints and bounds}\}$, and let $\text{conv}(\mathcal{K})$ be its convex hull. Any integer point $(\mathbf{x}, \mathbf{z}) \in \mathcal{K}$ corresponds to a unique strategy profile $\mathbf{x}$; conversely, every pure-strategy profile has at least one lifted representative in $\mathcal{K}$.

Let $\mathcal{S}_{\text{pne}}(\mathcal{G})$ denote the set of all PNE profiles of the IPG instance $\mathcal{G}$. Define the *perfect equilibrium formulation* as $\mathcal{E} := \{(\mathbf{x}, \mathbf{z}) \in \text{conv}(\mathcal{K}) : \mathbf{x} \in \text{conv}(\mathcal{S}_{\text{pne}}(\mathcal{G}))\}$. Let $\mathcal{BR}_i(\mathbf{x}^{-i})$ denote the set of BRs of player i to a profile $\mathbf{x}^{-i} \in \prod_{k \in N, k \neq i} \mathcal{X}_k$. ZR is a cutting-plane algorithm using EIs, that is, linear inequalities valid for $\mathcal{E}$. Such inequalities are obtained from BRs: for any player i and any BR $\hat{\mathbf{x}}^i \in \mathcal{BR}_i(\mathbf{x}^{-i})$, the inequality

$$u_i(\hat{\mathbf{x}}^i, \mathbf{x}^{-i}) \leq u_i(\mathbf{x}^i, \mathbf{x}^{-i}), \quad (2.5)$$

is valid for all $(\mathbf{x}, \mathbf{z}) \in \mathcal{E}$ (with utilities evaluated in the lifted space). Dragotto and Scatamacchia [58] show that collecting all such inequalities over all players and all BRs yields an *equilibrium closure* that coincides with $\mathcal{E}$ (Theorem 1 in [58]).

Given a candidate point $(\bar{\mathbf{x}}, \bar{\mathbf{z}}) \in \text{conv}(\mathcal{K})$, the *equilibrium separation oracle* solves, for each

player i , $\hat{\mathbf{x}}^i \in \arg \max_{\mathbf{x}^i \in \mathcal{X}_i} \{u_i(\mathbf{x}^i, \bar{\mathbf{x}}^{-i})\}$. If $u_i(\hat{\mathbf{x}}^i, \bar{\mathbf{x}}^{-i}) > u_i(\bar{\mathbf{x}}^i, \bar{\mathbf{x}}^{-i})$, the associated EI (2.5) is violated at $(\bar{\mathbf{x}}, \bar{\mathbf{z}})$ and is added as a cut; if no such violation occurs for any i , then $\bar{\mathbf{x}}$ is a PNE.

The original ZR algorithm iteratively solves a master problem of the form

$$\max\{\phi(\mathbf{x}, \mathbf{z}) : (\mathbf{x}, \mathbf{z}) \in \mathcal{K}, (\mathbf{x}, \mathbf{z}) \text{ satisfies all cuts in } \Omega\}, \quad (2.6)$$

where Ω is a growing family of EIs. At each iteration, ZR solves (2.6) to optimality in the lifted space, queries the equilibrium separation oracle at the optimal solution $(\bar{\mathbf{x}}, \bar{\mathbf{z}})$, and either (i) certifies $\bar{\mathbf{x}}$ as a welfare-maximizing PNE or (ii) adds violated EIs to Ω and repeats. This yields a cutting-plane method that is provably correct and finitely terminating whenever a PNE exists. We note that regardless of the choice of the objective function in (2.6), the capability of computing a PNE via the separation procedure remains valid.

Motivation for the BZR algorithm. The BZR algorithm is motivated by the complementary strengths and weaknesses of ZR and RRR-BRD.

(i) *Limitations of ZR and how RRR-BRD complements it.* For large-scale IPGs, the ZR approach faces several practical challenges. Solving the joint optimization problem (2.6) to optimality can be very expensive when $\mathcal{X}$ encodes many players and variables. Even when the POS is small, many cuts may be needed to move from the socially optimal solution $\bar{\mathbf{x}}_{\text{sw}}^*$ to $\hat{\mathbf{x}}_{\text{pne}}^*$. Moreover, the original implementation only adds EIs for BRs that are currently violated, even though additional BRs could, in principle, be used to cut off larger portions of the non-equilibrium region. RRR-BRD (Algorithm 2.4) addresses these issues by rapidly finding PNEs at scale and by generating many BRs along its trajectories. Embedding RRR-BRD as a randomized local-search subroutine inside ZR supplies PNEs in instances where

plain ZR struggles to find any PNE within the time limit, and provides a rich pool of best responses that can be recycled as additional EIs.

(ii) *Limitations of RRR-BRD and how ZR complements it.* RRR-BRD itself has only probabilistic guarantees: it can efficiently find PNEs when they exist but cannot certify that no PNE exists. When integrated with the lifted ZR formulation, RRR-BRD benefits from diverse, welfare-oriented starting points produced by the MIP solver through callbacks, which inject both randomness and structure into the initial profiles. At the same time, the combined BZR algorithm inherits ZR’s structural capabilities: given sufficient time, ZR can certify “no PNE” by proving infeasibility of the equilibrium formulation and can be configured to enumerate PNEs (e.g., via Hamming-distance cuts). Thus, ZR supplies the certification and enumeration machinery that RRR-BRD lacks, while RRR-BRD supplies the scalable search capability that ZR lacks.

2.4.2 Algorithm Details

We now present the BZR algorithm (Algorithm 2.7) and illustrate its behavior in Figure 2.2, which depicts the joint strategy space $\mathcal{X}$ for two players. In that figure the axes are the two players’ utilities, so each marker is the utility pair induced by one joint profile: hollow markers are profiles that are not equilibria and solid markers are PNEs. The outer shaded region is $\text{Conv}(\mathcal{X})$ and the inner one is $\text{Conv}(\mathcal{X} \cap \mathcal{S}_{pne})$, so BZR searches for the welfare-maximizing point of the inner region while optimizing over a relaxation of the outer one, and the dashed gray arrow points in the direction of increasing social welfare. The BZR algorithm can be viewed as a combination of a randomized local-search subroutine and a cutting-plane method with delayed (lazy) separation of EIs.

Callback function and separation procedure: We use the MIP solver’s callback mech-

anism that triggers on integer-feasible solutions (e.g., MIPSOL in Gurobi [80]). While many types of intervention in the branch-and-bound tree are possible (e.g., custom branching rules), we restrict our use of callbacks to (i) collect integer-feasible points and (ii) inject EIs. Each incumbent integer solution $\bar{\mathbf{x}}$ returned by the callback is passed to the separation procedure (Lines 4–9). The oracle checks whether $\bar{\mathbf{x}}$ is a PNE; if not, it adds violated EIs. In Figure 2.2, these cuts are shown in green.

RRR-BRD as a randomized local-search subroutine: If $\bar{\mathbf{x}}$ is not a PNE, it is used as an initial strategy profile for RRR-BRD (Line 11). Along each RRR-BRD trajectory, we store the BRs encountered; for any BR $\hat{\mathbf{x}}^i$ whose associated EI has not yet been added, we may inject it into Ω as an additional cut (Lines 12–14). In Figure 2.2, the RRR-BRD trajectory is illustrated by the blue arrow, and these additional EIs are drawn in red. Note that this step is valid regardless of whether RRR-BRD finds a PNE in that run: every trajectory consists of valid BRs, hence all associated EIs are valid. If a newly found PNE improves the current best value, we update the lower bound and set $\hat{\mathbf{x}}_{\text{pne}}^* \leftarrow \bar{\mathbf{x}}$ (Line 18). If RRR-BRD fails to find a PNE within the caps (R, L) , no primal update occurs and the algorithm simply proceeds to the next callback solution.

Termination: The dual (upper) bound UB is updated by solving $UB = \max \{ \phi(\mathbf{x}, \mathbf{z}) \mid \mathbf{x} \in \mathcal{K}_{\text{relax}} \cap \Omega \}$, where $\mathcal{K}_{\text{relax}}$ denotes the current relaxation of $\mathcal{K}$ and Ω is the set of accumulated EIs. The BZR algorithm terminates once $UB - LB \leq \text{tol}$ (Line 20–23), where tol is the solver’s default optimality tolerance. If the gap closes, BZR certifies either that the best PNE has been found or that the equilibrium formulation is infeasible, in which case no PNE exists. If the imposed time limit is reached first, BZR returns the best PNE found, if any, together with the incumbent dual bound, but supplies neither an optimality certificate nor a certificate of nonexistence.

Remark 2.6. The set $\mathcal{K}_{\text{relax}}$ consists of the LP relaxation of $\mathcal{K}$, augmented with general-purpose cutting planes generated by the solver. In our implementation, this dual bound UB may remain above $\bar{\mathbf{x}}_{\text{sw}}^*$ because BZR does *not* require solving $\max\{\phi(\mathbf{x}, \mathbf{z}) \mid (\mathbf{x}, \mathbf{z}) \in \mathcal{K} \cap \Omega\}$ to optimality at each iteration. Since UB is shaped jointly by general cuts and a subset of all possible EIs, we do not attempt to add every available EI. Instead, we add only a limited number per callback (e.g., up to n), and only for players whose BRs have not yet been used to generate cuts.

Algorithm 2.7 Best-Response Dynamics Incorporated Zero-Regret (BZR) algorithm

Require: IPG instance, parameters max rounds R , max restarts L , tolerance tol

Ensure: Best pure Nash Equilibrium $\hat{\mathbf{x}}_{\text{pne}}^*$ or **None**

```

1: Initialize:  $LB \leftarrow -\infty$ ,  $UB \leftarrow \max\{\phi(\mathbf{x}, \mathbf{z}) \mid (\mathbf{x}, \mathbf{z}) \in \mathcal{K}_{\text{relax}}\}$ ,  $\Omega \leftarrow \emptyset$ ,  $\mathcal{H} \leftarrow \emptyset$  ( $\mathcal{H}$  stores best
   responses)
2: while  $UB - LB > \text{tol}$  and time limit not exceeded do
3:    $\bar{\mathbf{x}} \leftarrow$  incumbent integer solution from MIPSOL of  $\max\{\phi(\mathbf{x}, \mathbf{z}) \mid \mathbf{x} \in \mathcal{K} \cap \Omega\}$   $\triangleright$  Callback
   Function
4:   for  $i \in N$  do
5:     Compute a best response:  $\hat{\mathbf{x}}^i \in \arg \max_{\mathbf{x}^i \in \mathcal{X}_i} \{u_i(\mathbf{x}^i, \bar{\mathbf{x}}^{-i})\}$ 
6:     if  $u_i(\hat{\mathbf{x}}^i, \bar{\mathbf{x}}^{-i}) > u_i(\bar{\mathbf{x}}^i, \bar{\mathbf{x}}^{-i})$  then
7:       Add  $u_i(\hat{\mathbf{x}}^i, \mathbf{x}^{-i}) \leq u_i(\mathbf{x}^i, \mathbf{x}^{-i})$  to  $\Omega$  and add  $\hat{\mathbf{x}}^i$  to  $\mathcal{H}$   $\triangleright$  Add violated EIs
8:
9:
10:  if any violation is found ( $\bar{\mathbf{x}}$  is not a PNE) then
11:     $\bar{\mathbf{x}} \leftarrow \text{RRR-BRD}(\bar{\mathbf{x}}, R, L)$  (Algorithm 2.4) and store the best responses found in  $\mathcal{H}$   $\triangleright$ 
    Apply RRR-BRD
12:    for (a limited number of) best responses  $\hat{\mathbf{x}}^i \in \mathcal{H}$ ,  $i \in N$  do
13:      if the EI for  $\hat{\mathbf{x}}^i$  has not yet been added then
14:        Add  $u_i(\hat{\mathbf{x}}^i, \mathbf{x}^{-i}) \leq u_i(\mathbf{x}^i, \mathbf{x}^{-i})$  to  $\Omega$   $\triangleright$  Add additional EIs
15:
16:
17:
18:  if  $\bar{\mathbf{x}}$  is a PNE and  $\phi(\bar{\mathbf{x}}) \geq LB$  then  $\hat{\mathbf{x}}_{\text{pne}}^* \leftarrow \bar{\mathbf{x}}$  and  $LB \leftarrow \phi(\bar{\mathbf{x}})$ 
19:
20:   $UB \leftarrow \max\{\phi(\mathbf{x}, \mathbf{z}) \mid (\mathbf{x}, \mathbf{z}) \in \mathcal{K}_{\text{relax}} \cap \Omega\}$ 
21:
22: if  $\hat{\mathbf{x}}_{\text{pne}}^* \neq \text{None}$  then return  $\hat{\mathbf{x}}_{\text{pne}}^*$  else return None  $\triangleright$  Return best PNE found or No PNE
   found
23:
```

Remark 2.8. Figure 2.2 depicts many EIs, with some appearing “deeper” (e.g., those that are binding at PNEs). However, we do not establish any general dominance relation among

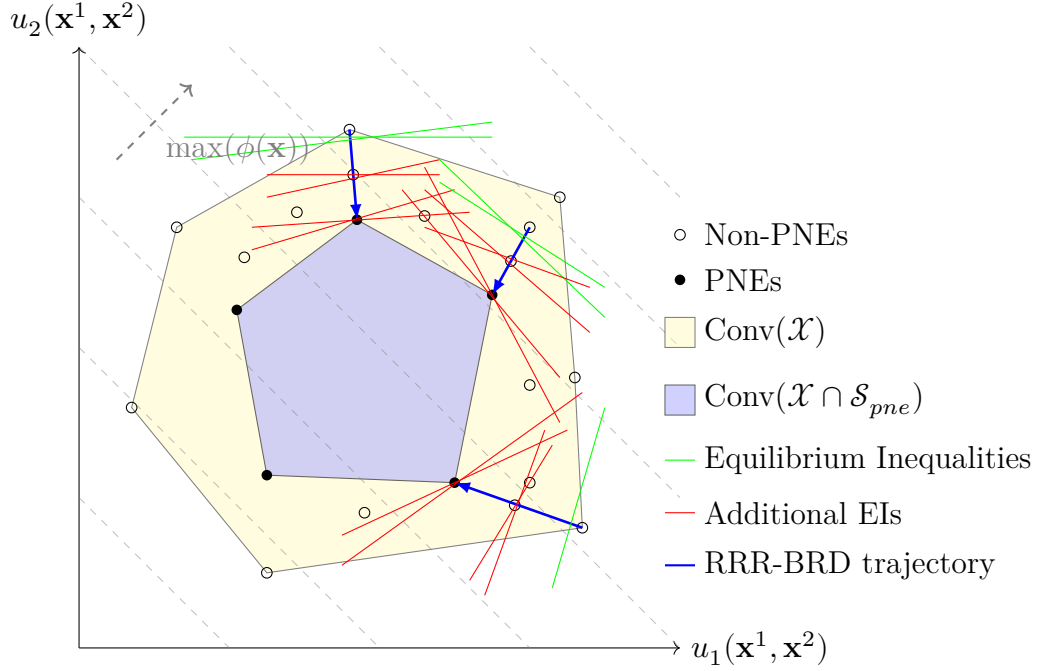


Figure 2.2: Illustration of the BZR algorithm in a two-player game.

EIs. For general IPGs, we are not aware of a principled way to identify a strictly stronger subclass of EIs. A baseline observation is that including EIs associated with *all* BRs yields the perfect-equilibrium formulation $\mathcal{E}$. Accordingly, BZR focuses on generating a practically useful subset of EIs, rather than attempting a full characterization.

An illustrative two-player example that works explicitly through the BZR computations is provided in the Appendix 2.F. To complement this small-scale walkthrough, we also include an exponential-separation example between ZR and BZR in Appendix 2.G.

2.4.3 RRR-BRD as a Randomized Local-Search Subroutine for Other Algorithms

Carvalho et al. [37] organize IPG algorithms around three building blocks: an *approx-*

mate phase, which constructs an inner or outer approximation of each player’s strategy set, yielding sets $\tilde{\mathcal{X}}_i$; a *play* phase, which computes a tentative equilibrium *with respect to* this approximated game; and an *improve* phase, which verifies the tentative solution against the original game and refines the approximation as needed. In existing methods, the play phase is typically implemented via one or more *joint* optimization problems that couple all (or a subset of) players’ decisions, either explicitly in normal form or implicitly through extended MIP and complementarity-based formulations. RRR-BRD provides a complementary play-phase mechanism: rather than solving a global joint problem over a potentially very large feasible region, it performs a randomized walk on the BR state graph, repeatedly solving BR problems without ever constructing the full normal form.

Table 2.3 summarizes the main exact algorithms for computing Nash equilibria in IPGs, adapted from Carvalho et al. [37] and extended with RRR-BRD. The first six methods are based on *deterministic* existence and correctness proofs for their targeted equilibrium notions (MNE or PNE) in the classes considered. By contrast, RRR-BRD provides a *probabilistic* (Las Vegas) guarantee. This trades deterministic worst-case guarantees for a random-walk–based mechanism that scales well in practice.

Table 2.3: Summary of algorithms for computing Nash equilibria in IPGs.

Method	NE Type	Approx	Play	Alg Feas	Largest IPG Instances Tested	Reference
SGM	Mixed	Inner	FP	Deter.	KPG: $n \leq 3, m \leq 100$	[36]
eSGM	Mixed	Inner	MIP	Deter.	KPG: $n \leq 3, m \leq 60$	[49]
CnP	Mixed	Outer	CP	Deter.	KPG: $n \leq 3, m \leq 100$	[38]
BM	Pure	Outer	CP	Deter.	qIPG: $n \leq 6, m \leq 3$	[128]
ZR	Pure	Outer	MIP	Deter.	KPG: $n \leq 3, m \leq 100$	[58]
BnP	Pure	Outer	MIP	Deter.	qIPG: $n \leq 3, m \leq 5$	[135]
RR-BRD	Pure	Inner	BRD	Prob.	KPG: $n \leq 30, m \leq 100$, EBMC: $n \leq 84$	This chapter

Note. This table is adapted from Table 1 in [37] and extended to include RR-BRD.

‘Approx’ = approximate phase, ‘Play’ = play phase. ‘Alg Feas’ indicates whether the algorithm has a *deterministic* (Deter.) or *probabilistic* (Prob.) theoretical guarantee to find its targeted equilibrium whenever one exists in the considered class of IPGs.

FP = feasibility problem, MIP = mixed-integer problem, CP = complementarity problem, KPG = knapsack problem games, qIPG = quadratic IPG. n : number of players; $m = m_i, i \in N$: number of integer variables per player.

Conceptually, RRR-BRD can be plugged into any IPG algorithm that occasionally produces an integer-feasible joint strategy profile $\mathbf{x} \in \mathcal{X}$. Once such a profile is available, RRR-BRD can be launched from $\mathbf{x}$ (on the induced finite game) to search locally for a PNE. If a PNE is found, it can be recorded as a feasible solution without altering the logical structure or exactness guarantees of the base algorithm; if not, the base algorithm simply continues unaffected. In this sense, RRR-BRD acts as a modular, PNE-oriented local-search layer that can be invoked whenever integer-feasible joint strategies become available.

Branching-type methods (BM and BnP). In the *branching method* (BM) of Sagratella [128] and the *branch-and-prune* (BnP) algorithm of Schwarze and Stein [135], the search for equilibria is organized as a tree whose nodes correspond to restricted subgames defined by additional bounds or fixings on players' variables. At each node, a continuous auxiliary problem (e.g., a relaxed NEP or complementarity system) is solved, pruning tests may discard parts of the search region, and branching refines players' feasible sets. Integer-feasible joint profiles arise, for example, from rounding, local search, or exhaustive enumeration within a node's box. Whenever such a profile is available, RRR-BRD can be run on the corresponding restricted subgame, using that profile as a starting point to search for a PNE within the node.

Sampling-based and outer-approximation methods (SGM, eSGM, CnP). In the sample generation method (SGM) [36], its enumeration variant eSGM [49], and the Cut-and-Play (CnP) algorithm [38], an IPG is approximated by a finite normal-form game built from sampled or externally generated pure strategies. At intermediate iterations, these methods maintain (i) finite strategy subsets $\tilde{S}_i \subseteq S_i$ for each player and (ii) one or more integer-feasible joint profiles in $\tilde{S} := \prod_i \tilde{S}_i$ (e.g., candidate equilibria or best-response combinations in the sampled game). RRR-BRD can then be applied to the current finite game restricted to $\tilde{S}$, using any such profile as a starting point. If RRR-BRD reaches a PNE of the sampled

game, then (by construction) this profile corresponds to a degenerate mixed Nash equilibrium of the original IPG and can be stored as a candidate solution or benchmark.

2.4.4 Approximate PNE Certificates upon Early Termination

RRR-BRD can also return an *approximate-equilibrium certificate* when it does not reach an exact PNE within the allotted budget. For any profile $\bar{\mathbf{x}}$, let $\hat{\mathbf{x}}^i \in \arg \max_{\mathbf{x}^i \in \mathcal{X}_i} u_i(\mathbf{x}^i, \bar{\mathbf{x}}^{-i})$ be a best response for player i , and assume $u_i(\mathbf{x}^i, \mathbf{x}^{-i}) > 0$ for all feasible profiles. Define

$$\alpha_i(\bar{\mathbf{x}}) := \frac{u_i(\hat{\mathbf{x}}^i, \bar{\mathbf{x}}^{-i})}{u_i(\bar{\mathbf{x}}^i, \bar{\mathbf{x}}^{-i})}, \quad \alpha(\bar{\mathbf{x}}) := \max_{i \in N} \alpha_i(\bar{\mathbf{x}}). \quad (2.7)$$

Then $\bar{\mathbf{x}}$ is an $\alpha(\bar{\mathbf{x}})$ -approximate PNE in the multiplicative sense, and $\alpha(\bar{\mathbf{x}}) = 1$ if and only if $\bar{\mathbf{x}}$ is an exact PNE. Implementation and reporting details are provided in Appendix [2.M](#).

2.5 EBMC Game Models

This section presents the edge-weighted budgeted maximum coverage (EBMC) game as a motivating example of large-scale IPGs. Section [2.5.1](#) introduces the EBMC problem, and Section [2.5.2](#) develops two EBMC games with different utility functions. Section [2.5.3](#) characterizes the role of types and network structure, and Section [2.5.4](#) connects EBMC to AIS-prevention resource allocation.

2.5.1 The Edge-weighted Budgeted Maximum Coverage (EBMC)

Problem

The budgeted maximum coverage (BMC) problem [43] aims to maximize element coverage by selecting subsets within a limited budget. The EBMC problem is a specialized case of BMC, where edges represent elements, and vertices represent subsets. Both problems are NP-hard [43], and [39] further demonstrate that EBMC remains NP-hard even when restricted to bipartite graphs.

We model the underlying network as a directed graph $(\mathcal{J}, \mathcal{A})$, where $\mathcal{J}$ is the set of vertices and $\mathcal{A}$ is the set of arcs. The players are indexed by $N := \{1, \dots, n\}$. Each player $i \in N$ controls a subset of vertices $\mathcal{J}_i \subseteq \mathcal{J}$, and these subsets form a partition of the vertex set, i.e., $\bigsqcup_{i \in N} \mathcal{J}_i = \mathcal{J}$. We write $\mathcal{A}_i \subseteq \mathcal{A}$ for the set of arcs whose endpoints both lie in $\mathcal{J}_i$, and use $\delta_+(\mathcal{J}_i)$ (resp. $\delta_-(\mathcal{J}_i)$) for the set of arcs with tail (resp. head) in $\mathcal{J}_i$ and head (resp. tail) in $\mathcal{J} \setminus \mathcal{J}_i$. Each arc $(j, k) \in \mathcal{A}$ has a nonnegative weight w_{jk} , and each player $i \in N$ has a budget $\mathcal{B}_i$. The decision variables are $x_j \in \{0, 1\}$, indicating whether vertex $j \in \mathcal{J}$ is selected, and $y_{jk} \in \{0, 1\}$, indicating whether arc $(j, k) \in \mathcal{A}$ is *covered* (i.e., at least one of its endpoints is selected).

The Social Welfare (SW) model ((2.8a) –(2.8c)) represents the centralized planning perspective. The objective function (2.8a) maximizes total edge coverage over the arc set $\mathcal{A}$. Constraint (2.8b) enforces that an edge (j, k) can be covered only if at least one of its adjacent vertices, j or k , is selected. Constraint (2.8c) imposes the individual budget limit for each player $i \in N$.

The Social Welfare (SW) Model

$$\max_{\substack{x_j \in \{0,1\} \quad \forall j \in \mathcal{J} \\ y_{jk} \in \{0,1\} \quad \forall (j,k) \in \mathcal{A}}} \sum_{(j,k) \in \mathcal{A}} w_{jk} y_{jk} \quad (2.8a)$$

$$\text{s.t.} \quad y_{jk} \leq x_j + x_k \quad \forall (j,k) \in \mathcal{A} \quad (2.8b)$$

$$\sum_{j \in \mathcal{J}_i} x_j \leq \mathcal{B}_i \quad \forall i \in N. \quad (2.8c)$$

The optimal solution to the SW model is the OSW solution $(\bar{\mathbf{x}}_{\text{sw}}^*)$, as introduced in Section [2.2](#).

2.5.2 A Non-cooperative EBMC Game Among N Players

Although the SW model in (2.8) allocates budgets to each player, it primarily represents the objectives of the central planner, not the players. Consequently, the optimal solutions of this model may not align with the individual goals of the players. We start by defining a player-specific optimization problem that is contingent on the decisions of other players. Subsequently, we introduce two types of utility functions to model player interactions: a locally altruistic function and a selfish function, based on the different sets of arcs.

For each player $i \in N$, let $\mathcal{J}_{-i}$ denote the set of vertices not controlled by player i , and let $\bar{\mathbf{x}}^{-i} \in \{0,1\}^{\mathcal{J}_{-i}}$ be the corresponding decisions of all other players. The optimization model for a single player is given by:

The Single-Player (SP) Model

$$\max \quad \tilde{u}_i(\mathbf{y}) \tag{2.9a}$$

$$\text{s.t.} \quad y_{jk} \leq x_j + x_k \quad \forall (j, k) \in \mathcal{A} \tag{2.9b}$$

$$\sum_{j \in \mathcal{J}_i} x_j \leq \mathcal{B}_i \tag{2.9c}$$

$$x_j = \bar{x}_j^{-i} \quad \forall j \in \mathcal{J}_{-i} \tag{2.9d}$$

$$x_j \in \{0, 1\} \quad \forall j \in \mathcal{J}_i \tag{2.9e}$$

$$y_{jk} \in \{0, 1\} \quad \forall (j, k) \in \mathcal{A}. \tag{2.9f}$$

The SP model (2.9), defined by (2.9a)–(2.9f), describes player i 's problem parameterized by $\bar{\mathbf{x}}^{-i}$. When an edge connects a vertex $j \in \mathcal{J}_i$ of player i to a vertex $k \in \mathcal{J}_{-i}$ of some other player, a positive decision $x_k = 1$ allows edge (j, k) to be covered even if $x_j = 0$. Thus, the actions of other players can have a positive external effect on player i 's objective. The objective function $\tilde{u}_i(\mathbf{y})$ in (2.9a) is left generic so that it can be tailored to the specific interests of player i . Since each y_{jk} can be written as $y_{jk} = \max(x_j, x_k) = x_j + x_k - x_j x_k$, the objective can equivalently be expressed as a function of $\mathbf{x}$, denoted by $u_i(\mathbf{x})$. This equivalence enables interchangeable use of $\tilde{u}_i(\mathbf{y})$ and $u_i(\mathbf{x})$ under the condition that $y_{jk} = \max(x_j, x_k)$. While using $\mathbf{y}$ -variables emphasizes edge-level outcomes influenced by other players, expressing the utility directly in terms of $\mathbf{x}$ -variables reveals player-level interactions via bilinear terms, which is standard in many IPG formulations.

Including all edges in Constraints (2.9b) and (2.9f) is computationally inefficient, especially for large graphs. Instead, arc sets can be restricted to relevant edges using the refined definition introduced below.

Definition 2.9 (Induced Arc Sets). Let $\mathcal{D} = (\mathcal{J}, \mathcal{A})$ be a directed graph and let the subset of vertices $\mathcal{J}_i, i \in N$ be a partition of $\mathcal{J}$, i.e., $\mathcal{J} := \bigsqcup_{i \in N} \mathcal{J}_i$. The *induced neighborhood arc set* $\mathcal{A}[\mathcal{J}_i]$ consists of all arcs that have at least one endpoint in $\mathcal{J}_i$. The *induced inbound arc set* $\mathcal{A}^-[\mathcal{J}_i]$ consists of all arcs that have their terminal points in $\mathcal{J}_i$. Formally, $\mathcal{A}[\mathcal{J}_i] := \{(j, k) \in \mathcal{A}_i \cup \delta_-(\mathcal{J}_i) \cup \delta_+(\mathcal{J}_i)\}$ and $\mathcal{A}^-[\mathcal{J}_i] := \{(j, k) \in \mathcal{A}_i \cup \delta_-(\mathcal{J}_i)\}$.

Regardless of the choice of induced arc sets in Definition 2.9, the joint constraints of the SP model (2.9)–interpreted by treating the parameterized variables $\bar{\mathbf{x}}^{-i}$ as decision variables in Constraint (2.9d)–correspond to those of the SW model (2.8b)–(2.8c). Based on these arc-set definitions, we consider two variants of the EBMC game, using different utility functions : (i) *locally altruistic* utilities, where player i maximizes edge coverage over $\mathcal{A}[\mathcal{J}_i]$, i.e., all edges incident to $\mathcal{J}_i$ (including outgoing arcs); and (ii) *selfish* utilities, where player i maximizes edge coverage over $\mathcal{A}^-[\mathcal{J}_i]$, i.e., only edges whose head lies in $\mathcal{J}_i$ (incoming arcs). Interpreting an edge (j, k) as a risky or critical edge (e.g., for environmental risk, security, or regulatory responsibility), Case (i) models a player who shares responsibility for any risk connected to their nodes, whereas Case (ii) models a player who is concerned only with risks incident to its own vertices $\mathcal{J}_i$.

Locally Altruistic Game In this game, the utility function of the player i is the coverage of the edges using the induced neighborhood arcs. Each player i solves the following optimization problem:

Locally Altruistic Single-Player (LASP) Model

$$\max \quad \tilde{u}_i^{\text{Alt}}(\mathbf{y}) := \sum_{(j,k) \in \mathcal{A}[\mathcal{J}_i]} w_{jk} y_{jk} \quad (2.10a)$$

$$\text{s.t.} \quad y_{jk} \leq x_j + x_k \quad \forall (j,k) \in \mathcal{A}[\mathcal{J}_i] \quad (2.10b)$$

$$(2.9c) - (2.9e) \quad (2.10c)$$

$$y_{jk} \in \{0, 1\} \quad \forall (j,k) \in \mathcal{A}[\mathcal{J}_i]. \quad (2.10d)$$

Note that the arc sets $\mathcal{A}$ in Constraints (2.9b) and (2.9f) are replaced by $\mathcal{A}[\mathcal{J}_i]$. The locally altruistic game, denoted by $\mathcal{G}^{\text{Alt}}$, is an EBMC game with N players, where each player solves a LASP model (2.10), defined by (2.10a)–(2.10d). The locally altruistic game is not fully altruistic: player i 's utility depends only on arcs incident to $\mathcal{J}_i$ not on the entire network.

Selfish Game In this game, the utility function of player i is the edge coverage of the induced inbound arc set, which contains incoming arcs controlled by player i . Then, each player i solves the following problem:

Selfish Single-Player (SSP) Model

$$\max \quad \tilde{u}_i^{\text{Self}}(\mathbf{y}) := \sum_{(j,k) \in \mathcal{A}^-[\mathcal{J}_i]} w_{jk} y_{jk} \quad (2.11a)$$

$$\text{s.t.} \quad y_{jk} \leq x_j + x_k \quad \forall (j,k) \in \mathcal{A}^-[\mathcal{J}_i] \quad (2.11b)$$

$$(2.9c) - (2.9e) \quad (2.11c)$$

$$y_{jk} \in \{0, 1\} \quad \forall (j,k) \in \mathcal{A}^-[\mathcal{J}_i]. \quad (2.11d)$$

Note that the arc sets $\mathcal{A}$ in Constraints (2.9b) and (2.9f) are replaced by $\mathcal{A}^-[\mathcal{J}_i]$. The selfish game, denoted by $\mathcal{G}^{\text{Self}}$, is an EBMC game with N players, where each player solves their

SSP model (2.11), defined by (2.11a)–(2.11d). We provide an illustrative two-player example of selfish EBMC interactions and the resulting BRD trajectory in the Appendix 2.H.

Selfish Non-Game In this scenario, each player i solves the SSP model (2.11) independently, without engaging in any strategic interaction—that is, each player assumes that no other players will select any vertices. Consequently, player i determines its strategy by solving $\bar{\mathbf{x}}_{\text{ng}}^i \in \arg \max_{\mathbf{x}^i \in \{0,1\}^{\mathcal{J}_i}} u_i(\mathbf{x}^i, \mathbf{0}^{-i})$, where $\mathbf{0}^{-i}$ denotes the zero vector for all players other than i . The combination of these individually optimal strategies forms the *selfish non-game* strategy profile, denoted by $\bar{\mathbf{x}}_{\text{ng}}$.

Equilibrium Inequality for Selfish EBMC Games We will show in Section 2.6.1 that the OSW solution $\bar{\mathbf{x}}_{\text{sw}}^*$ is a PNE for locally altruistic games. Thus, $\bar{\mathbf{x}}_{\text{sw}}^*$ will be treated as the best PNE in this setting, and no further algorithmic comparisons are needed. In contrast, for selfish games, we derive the following equilibrium inequality (EI). For any player $i \in N$ and any best response $\hat{\mathbf{x}}^i$ to a profile $\bar{\mathbf{x}}^{-i}$, we have

$$\sum_{(j,k) \in \mathcal{A}_i} w_{jk} (\hat{x}_j + \hat{x}_k - \hat{x}_j \hat{x}_k) + \sum_{(j,k) \in \delta_-(\mathcal{J}_i)} w_{jk} (x_j + \hat{x}_k - x_j \hat{x}_k) \leq \sum_{(j,k) \in \mathcal{A}_i \cup \delta_-(\mathcal{J}_i)} w_{jk} y_{jk},$$

for all $x_j, x_k, y_{jk} \in \{0,1\}$ satisfying Constraints (2.11b)–(2.11d). Note that only the best-response variables $\hat{x}_j$ with $j \in \mathcal{J}_i$ can be fixed in the EI. Moreover, in EBMC the $\mathbf{y}$ -variables serve as auxiliary variables in the lifted formulation, so when constructing EIs we can substitute $y_{jk} = x_j + x_k - x_j x_k$ and fix the appropriate $\mathbf{x}$ -variables as implied by the relevant arc-set definitions.

Explicit Reformulation to Standard IPGs The objective functions of the locally altruistic and selfish games can be reformulated into the standard IPG form by substituting

$\mathbf{y}$ -variables with $\mathbf{x}$ -variables. These reformulations are detailed in Appendix 2.I.

2.5.3 Types and Network Structure in EBMC

We enrich the EBMC formulation with a finite set of *types* $\mathcal{L}$, which may represent invasive species, commodity classes, or risk categories. For each vertex $j \in \mathcal{J}$ and type $l \in \mathcal{L}$, let $f_{jl} \in \{0, 1\}$ indicate whether j has type l , and let $n_{jk} \geq 0$ denote the interaction intensity from j to k (e.g., traffic or flow). An edge (j, k) is type- l risky if $f_{jl} = 1$ and $f_{kl} = 0$. We aggregate risk into the edge weight $w_{jk} := \sum_{l \in \mathcal{L}} f_{jl} (1 - f_{kl}) n_{jk}$, so only interactions that may transmit at least one type from j to k contribute to w_{jk} .

This structure induces a useful graph-theoretic dichotomy. When $|\mathcal{L}| = 1$, vertices split into those with and without the unique type, and risky edges run only from the former to the latter, yielding a bipartite network. When $|\mathcal{L}| \geq 2$, vertices can differ across types, and both (j, k) and (k, j) may be risky (for different types), so the network is generally non-bipartite.

2.5.4 A Resource Allocation Game among Minnesota Counties

The EBMC games in Section 2.5.2 are motivated by aquatic invasive species (AIS) prevention in Minnesota (MN). Currently, 84 county planners manage AIS inspection programs independently, each with its own AIS prevention budget supported by direct state aid [57]. The Minnesota Department of Natural Resources (MN DNR) facilitates information sharing (e.g., workshops and tools such as AIS Explorer [99]) but does not enforce a formal inter-county coordination mechanism for placing inspection stations. Counties report inspection activity (e.g., hours) to MN DNR, which can be used to infer deployed stations, and the increasing use of AIS Explorer facilitates transparency about allocations.

In this chapter, we therefore model the *current* practice as a non-cooperative game with complete information, where each county chooses how to allocate its own inspection stations and positive externalities across counties arise endogenously from these choices. The statewide goal of the MN DNR is captured by the social-welfare objective (total inspected risky boat movements). Designing regulatory interventions or incentive schemes that better align county-level decisions with statewide welfare is an important direction for future work but lies beyond the scope of this chapter. Instead, we focus on two utility specifications: a *selfish* utility, where counties care only about risky boats entering lakes in their own $\mathcal{J}_i$, and a *locally altruistic* utility, where they also value inspection of boats leaving their lakes (Section 2.5.2). Our central questions are: *given these utility specifications and network types, when do PNE solutions exist in the resulting EBMC games, how do the resulting PNE solutions compare—both theoretically and computationally—for both the selfish and locally altruistic formulations, and what social welfare levels do these PNE solutions achieve?*

The vertices $j \in \mathcal{J}$ represent lakes, partitioned by county so that $\mathcal{J}_i$ denotes the lakes controlled by county $i \in N$. For each ordered pair of lakes (j, k) , we let n_{jk} denote the estimated number of boat trips from j to k over a given period, obtained from survey-based lake-to-lake movement data [90]. Since Minnesota lakes host different AIS types, the edge weight w_{jk} aggregates type-specific risk following the setup in Section 2.5.3. Following the empirical AIS literature [81, 82], we focus on lake-to-lake risk driven by the *most recent* trip on a given route; in particular, we do not model spread along multi-edge boat routes, both because current data and practice are organized at the origin–destination level and because infestation risk is driven primarily by short-term carryover from the immediately visited infested lake. Each county i has a budget $\mathcal{B}_i$ representing the maximum number of inspection stations it can deploy, under the assumption that one station covers one lake.

2.6 Pure Nash Equilibrium for EBMC Games

As discussed in Section 2.2, the existence of a PNE is not guaranteed for general IPGs. In particular, when the interaction matrices Q_p^i in (2.1) are dense, a player's utility is heavily influenced by externalities (i.e., the decisions of other players), making it unlikely that there are simple, general sufficient conditions for PNE existence. However, if an IPG is a potential game [105], then a PNE is guaranteed to exist. Building on this idea, we first show that locally altruistic EBMC games admit a PNE. We then establish sufficient conditions for the existence of a PNE in selfish EBMC games, with a particular focus on large-scale IPG settings where externalities tend to dominate internal utility contributions.

We define the objective function of the SW model (2.8a) as the global function $\tilde{\phi}(\mathbf{y}) = \sum_{(j,k) \in \mathcal{A}} w_{jk} y_{jk} = \sum_{(j,k) \in \mathcal{A}} w_{jk} \max(x_j, x_k) = \sum_{(j,k) \in \mathcal{A}} w_{jk} (x_j + x_k - x_j x_k) = \phi(\mathbf{x})$. The next proposition shows that $\phi(\mathbf{x})$ coincides with the sum of selfish utilities $u_i^{\text{Self}}(\mathbf{x})$ as defined in (2.11a) where $\tilde{u}_i^{\text{Self}}(\mathbf{y})$ is rewritten as $u_i^{\text{Self}}(\mathbf{x})$.

Proposition 2.10 (Social welfare as sum of selfish utilities). *Let $\mathcal{D} = (\mathcal{J}, \mathcal{A})$ be a directed graph and let the subsets of vertices $\mathcal{J}_i$, $i \in N$, form a partition of $\mathcal{J}$, i.e., $\mathcal{J} := \bigsqcup_{i \in N} \mathcal{J}_i$. Then $\mathcal{A} = \bigsqcup_{i \in N} \mathcal{A}^-[\mathcal{J}_i]$, and, for any feasible $(\mathbf{x}, \mathbf{y})$ with $y_{jk} = \max(x_j, x_k)$, the social welfare objective satisfies $\phi(\mathbf{x}) = \sum_{i \in N} u_i^{\text{Self}}(\mathbf{x})$.*

The proof is given in Appendix 2.B. Note that $\phi(\mathbf{x})$ is not equal to the sum of players' utilities in the locally altruistic game. Thus, under the standard definition of social welfare (SW) as the sum of utilities, $\phi(\mathbf{x})$ coincides with SW only for the selfish game. Nevertheless, since $\phi(\mathbf{x})$ maximizes edge coverage over the entire network, we refer to it as the SW function throughout, including in the locally altruistic setting.

2.6.1 Existence of a PNE in the Locally Altruistic Games

We now show that locally altruistic EBMC games are exact potential games. A game is an *exact potential game* if there exists a function $\psi : \mathcal{X} \rightarrow \mathbb{R}$ (an *exact potential*) such that for all $i \in N$, any strategies $\bar{\mathbf{x}}^{-i} \in \mathcal{X}_{-i}$ and $\bar{\mathbf{x}}^i, \hat{\mathbf{x}}^i \in \mathcal{X}_i$ satisfy $\psi(\hat{\mathbf{x}}^i, \bar{\mathbf{x}}^{-i}) - \psi(\bar{\mathbf{x}}^i, \bar{\mathbf{x}}^{-i}) = u_i(\hat{\mathbf{x}}^i, \bar{\mathbf{x}}^{-i}) - u_i(\bar{\mathbf{x}}^i, \bar{\mathbf{x}}^{-i})$. Monderer and Shapley [105] show that every potential game possesses at least one PNE: any strategy profile that maximizes the potential function. This existence result assumes that each player's strategy set $\mathcal{X}_i$ is nonempty and compact. In our setting, each $\mathcal{X}_i$ is a finite set of binary-feasible solutions (hence compact), and it is nonempty because the all-zero strategy is feasible. We now establish the following results; the proofs are given in Appendix 2.B.

Theorem 2.11. *The social welfare function ϕ is an exact potential function for locally altruistic games.*

Corollary 2.12. *The strategy profile $\bar{\mathbf{x}}_{sw}^*$ is a PNE for locally altruistic EBMC games.*

This corollary allows central planners to recommend an optimal strategy profile $\bar{\mathbf{x}}_{sw}^*$ if all players are locally altruistic. Additionally, assuming that players have not decided on a utility function, central planners can encourage players to adopt a more altruistic approach by highlighting mutual benefits. Decision makers at any level, central planner or player level, can directly solve the SW model (2.8) and implement the solution.

2.6.2 Existence of a PNE in the Selfish Games

Next, we establish sufficient conditions for the existence of a PNE in selfish EBMC games, particularly in regimes where externalities from other players' decisions $\bar{\mathbf{x}}^{-i}$ dominate the

utility generated by a player's own choices $\mathbf{x}^i$. We denote the selfish game with a single type by $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$ and with multiple types by $\mathcal{G}_{|\mathcal{L}|\geq 2}^{\text{Self}}$.

Let $\mathcal{J} = \mathcal{J}^+ \cup \mathcal{J}^0$, where $\mathcal{J}^+ := \{j \in \mathcal{J} : \exists l \in \mathcal{L} \text{ s.t. } f_{jl} = 1\}$ and $\mathcal{J}^0 := \{j \in \mathcal{J} : f_{jl} = 0 \text{ for all } l \in \mathcal{L}\}$. Thus, $\mathcal{J}^+$ is the set of vertices that have at least one type, while $\mathcal{J}^0$ is the set of vertices with no types. The following result formalizes these conditions; its proof is provided in Appendix 2.B.

Theorem 2.13. *Let $\mathcal{J}^{ng}$ be the set of vertices selected in the non-game strategy profile $\bar{\mathbf{x}}_{ng}$. If $\mathcal{J}^{ng} \subseteq \mathcal{J}^0$, then $\bar{\mathbf{x}}_{ng}$ is a PNE for both $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$ and $\mathcal{G}_{|\mathcal{L}|\geq 2}^{\text{Self}}$. Additionally, if $\mathcal{J}^{ng} \subseteq \mathcal{J}^+$, $\bar{\mathbf{x}}_{ng}$ is a PNE for $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$.*

We now examine conditions aligned with Theorem 2.13. For a vertex $j \in \mathcal{J}_i$, let $w(j)$ denote the total weight of inbound arcs incident to j within the induced inbound arc set $\mathcal{A}^-[\mathcal{J}_i]$. In the random graphs we generated, the condition $\mathcal{J}_i^{\text{ng}} \subseteq \mathcal{J}_i^0$ for $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$ frequently holds. This phenomenon arises from the structure of $\mathcal{A}^-[\mathcal{J}_i]$, where $w(k)$ for vertices $k \in \mathcal{J}_i^0$ often exceeds $w(j)$ for vertices $j \in \mathcal{J}_i^+$. As the number of players increases, the aggregate weight from incoming arcs typically exceeds that from internal arcs, leading to a non-game strategy profile $\bar{\mathbf{x}}_{ng}^i$ that favors vertices in $\mathcal{J}_i^0$.

Even when $\mathcal{J}_i^{\text{ng}} \subseteq \mathcal{J}_i^0$ is not strictly satisfied, we observe computationally (in randomly generated instances) that $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$ almost always admits a PNE. Intuitively, large (though not strictly dominant) externalities can guide BRD to a PNE in only a few iterations. Typically, $\mathcal{J}_i^{\text{ng}}$ consists primarily of vertices in $\mathcal{J}_i^0$ due to the high volume of risky incoming arcs. If BRD starts from a conservative profile such as $\bar{\mathbf{x}}_{ng}$, players benefit iteratively: selecting vertices in $\mathcal{J}_i^+$ induces other players to select previously uncovered vertices in $\mathcal{J}_i^0$. Importantly, selecting vertices in $\mathcal{J}_i^0$ benefits only player i , whereas selecting vertices in $\mathcal{J}_i^+$ benefits others. Since $\mathcal{J}_i^+$ is finite, this improvement process eventually terminates, yielding a PNE. In Section 2.7.2,

we confirm that all instances of the selfish game $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$ admit a PNE.

However, this dynamic pattern typically holds under symmetric or balanced edge benefits. We obtain the counterexamples in Appendix 2.J either via carefully calibrated edge weights and graph structures (for $|\mathcal{L}| = 1$) or via multiple AIS types with conflicting priorities (for $|\mathcal{L}| \geq 2$).

Summary and pointers. Section 2.6 establishes PNE existence for locally altruistic EBMC games via an exact potential and provides sufficient conditions (and counterexamples) for the selfish games. For a brief discussion of the complexity of deciding PNE existence, see Appendix 2.I.

2.7 Computational Results

This section evaluates the effectiveness of RRR-BRD (Algorithm 2.4), BZR (Algorithm 2.7), and the ZR algorithm of Dragotto and Scatamacchia [58] for computing PNEs and identifying high-welfare (best-found) PNEs in EBMC games and KPGs. To implement ZR, we use Algorithm 2.7 with Lines 10–14 disabled. The main manuscript reports (i) summary results for RRR-BRD under multiple initial profiles and (ii) summary results and selected comparisons of BZR versus ZR for EBMC and KPG instances; complete results are provided in Appendix 2.K–2.L. For instances where no PNE is found across all runs, we report approximate-PNE certificates in Appendix 2.M. Section 2.7.1 describes the computational settings and performance metrics. Section 2.7.2 reports results for EBMC games on randomly generated instances with varying numbers of players ($n = 2, 3, 5, 8, 10, 15, 20, 25, 30$). Section 2.7.3 presents results for the Minnesota dataset with 84 county-level players and discusses managerial insights. Finally, Section 2.7.4 reports results for KPGs using the

instance-generation scheme of Dragotto and Scatamacchia [58], scaling from small ($n = 2, 3$) to larger games ($n = 5, 8, 10, 15, 20, 25, 30$).

2.7.1 Computational Setup and Metrics

All experiments were run on a machine with an Intel(R) Core(TM) i9-13900F CPU @ 3.10GHz (24 cores) and 64GB RAM. All algorithms (RRR-BRD, BZR, ZR) were implemented in Python and solved using Gurobi Optimizer (v11.0.3) with 16 threads. Equilibrium inequalities were enforced via Gurobi callbacks and lazy constraints. Code, instances, and results are available on GitHub.¹

Model construction time for best-response problems is non-negligible, especially in large-scale IPGs where best responses must be solved repeatedly across players. To reduce overhead, we construct each player’s best-response model once and then update only the components that change across solves (e.g., fixing $\mathbf{x}^{-i}$), consistently across all algorithms. When integrating RRR-BRD trajectories into callbacks (e.g., within BZR), we also control the number of additional best-response inequalities generated from a trajectory; in most instances, we add at most n such inequalities. For the Minnesota instance, we disable these additional cuts, since adding many inequalities can substantially slow the overall solving process.

Table 2.4 summarizes the acronyms and shorthands used in this chapter and in this section. While OSW and POS are defined in Section 2.2, we report time-limited approximations—denoted $\widetilde{\text{OSW}}$ and $\widetilde{\text{POS}}$ —due to the scale of the instances. Both quantities are empirical: $\widetilde{\text{OSW}}$ is the best social-welfare incumbent returned by the SW model within its time budget, hence a lower bound on the true optimal social welfare, and $\widetilde{\text{POS}}$ is the corresponding empirical welfare ratio rather than a certified price of stability. In particular, $\widetilde{\text{POS}}$ can fall

¹<https://github.com/HyunwooLee0429/Best-response-dynamics-IPG>

below 1 on the largest instances, where the welfare solver’s incumbent is weaker than the best PNE found by BZR; such a value indicates an unconverged welfare solve, not a PNE that outperforms the social optimum. For both BZR and ZR, the time limit was 1800 seconds. The SW model (2.8) was also solved with a 1800-second limit to compute $\phi(\tilde{\mathbf{x}}_{\text{sw}}^{1800})$. For RRR-BRD, we use two initial strategy profiles: $\mathbf{0}^N$ and $\tilde{\mathbf{x}}_{\text{sw}}^{300}$. For standalone RRR-BRD, we set the maximum numbers of restarts and rounds to $L = 10$ and $R = 20$, respectively. When used as a subroutine within BZR, we use $L = 3$ and $R = 20$. Finally, all objective values (ϕ) are reported in thousands and rounded to the nearest tenth in all tables.

Table 2.4: Acronyms and shorthands

Symbol / Abbreviation	Full Term / Meaning
$\mathbf{0}^N$	Initial strategy profile: all-zero vector
$\tilde{\mathbf{x}}_{\text{sw}}^{\text{time}}$	Initial strategy profile obtained by solving the SW model with time limit (secs)
$\bar{\mathbf{x}}_{\text{ng}}$	Non-game strategy profile
$\hat{\mathbf{x}}_{\text{pne}}^1 / \hat{\mathbf{x}}_{\text{pne}}^*$	First identified PNE / Best identified PNE (maximizing ϕ) within time limit
$\phi(\cdot)$	Social welfare function; values are reported in thousands and rounded to the nearest tenth.
$\widetilde{\text{OSW}}$	Approximate optimal social welfare ($\phi(\tilde{\mathbf{x}}_{\text{sw}}^{\text{time}})$)
$\widetilde{\text{POS}}$	Approximated price of stability given by $\phi(\tilde{\mathbf{x}}_{\text{sw}}^{\text{time}})/\phi(\hat{\mathbf{x}}_{\text{pne}}^*)$
$\#_{\text{cuts}}(\text{ZR}) / \#_{\text{cuts}}(\text{BZR})$	Number of equilibrium inequalities added by ZR / BZR
$\#_{\text{pne}}(\text{ZR}) / \#_{\text{pne}}(\text{BZR})$	Number of distinct PNEs found by ZR / BZR
$T(\text{ZR}^{1\text{st}}) / T(\text{BZR}^{1\text{st}})$	Time to find the first PNE using ZR / BZR
$T(\text{ZR}) / T(\text{BZR})$	Total runtime of the ZR / BZR algorithm

We report the number of players (n), the budget ratio (BG), and additional performance metrics.

- **RRR-BRD metrics:** For each instance and each initial profile ($\mathbf{0}^N$ and $\tilde{\mathbf{x}}_{\text{sw}}^{300}$), we report the best-found PNE (in terms of ϕ), runtime, number of rounds, $\widetilde{\text{OSW}}$, and $\widetilde{\text{POS}}$ (computed using the higher-welfare PNE among the two runs). We also include reference values from BZR, such as the best PNE found and the best dual bound. The

time to obtain $\tilde{\mathbf{x}}_{\text{sw}}^{300}$ is included in the reported runtime of RRR-BRD.

- **ZR and BZR metrics:** For each instance, we report the number of distinct PNEs found ($\#_{\text{pne}(\text{ZR})}$, $\#_{\text{pne}(\text{BZR})}$), the best-found PNE, the dual bound, time to the first PNE (e.g., $T(\text{BZR}^{1\text{st}})$), total runtime (e.g., $T(\text{BZR})$), $\widetilde{\text{OSW}}$, $\widetilde{\text{POS}}$ (based on the best-found PNE), and the number of equilibrium inequalities added (e.g., $\#_{\text{cuts}(\text{BZR})}$). As noted in Remark 2.6, the gap between the best-found PNE and the dual bound can remain large at termination, reflecting the large-scale nature of IPG instances. The best-found PNE is certified as welfare-optimal when $\phi(\hat{\mathbf{x}}_{\text{pne}}^*)$ coincides with the dual bound (within a tolerance).

2.7.2 EBMC Games with Random Dataset

We generate random EBMC instances by varying the number of counties, budget levels, infestation patterns, and network density. The number of counties is chosen from $\{2, 3, 5, 8, 10, 15, 20, 25, 30\}$, and each county controls 50 lakes. For the single-AIS case ($|\mathcal{L}| = 1$), each lake is independently infested with a probability drawn from $\{0.2, 0.4, 0.6, 0.8, 1.0\}$. For the multiple-AIS case ($|\mathcal{L}| \geq 2$), infestation is generated independently by AIS type, with type-specific probabilities drawn from nested candidate sets: $\{0.2, 0.4, 0.6, 0.8, 1.0\}$ (type 1), $\{0.2, 0.4, 0.6, 0.8\}$ (type 2), $\{0.2, 0.4, 0.6\}$ (type 3), and $\{0.2, 0.4\}$ (type 4); this construction biases instances so that most lakes remain uninfested for at least one AIS type. We consider three budget ratios, $\{0.3, 0.5, 0.8\}$, and set each county's budget as the number of infested lakes in that county multiplied by the chosen ratio. Arc weights are drawn i.i.d. uniformly from $[10, 20]$. To avoid unrealistically dense (nearly complete) networks, we start from a complete directed graph and randomly delete arcs: 20% when $|\mathcal{L}| = 1$ and 50% when $|\mathcal{L}| \geq 2$, with the higher deletion rate also helping control instance size. Overall, this

procedure yields 27 instances per case (single AIS and multiple AIS), corresponding to the 9 county counts combined with the 3 budget ratios.

Table 2.5 reports standalone RRR-BRD performance over these 27 instances using two initial profiles, $\mathbf{0}^N$ and $\tilde{\mathbf{x}}_{\text{sw}}^{300}$; complete results are provided in Appendix 2.K. Relative to the single-AIS game $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$, the multiple-AIS game $\mathcal{G}_{|\mathcal{L}|\geq 2}^{\text{Self}}$ requires substantially more rounds on average, suggesting that strategic dynamics become markedly more complex when multiple AIS types are present.

Table 2.5: Summarized RRR-BRD results with different initial strategy profiles for $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$ and $\mathcal{G}_{|\mathcal{L}|\geq 2}^{\text{Self}}$.

type $\mathcal{L}$	Avg [†] PNE (ϕ)		Avg Price of Stability		Avg References		Avg Time (secs)		Avg Round	
	init: $\mathbf{0}^N$	init: $\tilde{\mathbf{x}}_{\text{sw}}^{300}$	$\widetilde{\text{OSW}}$	$\widetilde{\text{POS}}$	BZR(ϕ)	BZR(Bound)	init: $\mathbf{0}^N$	init: $\tilde{\mathbf{x}}_{\text{sw}}^{300}$	init: $\mathbf{0}^N$	init: $\tilde{\mathbf{x}}_{\text{sw}}^{300}$
$ \mathcal{L} = 1$	786.8	1010.5	1055.5	1.051	1023.6	1089.5	7.5	212.1	2.0	1.6
$ \mathcal{L} \geq 2$	3658.0	3658.0	4010.0	1.097	3658.0	5409.3	25.6	289.5	4.0	4.7

Note. †: Averages of ϕ , OSW, and POS are calculated from instances where a PNE is found by both initial strategy profiles.

Table 2.6 summarizes the performance of BZR and ZR on $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$ and $\mathcal{G}_{|\mathcal{L}|\geq 2}^{\text{Self}}$. We report (out of 27 instances) how often a PNE was found, multiple PNEs were identified, the best PNE was obtained, and a tighter dual bound was achieved. “Ties” indicate the number of instances in which both algorithms either identify the same best PNE or attain the same dual bound. The full results for $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$ appear in Table 2.7, while the full results for $\mathcal{G}_{|\mathcal{L}|\geq 2}^{\text{Self}}$ are reported in Appendix 2.L. In both settings, BZR consistently outperforms ZR in (i) time to the first PNE, (ii) the number of distinct PNEs found, and (iii) the best PNE value identified.

For the single-AIS case ($|\mathcal{L}| = 1$), BZR typically finds a PNE within a few seconds when $n \leq 10$, and within roughly 100 seconds for $15 \leq n \leq 30$, indicating strong scalability. In contrast, ZR returns a PNE in only 11 of the 27 instances within the time limit; when both algorithms find a PNE, BZR more frequently identifies the best PNE. BZR also tends to

Table 2.6: Summarized comparison of BZR and ZR for $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$ and $\mathcal{G}_{|\mathcal{L}|\geq 2}^{\text{Self}}$.

type $\mathcal{L}$	# PNEs Found		Finding a PNE		Best PNE			Dual Bounds			Avg POS	# Avg EI Cuts	
	# _{pne(ZR)}	# _{pne(BZR)}	ZR	BZR	Ties	ZR	BZR	Ties	ZR	BZR	$\widetilde{\text{POS}}$	# _{cuts(ZR)}	# _{cuts(BZR)}
$ \mathcal{L} = 1$	32	150	11	27	10	0	17	9	4	14	1.033	59.3	94.4
$ \mathcal{L} \geq 2$	3	29	3	23	3	0	20	5	18	4	1.097	66.0	116.3

provide tighter dual bounds in this setting.

For the multiple-AIS case ($|\mathcal{L}| \geq 2$), the performance gap widens: BZR finds a PNE in 23 of the 27 instances, whereas ZR succeeds in only 3 instances within the time limit. Although BZR continues to identify high-quality PNEs, its dual bounds are often not substantially tightened, even after adding additional equilibrium inequalities. This behavior is expected for three reasons: (i) the time spent in RRR-BRD can be significant in the hardest instances, (ii) dual progress is driven largely by the LP relaxation and general-purpose cutting planes, and (iii) the added equilibrium inequalities primarily enforce equilibrium feasibility rather than strongly strengthening the continuous relaxation. Since the primary goal of ZR/BZR is to compute one or more PNEs (and, when possible, a best PNE), we treat dual bounds as auxiliary metrics.

2.7.3 EBMC Games with Minnesota Dataset

Tables 2.8 and 2.9 report results for the selfish EBMC game $\mathcal{G}_{|\mathcal{L}|\geq 2}^{\text{Self}}$ on the Minnesota dataset: Table 2.8 summarizes standalone RRR-BRD performance, and Table 2.9 compares BZR and ZR. The dataset, provided by the Minnesota Aquatic Invasive Species Research Center (MAISRC), contains lake-to-lake boat movement estimates among 9,182 lakes and uses 2018 infestation records, following the construction in Kao et al. [90]. We exclude counties with no lakes or with no incoming or outgoing risky movements.

To reflect the scale of county-level inspection programs, we model statewide support cor-

Table 2.7: Comparison of BZR and ZR for selfish EBMC games $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$.

n	BG	# PNEs Found		Best PNE (ϕ)		Dual Bounds		Price of Stability		Time (secs)				# EI Cuts	
		# _{pne(ZR)}	# _{pne(BZR)}	ZR	BZR	ZR	BZR	$\overline{\text{OSW}}$	$\overline{\text{POS}}$	T(ZR ^{1st})	T(ZR)	T(BZR ^{1st})	T(BZR)	# _{cuts(ZR)}	# _{cuts(BZR)}
2	0.3	1	1	19.2	19.2	19.2	19.2	19.2	1.000	0.2	0.2	0.2	0.4	8	13
	0.5	6	7	22.3	22.3	22.3	22.3	22.3	1.000	0.9	1.7	0.2	2.2	11	25
	0.8	1	6	23.3	23.3	23.3	23.3	24.9	1.066	1.7	1.7	0.3	2.0	12	26
3	0.3	1	6	27.6	27.6	27.6	27.6	27.6	1.000	2.4	2.4	0.6	3.4	9	19
	0.5	5	4	42.1	42.1	42.1	42.1	42.1	1.000	8.2	11.1	0.5	12.1	15	27
	0.8	1	1	53.1	53.1	53.1	53.1	53.1	1.000	0.6	0.6	0.5	1.0	12	19
5	0.3	1	8	90.4	90.4	90.4	90.4	90.4	1.000	12.9	13.1	1.3	15.0	17	38
	0.5	1	4	100.2	100.6	101.3	101.2	100.6	1.000	1474.5	1800.5	1.9	1800.3	94	45
	0.8	10	25	156.7	156.7	157.9	157.4	156.7	1.000	1134.4	1800.9	1.5	1800.3	182	58
8	0.3	3	5	134.1	134.1	134.2	134.2	149.9	1.118	40.4	74.5	6.0	113.1	41	68
	0.5	0	3	-	252.2	260.1	260.2	252.2	1.000	-	1800.8	4.9	1800.2	28	51
	0.8	0	10	-	349.2	379.6	353.4	381.3	1.092	-	1800.3	5.0	1800.6	28	83
10	0.3	2	5	317.2	317.2	317.3	317.3	317.2	1.000	535.8	1359.9	7.9	971.2	54	82
	0.5	0	24	-	361.7	372.5	372.4	361.7	1.000	-	1801.5	5.2	1801.0	28	105
	0.8	0	3	-	538.1	558.8	549.3	539.8	1.003	-	1800.2	7.6	1800.2	31	59
15	0.3	0	4	-	725.4	743.5	745.9	725.4	1.000	-	1800.1	16.3	1800.0	51	94
	0.5	0	6	-	708.4	729.6	730.1	857.5	1.210	-	1800.1	22.3	1800.1	75	143
	0.8	0	2	-	1216.5	1328.9	1249.2	1219.7	1.003	-	1800.1	18.0	1800.1	49	91
20	0.3	0	3	-	904.5	937.7	922.5	904.5	1.000	-	1800.3	42.6	1800.1	79	138
	0.5	0	3	-	1282.1	1469.8	1472.8	1529.1	1.193	-	1800.3	38.2	1800.1	88	147
	0.8	0	2	-	2024.4	2441.1	2207.2	2415.5	1.193	-	1800.3	38.5	1800.2	78	136
25	0.3	0	4	-	1765.0	1823.2	1819.5	1765.0	1.000	-	1800.3	61.6	1800.1	93	166
	0.5	0	4	-	2597.4	2782.7	2767.9	2597.7	1.000	-	1800.3	61.1	1800.1	100	173
	0.8	0	2	-	3307.4	3883.9	3541.1	3312.8	1.002	-	1800.3	51.1	1800.3	86	156
30	0.3	0	3	-	2448.1	2572.0	2556.0	2448.3	1.000	-	1800.4	84.2	1800.2	98	180
	0.5	0	2	-	3260.8	3509.6	3482.4	3263.8	1.001	-	1800.4	90.9	1808.0	130	218
	0.8	0	3	-	4910.4	5574.9	5399.1	4919.7	1.002	-	1800.4	82.1	1800.3	103	189

Note. Boldfaced values indicate better solutions found within the time limit, tighter dual bounds, or more PNEs.

responding to 436 inspection stations. Minnesota allocates roughly \$10 million per year to AIS inspection programs and employs approximately 800–1000 inspectors [102]; assuming two inspectors per lake, this staffing level corresponds to operating about 400–500 inspection stations statewide. We allocate county budgets proportionally to county-level funding levels [102]. We model four AIS types—Zebra Mussels, Starry Stonewort, Eurasian Watermilfoil, and Spiny Water Fleas—yielding $\mathcal{G}_{|\mathcal{L}|\geq 2}^{\text{Self}}$.

Because this instance is extremely large (84 players, roughly 100 lakes per county on average, and millions of arcs), we use a 3600-second time limit for ZR, BZR, and the SW model, and a 600-second limit to compute the warm-start profile $\tilde{\mathbf{x}}_{\text{sw}}^{600}$. Standalone RRR-BRD requires more than 400 seconds from either $\mathbf{0}^N$ or $\tilde{\mathbf{x}}_{\text{sw}}^{600}$ (Table 2.8), indicating a substantially more difficult best-response landscape than in our synthetic instances. Table 2.9 shows that BZR identifies multiple PNEs within the time limit, whereas ZR finds none.

Table 2.8: RRR-BRD results for selfish EBMC games $\mathcal{G}_{|\mathcal{L}| \geq 2}^{\text{Self}}$ in the Minnesota dataset.

n	RRR-BRD (ϕ)		Price of Stability		References		Time (secs)		Round	
	init: $\mathbf{0}^N$	init: $\tilde{\mathbf{x}}_{\text{sw}}^{600}$	$\widetilde{\text{OSW}}$	$\widetilde{\text{POS}}$	BZR(ϕ)	BZR(Bound)	init: $\mathbf{0}^N$	init: $\tilde{\mathbf{x}}_{\text{sw}}^{600}$	init: $\mathbf{0}^N$	init: $\tilde{\mathbf{x}}_{\text{sw}}^{600}$
84	758.8	761.4	887.7	1.166	761.4	861.7	425.4	586.5	5	7

Table 2.9: Comparison of BZR and ZR for selfish EBMC games $\mathcal{G}_{|\mathcal{L}| \geq 2}^{\text{Self}}$ in the Minnesota dataset.

n	# PNEs Found		Best PNE (ϕ)		Dual Bounds		Price of Stability		Time (secs)				# EI Cuts	
	# _{pne(ZR)}	# _{pne(BZR)}	ZR	BZR	ZR	BZR	$\widetilde{\text{OSW}}$	$\widetilde{\text{POS}}$	T(ZR ^{1st})	T(ZR)	T(BZR ^{1st})	T(BZR)	# _{cuts(ZR)}	# _{cuts(BZR)}
84	0	5	-	761.4	861.7	861.7	887.7	1.166	-	3600.8	790.7	3600.8	363	363

We emphasize that PNE existence in this instance is not guaranteed a priori by our sufficient condition (Theorem 2.13). The Minnesota results therefore provide a computational witness of existence for the selfish game, complementing the locally altruistic case where a PNE is immediate via Corollary 2.12. Table 2.10 reports county-level selfish utilities on the Minnesota instance under several strategy profiles: the non-game profile $\bar{\mathbf{x}}_{\text{ng}}$, the first PNE $\hat{\mathbf{x}}_{\text{pne}}^1$, the best PNE $\hat{\mathbf{x}}_{\text{pne}}^*$ found by BZR, and the welfare benchmark $\tilde{\mathbf{x}}_{\text{sw}}^{3600}$. For each county i , we compare utilities column-wise via $u_i^{\text{Self}}(\bar{\mathbf{x}}_{\text{ng}}) \leq u_i^{\text{Self}}(\hat{\mathbf{x}}_{\text{pne}}^1)$, $u_i^{\text{Self}}(\hat{\mathbf{x}}_{\text{pne}}^1) \leq u_i^{\text{Self}}(\hat{\mathbf{x}}_{\text{pne}}^*)$, $u_i^{\text{Self}}(\hat{\mathbf{x}}_{\text{pne}}^*) \leq u_i^{\text{Self}}(\tilde{\mathbf{x}}_{\text{sw}}^{3600})$, and highlight any violations in bold. The complete 84-county table is provided in Appendix 2.N.

Overall, the results suggest that (i) many counties benefit from strategic interaction relative to the non-game baseline, due to unintended positive spillovers; (ii) among equilibria, the best PNE improves utilities for most counties; and (iii) the welfare-oriented benchmark $\tilde{\mathbf{x}}_{\text{sw}}^{3600}$ (and, in the locally altruistic game, the OSW solution) often yields the highest utilities, although exceptions exist (e.g., Sibley and Winona).

These comparisons illustrate why computed equilibria are policy-relevant benchmarks. A PNE is stable in the sense that no county has an incentive to deviate unilaterally, and the socially best PNE for the selfish game offers a tractable compromise between stability and

statewide performance. Moreover, the locally altruistic specification highlights a concrete policy target: if counties can adopt (or are incentivized to adopt) utilities that internalize spillovers, then the welfare-optimal solution is self-enforcing—indeed, Corollary 2.12 shows that the OSW solution is a PNE under locally altruistic utilities. Nevertheless, the best selfish-game PNE can still deviate substantially from the welfare benchmark, as reflected by a POS of 1.166. This motivates future work on (i) incentive design that induces counties to internalize externalities and (ii) equilibrium-informed coordination that leverages equilibrium information from selfish games to improve social welfare.

Table 2.10: County-level utility values using different strategy profiles.

County	$u_c^{\text{Self}}(\bar{\mathbf{x}}_{\text{ng}})$	$u_c^{\text{Self}}(\bar{\mathbf{x}}_{\text{pne}}^1)$	$u_c^{\text{Self}}(\bar{\mathbf{x}}_{\text{pne}}^*)$	$u_c^{\text{Self}}(\bar{\mathbf{x}}_{\text{sw}}^{3600})$	County	$u_c^{\text{Self}}(\bar{\mathbf{x}}_{\text{ng}})$	$u_c^{\text{Self}}(\bar{\mathbf{x}}_{\text{pne}}^1)$	$u_c^{\text{Self}}(\bar{\mathbf{x}}_{\text{pne}}^*)$	$u_c^{\text{Self}}(\bar{\mathbf{x}}_{\text{sw}}^{3600})$
Anoka	11292	12466	12495	15723	McLeod	7194	7479	7503	8933
Becker	24219	26489	26497	31712	Meeker	8866	9423	9428	9964
Hennepin	21136	22442	22474	27024	Sherburne	5077	5557	5564	6698
Hubbard	22780	24426	24444	29543	Sibley	2390	2388	2388	2429
Isanti	3635	3876	3884	4227	Stearns	20200	21544	21554	26076
Lyon	2752	2925	2939	3314	Winona	1561	1579	1579	1444
Mahnomen	4177	4424	4428	6012	Wright	22047	23782	23805	29464

Note. Counties for which the inequalities regarding utility value comparisons do not hold are highlighted in bold.

2.7.4 Large-scale Knapsack Problem Games Datasets and Results

To further assess the effectiveness of RRR-BRD and BZR, we report extensive computational results for large-scale knapsack problem games (KPGs), a standard benchmark in the IPG literature [36, 38, 49, 58]. In a KPG, each player i solves a binary knapsack problem of the form $\max_{\mathbf{x}^i} \left\{ \sum_{j=1}^m p_j^i x_j^i + \sum_{k=1, k \neq i}^n \sum_{j=1}^m f_{k,j}^i x_j^i x_j^k : \sum_{j=1}^m w_j^i x_j^i \leq \mathcal{B}^i, \mathbf{x}^i \in \{0, 1\}^m \right\}$, where $x_j^i = 1$ if player i selects item j (and 0 otherwise). The parameters p_j^i , w_j^i , and $\mathcal{B}^i$ denote item profit, item weight, and player budget, respectively. Strategic interactions are captured by the bilinear coefficients $f_{k,j}^i$, which contribute to player i 's payoff when both players i and k select item j . To linearize $x_j^i x_j^k$, we introduce auxiliary variables $z_j^{i,k}$ imposing the following constraints: $z_j^{i,k} \geq x_j^i + x_j^k - 1$, $z_j^{i,k} \leq x_j^i$, $z_j^{i,k} \leq x_j^k$, $\forall j \in \{1, \dots, m\}$, $\forall i \in N$, $\forall k \in N \setminus \{i\}$.

Following Dragotto and Scatamacchia [58], we also incorporate two classes of problem-specific equilibrium inequalities—strategic dominance and strategic payoff inequalities—along with the standard equilibrium inequalities. For larger instances ($n \geq 15$), identifying dominance relations and minimal interaction sets for strategic payoff inequalities becomes computationally expensive, so we omit these inequalities.

We adopt the data generation scheme of Dragotto and Scatamacchia [58] for interaction types (A) and (B), and use a modified version of type (C) to better match our experimental goal. In type (A), each $f_{k,j}^i$ is set to a constant drawn uniformly from $[1, 100]$; in type (B), each $f_{k,j}^i$ is drawn independently from $[1, 100]$. In the original type (C), $f_{k,j}^i \sim \text{Unif}[-100, 100]$, which can induce very strong negative (adversarial) interactions; as also observed in Dragotto and Scatamacchia [58], such instances often fail to admit a PNE even for small sizes ($n = 2, 3$), and this issue becomes more pronounced as n grows. Because our convergence guarantees for RRR-BRD (and hence BZR’s use of it as a subroutine) are conditional—on *PNE existence*—we prioritize stress-testing instances that remain adversarial yet are not overwhelmingly likely to be PNE-free. Concretely, we modify type (C) by sampling $f_{k,j}^i$ from $[-20, 0]$, which preserves negative interactions but reduces their magnitude to an “admissible” level that more frequently yields equilibria in practice. Budgets are set as a fraction of total item weight: $\mathcal{B}^i \in \left\{0.2 \sum_{j=1}^m w_j^i, 0.5 \sum_{j=1}^m w_j^i, 0.8 \sum_{j=1}^m w_j^i\right\}$. We start from the largest KPG setting in Dragotto and Scatamacchia [58] ($n \in \{2, 3\}$, $m = 100$) and scale to $n \in \{2, 3, 5, 8, 10, 15, 20, 25, 30\}$ with $m = 100$, yielding 27 generated instances per interaction type.

Table 2.11 summarizes RRR-BRD performance under two initial profiles, $\mathbf{0}^N$ and $\tilde{\mathbf{x}}_{\text{sw}}^{300}$; complete results appear in Appendix 2.K. Across interaction types, we observe that type (C) instances exhibit larger $\widetilde{\text{POS}}$, and require substantially more rounds before reaching a PNE. These trends indicate that negative interactions produce markedly more complex

best-response dynamics than types (A) and (B).

Table 2.11: Summarized RRR-BRD results with different initial strategy profiles for Type A, B, and C KPG.

type	Avg [†] PNE (ϕ)		Avg Price of Stability		Avg References		Avg Time (secs)		Avg Round	
	init: $\mathbf{0}^N$	init: $\tilde{\mathbf{x}}_{\text{sw}}^{300}$	$\widetilde{\text{OSW}}$	$\widetilde{\text{POS}}$	BZR(ϕ)	BZR(Bound)	init: $\mathbf{0}^N$	init: $\tilde{\mathbf{x}}_{\text{sw}}^{300}$	init: $\mathbf{0}^N$	init: $\tilde{\mathbf{x}}_{\text{sw}}^{300}$
Type A	700.7	711.0	714.1	1.012	711.2	715.6	2.6	192.9	7.7	3.9
Type B	706.1	714.7	717.1	1.010	715.5	719.2	2.9	198.2	8.6	4.6
Type C	14.0	14.1	16.9	1.171	14.2	16.3	1.8	73.2	29.5	16.5

Note. †: The averages of ϕ , OSW, and POS are computed only from instances where a PNE is found under both initial solutions.

Table 2.12 provides a summarized comparison of BZR and ZR across the three KPG datasets, reporting how often a PNE was found, multiple PNEs were identified, the best PNE was obtained, and a tighter dual bound was achieved. Full results for type (B) are given in Table 2.13, while those for types (A) and (C) appear in Appendix 2.L. Overall, BZR consistently outperforms ZR across all interaction types in (i) time to the first PNE, (ii) the number of distinct PNEs found, and (iii) the best PNE value identified. For dual bounds, the advantage is clearer in types (A) and (B), whereas in type (C) the two methods exhibit complementary strengths: as shown in Table 2.30 (Appendix 2.L), BZR generally produces tighter dual bounds for $n \leq 15$, while ZR tends to perform better for $n \geq 20$.

Table 2.12: Summarized comparison of BZR and ZR for Type A, B, and C KPG.

type	# PNEs Found		Finding a PNE		Best PNE			Dual Bounds			Avg POS	Avg # EI Cuts	
	# _{pne(ZR)}	# _{pne(BZR)}	ZR	BZR	Ties	ZR	BZR	Ties	ZR	BZR	$\widetilde{\text{POS}}$	# _{cuts(ZR)}	# _{cuts(BZR)}
Type A	14	719	7	27	7	0	20	10	0	17	1.010	1206.7	1012.0
Type B	12	1766	5	27	5	0	22	7	1	19	1.006	945.9	966.6
Type C	12	675	7	24	7	0	17	8	7	12	1.175	2180.9	2516.4

How Interaction Coefficients and the Number of Players Affect $q_\mu[R]$. When interaction coefficients are predominantly positive, improving BR moves more often enter equilibrium basins, leading to larger $\tilde{q}_\mu[R]$ and faster PNE discovery; as interactions become

Table 2.13: Comparison of BZR and ZR for Type B KPG.

n	BG	# PNEs Found		Best PNE (ϕ)		Dual Bounds		Price of Stability		Time (secs)				# EI Cuts	
		# _{pne} (ZR)	# _{pne} (BZR)	ZR	BZR	ZR	BZR	$\widetilde{\text{OSW}}$	$\widetilde{\text{POS}}$	T(ZR ^{1st})	T(ZR)	T(BZR ^{1st})	T(BZR)	# _{cuts} (ZR)	# _{cuts} (BZR)
2	0.2	3	9	8.2	8.2	8.2	8.2	8.3	1.018	9.7	13.4	0.7	10.0	1944	1913
	0.5	4	125	14.2	14.2	14.2	14.2	14.4	1.010	72.9	130.9	0.3	72.9	2281	2154
	0.8	1	9	19.1	19.1	19.1	19.1	19.1	1.002	1.2	1.4	0.3	2.2	1224	1239
3	0.2	0	195	-	15.6	15.7	15.6	15.9	1.023	-	1800.3	1.1	1800.3	2679	2161
	0.5	0	222	-	28.9	29.2	29.1	29.4	1.016	-	1800.3	0.6	1800.2	2744	2100
	0.8	3	291	40.0	40.0	40.1	40.1	40.3	1.007	590.5	1800.3	1.2	1801.3	2287	2361
5	0.2	0	84	-	40.5	40.8	40.8	40.9	1.012	-	1800.2	2.0	1800.2	663	832
	0.5	0	129	-	78.0	78.8	78.8	78.8	1.011	-	1800.4	2.2	1802.9	1377	1325
	0.8	0	107	-	108.3	108.7	108.6	108.7	1.003	-	1800.5	4.0	1800.4	1790	983
8	0.2	0	75	-	91.6	92.7	92.7	92.6	1.011	-	1800.3	4.4	1800.5	755	1055
	0.5	0	89	-	188.6	190.0	190.0	189.9	1.007	-	1801.1	10.0	1801.3	874	1282
	0.8	0	12	-	272.9	273.4	273.2	273.1	1.001	-	1801.2	6.9	1800.8	998	176
10	0.2	0	43	-	135.9	137.1	137.0	136.9	1.007	-	1801.1	2.3	1800.4	523	721
	0.5	0	28	-	287.8	289.5	289.4	289.0	1.004	-	1800.8	6.1	1801.2	306	508
	0.8	0	27	-	426.9	428.4	428.3	427.8	1.002	-	1800.5	7.4	1801.4	667	435
15	0.2	0	41	-	287.3	290.6	290.6	289.5	1.008	-	1801.9	20.4	1801.0	358	960
	0.5	0	19	-	642.1	645.7	645.5	644.0	1.003	-	1800.9	20.4	1801.5	252	466
	0.8	0	12	-	963.3	967.1	966.6	965.2	1.002	-	1800.9	23.0	1801.4	258	312
20	0.2	0	16	-	488.4	493.2	493.1	491.2	1.006	-	1800.7	13.4	1800.6	334	496
	0.5	0	11	-	1123.1	1129.2	1129.1	1125.5	1.002	-	1801.2	27.6	1801.3	309	402
	0.8	0	10	-	1671.7	1677.5	1677.3	1672.9	1.001	-	1800.8	24.5	1800.9	264	367
25	0.2	0	12	-	760.9	767.6	767.5	763.5	1.003	-	1800.8	21.6	1801.0	315	444
	0.5	0	11	-	1716.7	1728.7	1728.5	1723.7	1.004	-	1800.9	19.2	1800.9	338	448
	0.8	0	9	-	2610.3	2623.1	2622.8	2614.3	1.002	-	1800.9	28.9	1800.7	371	420
30	0.2	0	10	-	1083.3	1094.7	1094.6	1088.8	1.005	-	1801.0	34.1	1800.7	399	540
	0.5	0	11	-	2459.8	2473.3	2473.4	2466.1	1.003	-	1800.7	98.9	1801.0	237	584
	0.8	0	8	-	3748.6	3764.1	3764.0	3750.8	1.001	-	1800.8	27.0	1800.6	425	430

Note. Boldfaced values indicate better solutions found within the time limit, tighter dual bounds, or more PNEs.

more complex—e.g., when coefficients are mixed or negative—the dynamics can become more complicated and $q_\mu[R]$ may decrease. This viewpoint aligns with our EBMC variants: in $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$, other players' selections are typically helpful for player i (Section 2.6.2), whereas in $\mathcal{G}_{|\mathcal{L}|\geq 2}^{\text{Self}}$, other players' updates can affect player i either positively or negatively, yielding more complicated BR trajectories. Similarly, KPG types (A) and (B) behave like predominantly positive-interaction instances, while our type (C) is closer to a mixed regime. Finally, increasing the number of players can amplify externalities; for example, the original type (C) construction in Dragotto and Scatamacchia [58] can exhibit a more adversarial landscape as externalities increase (Section 2.7.4), which may drive $q_\mu[R]$ to be very small. This is consistent with our results: $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$ and KPG types (A) and (B) find PNEs in all instances, whereas $\mathcal{G}_{|\mathcal{L}|\geq 2}^{\text{Self}}$ fails in four instances and our KPG type (C) fails in three instances.

2.8 Conclusion

Computing PNEs in IPGs remains challenging, primarily because most existing exact approaches rely on joint formulations that couple players’ strategy sets, and computational evidence is therefore largely restricted to instances with only a few players and modest per-player model sizes. This scalability gap is particularly salient in large-scale resource-allocation settings with many stakeholders, such as the EBMC games studied in this chapter. To address this challenge, we introduce round random-restart best-response dynamics (RRR-BRD) for IPGs and provide a probabilistic convergence guarantee whenever a PNE exists and the restart law assigns positive probability to basins that reach a PNE.

We further embed RRR-BRD as a randomized local-search subroutine within the ZR framework, yielding substantial practical gains in finding PNEs and high-social-welfare PNEs, and often improving dual bounds. Overall, these improvements translate into an order-of-magnitude enhancement in the tractable scale of equilibrium computation for our tested IPG classes. We also discuss how RRR-BRD can serve as a local-search subroutine within other IPG algorithms whenever integer-feasible profiles are available. On the modeling side, we study two EBMC game variants—locally altruistic and selfish—by specifying how players value edge coverage, and we establish sufficient conditions for PNE existence in each setting. Using the MN dataset, we illustrate how equilibrium outcomes depend on these behavioral specifications and provide managerial insights for allocating limited inspection resources at both the statewide and county levels.

Finally, our results point to two important directions for future research. First, our probabilistic guarantees are expressed through the instance-dependent success probability under restarts, $q_\mu[R]$, which is difficult to compute in general; deriving tractable estimators/bounds for $q_\mu[R]$ under additional structural assumptions, and designing restart laws that improve it,

are promising extensions. Second, while we focus on equilibrium computation and comparative analysis under different utility specifications, we do not design incentive mechanisms; developing implementable policy levers that steer equilibria toward socially desirable outcomes, and analyzing welfare–stability trade-offs and sensitivity, are natural next steps.

Acknowledgments. H. Lee and R. Hildebrand were partially funded by AFOSR grant FA9550-21-1-0107. H. Lee and İ. E. Büyüktaktın have also been funded by the Grado Department of ISE at VT. S. Cai was partially funded by MAISRC Subaward H009064601, and all authors were partially funded by the MAISRC Subaward H0113450M1. The authors are grateful to MAISRC for making the Minnesota dataset publicly accessible, and Amy Kinsley (MAISRC), Nick Phelps (MAISRC), Alex Bajcz (MAISRC), Bob Haight (USFS), Tina Fitzgerald (DNR) and Adam Doll (DNR) for pointing us to the Minnesota dataset and for helpful discussion on the AIS inspection problem in MN. Any opinions, findings, conclusions, or recommendations expressed in this study are those of the authors and do not necessarily reflect the views of the Air Force Office of Scientific Research and MAISRC.

Appendix

2.A Acronyms and Notations

Table 2.14 lists acronyms used throughout this chapter. Tables 2.15–2.16 summarize the notation.

Table 2.14: List of acronyms used in this chapter and its appendix

Acronym	Meaning
IPG / IP	Integer programming game / integer program.
PNE	Pure Nash equilibrium.
BR	Best response (noun); best-response (adjective).
SCC	Strongly connected component in a directed graph.
DAG	Directed acyclic graph.
RR-BRD	Random-Restart best-response dynamics.
RRR-BRD	Round RR-BRD, i.e., a round-based implementation of RR-BRD.
EI	Equilibrium inequality.
ZR	Zero-regret (algorithm).
BZR	BRD-incorporated zero-regret (algorithm).
KPG	Knapsack problem game.
EBMC	Edge-weighted budgeted maximum coverage (game/model).
AIS	Aquatic invasive species.
OSW	Optimal social welfare (social optimum).
POS	Price of stability.

Table 2.15: Summary of notation used in this chapter and its appendix (core notation)

Symbol	Meaning
IPGs.	
$\mathcal{G}$	An integer programming game (IPG); the IPG tuple as defined in Section 2.1.
$N = \{1, \dots, n\}$	Set of players; $n := N $.
$m_i; m$	Number of integer variables of player i ; when m_i is the same for all $i \in N$, we write m .
$\mathcal{X}_i$	Polyhedrally bounded pure-integer set of player i in the IPG (i.e., a finite strategy set).
$\mathcal{X}$	Joint strategy set, $\mathcal{X} := \mathcal{X}_1 \times \dots \times \mathcal{X}_n$.
$\mathbf{x}^i \in \mathcal{X}_i; \mathbf{x} = (\mathbf{x}^i)_{i \in N}$	Player i 's strategy; joint strategy profile.
$u_i(\mathbf{x}^i, \mathbf{x}^{-i})$	Utility (objective value) of player i under profile $\mathbf{x}$.
$\phi(\mathbf{x})$	Social welfare function, $\phi(\mathbf{x}) = \sum_{i \in N} u_i(\mathbf{x})$.
$\bar{\mathbf{x}}_{\text{sw}}^*$	OSW solution.
$\hat{\mathbf{x}}_{\text{pne}}^*$	Best PNE (socially optimal PNE).
Normal-Form Games, BR State Graph, and RRR-BRD.	
S_i	Set of pure strategies of player i in the finite normal-form game.
$S = \prod_{i \in N} S_i$	Set of pure strategy profiles (normal-form viewpoint).
$s = (s_i, s_{-i})$	Strategy profile; s_{-i} denotes the strategies of all players except i .
$\mathcal{BR}_i^+(s_{-i})$	Set of (improving) best responses at s_{-i} .
SCC; sink SCC	Strongly connected component; a sink SCC has no outgoing edges in the condensation DAG.
Condensation DAG	Directed acyclic graph obtained by contracting SCCs of the BR state graph.
μ	Restart distribution over S used by RRR-BRD.
R	Round cap per attempt in RRR-BRD.
$q_\mu[R]$	Probability that an attempt from $s \sim \mu$ reaches a PNE within R rounds.
Equilibrium Inequalities and Lifted Formulations.	
$\mathcal{S}_{\text{PNE}}(\mathcal{G})$	Set of pure Nash equilibria of game $\mathcal{G}$.
$\mathcal{K}$	Lifted/linearized formulation of the joint strategy profile.
$\mathcal{E}$	Equilibrium closure.
Ω	Set over which equilibrium inequalities are generated/enforced.

Table 2.16: Summary of notation used in this chapter and its appendix (applications and experiments)

Symbol	Meaning
EBMC Games.	
$i \in N$	Player index in the EBMC model/games.
$\mathcal{I}_i$	Set of vertices controlled by player i (player-level decision set).
x_j	Binary variable indicating whether vertex j is selected.
y_{jk}	Binary variable indicating whether arc (j, k) is covered.
$\mathcal{B}_i$	Budget of player i (inspection budget).
w_{jk}	Edge weight for edge (j, k) .
$\mathcal{G}_{ \mathcal{L} =1}^{\text{Self}}$	Selfish EBMC game with a single AIS type.
$\mathcal{G}_{ \mathcal{L} \geq 2}^{\text{Self}}$	Selfish EBMC game with multiple AIS types.
$\mathcal{G}^{\text{Alt}}$	Locally altruistic EBMC game.
KPGs.	
J	Set of items in the knapsack problem game.
$x_j^i \in \{0, 1\}$	Binary selection of item j by player i .
$\mathcal{B}_i$	Knapsack capacity of player i (in KPG).
p_j^i	Profit of item j for player i .
$f_{k,j}^i$	Bilinear coefficient associated with item j for players i and k .
Computational Results.	
$\tilde{q}_\mu[R]$	$q_\mu[R]$ estimated via Monte Carlo simulation.
$\mathbf{0}^N$	Initial strategy profile (all-zero vector).
$\tilde{\mathbf{x}}_{\text{sw}}^{\text{time}}$	Initial strategy profile obtained by solving the SW model with a time limit (seconds).
$\bar{\mathbf{x}}_{\text{ng}}$	Non-game strategy profile.
$\hat{\mathbf{x}}_{\text{pne}}^1 / \hat{\mathbf{x}}_{\text{pne}}^*$	First identified PNE / best identified PNE (maximizing ϕ) within the time limit.
$\phi(\cdot)$	Social welfare function; values are reported in thousands and rounded to the nearest tenth.
$\widetilde{\text{OSW}}$	Approximate optimal social welfare, $\phi(\tilde{\mathbf{x}}_{\text{sw}}^{\text{time}})$.
$\widetilde{\text{POS}}$	Approximate price of stability, $\phi(\tilde{\mathbf{x}}_{\text{sw}}^{\text{time}})/\phi(\hat{\mathbf{x}}_{\text{pne}}^*)$.
$\#_{\text{cuts}}(\text{ZR}) / \#_{\text{cuts}}(\text{BZR})$	Number of equilibrium inequalities added by ZR / BZR.
$\#_{\text{pne}}(\text{ZR}) / \#_{\text{pne}}(\text{BZR})$	Number of distinct PNEs found by ZR / BZR.
$\text{T}(\text{ZR}^{\text{1st}}) / \text{T}(\text{BZR}^{\text{1st}})$	Time to find the first PNE using ZR / BZR.
$\text{T}(\text{ZR}) / \text{T}(\text{BZR})$	Total runtime of the ZR / BZR algorithm.

2.B Proofs

Proof of Theorem 2.2

Proof. (i) For attempt r , let $I_r := \mathbf{1}\{\tau_{\text{PNE}} \leq R \text{ in attempt } r\}$, i.e., $I_r = 1$ iff the r -th attempt reaches (and certifies) a PNE within at most R rounds. Across attempts we independently resample $s \sim \mu$ and re-draw fresh round permutations; hence $(I_r)_{r \geq 1}$ are i.i.d. Bernoulli with parameter $q_\mu[R]$. Let

$$R^* := \inf\{r \geq 1 : I_r = 1\}$$

be the number of attempts until success. Then

$$\Pr(R^* > r) = (1 - q_\mu[R])^r \xrightarrow{r \rightarrow \infty} 0,$$

so $\Pr(R^* < \infty) = 1$ and the algorithm finds a PNE almost surely.

(ii) Let L_r be the number of BR solves in attempt r . For each failed attempt $r < R^*$, exactly R rounds are executed, so $L_r = nR$. On the successful attempt, a PNE is first hit during one of the R rounds, and the algorithm certifies it by the end-of-round no-change test; thus $L_{R^*} \leq nR$. Taking expectations and using $\mathbb{E}[R^*] = 1/q_\mu[R]$ (geometric),

$$\mathbb{E}\left[\sum_{r=1}^{R^*} L_r\right] \leq nR \mathbb{E}[R^*] = \frac{nR}{q_\mu[R]}.$$

This establishes the bound in part (ii). □

Proof of Corollary 2.5

Proof. Because each $\mathcal{X}_i$ is a bounded integer polyhedron, it contains finitely many integer points, so $\mathcal{X} = \prod_{i \in N} \mathcal{X}_i$ is finite and can be viewed as the pure strategy space S of a finite normal-form game. The RRR-BRD-IPG algorithm (Algorithm 2.4) is exactly RRR-BRD (Algorithm 2.1) applied to this finite game, with restart law μ induced by **RANDOM_GENERATION**. If $\mathcal{G}$ admits a PNE and $q_\mu[R] > 0$, then the hypotheses of Theorem 2.2 hold, so the almost-sure convergence and the bound on the expected number of BR solves follow directly. $\square$

Proof of Proposition 2.10

Proof. By Definition 2.9, $\mathcal{A}^-[\mathcal{J}_i] := \delta_-(\mathcal{J}_i) \cup \mathcal{A}_i$. Fix any edge $(j, k) \in \mathcal{A}$. Since $\mathcal{J} = \bigsqcup_{i \in N} \mathcal{J}_i$ is a partition, there is a *unique* index $i \in N$ such that $k \in \mathcal{J}_i$. There are two cases: (i) if $j \in \mathcal{J}_i$ as well, then $(j, k) \in \mathcal{A}_i \subseteq \mathcal{A}^-[\mathcal{J}_i]$; (ii) if $j \notin \mathcal{J}_i$, then $(j, k) \in \delta_-(\mathcal{J}_i) \subseteq \mathcal{A}^-[\mathcal{J}_i]$. Hence every edge $(j, k) \in \mathcal{A}$ belongs to $\bigcup_{i \in N} \mathcal{A}^-[\mathcal{J}_i]$. Furthermore, since each edge (j, k) is unique, it must exclusively belong to one set $\mathcal{A}^-[\mathcal{J}_i]$. Therefore, $\mathcal{A} = \bigsqcup_{i \in N} \mathcal{A}^-[\mathcal{J}_i]$.

Using this partition and the definition of u_i^{Self} , we obtain

$$\phi(\mathbf{x}) = \sum_{(j,k) \in \mathcal{A}} w_{jk} y_{jk} = \sum_{i \in N} \sum_{(j,k) \in \mathcal{A}^-[\mathcal{J}_i]} w_{jk} y_{jk} = \sum_{i \in N} \tilde{u}_i^{\text{Self}}(\mathbf{y}) = \sum_{i \in N} u_i^{\text{Self}}(\mathbf{x}),$$

which proves the claim. $\square$

Proof of Theorem 2.11

Proof. Fix a player $i \in N$. The arcs of the entire graph can be partitioned into those that are incident to $\mathcal{J}_i$ and those that are not, i.e., $\mathcal{A} = \mathcal{A}[\mathcal{J}_i] \sqcup \mathcal{A}_{-i}$, where $\mathcal{A}[\mathcal{J}_i] = \mathcal{A}_i \cup \delta_+(\mathcal{J}_i) \cup \delta_-(\mathcal{J}_i)$ and $\mathcal{A}_{-i}$ collects all remaining arcs. For a given profile $(\mathbf{x}^i, \bar{\mathbf{x}}^{-i})$, the

potential can be written as

$$\begin{aligned}\phi(\mathbf{x}) &= \sum_{j,k \in \mathcal{J}_i} w_{jk} \cdot \max(x_j, x_k) + \sum_{j \in \mathcal{J}_i, k \in \mathcal{J}_{-i}} w_{jk} \cdot \max(x_j, \bar{x}_k) + \sum_{j \in \mathcal{J}_{-i}, k \in \mathcal{J}_i} w_{jk} \cdot \max(\bar{x}_j, x_k) + \sum_{j,k \in \mathcal{J}_{-i}} w_{jk} \cdot \max(\bar{x}_j, \bar{x}_k) \\ &= \sum_{(j,k) \in \mathcal{A}[\mathcal{J}_i]} u_i^{\text{Alt}}(\mathbf{x}^i, \bar{\mathbf{x}}^{-i}) + \sum_{j,k \in \mathcal{J}_{-i}} w_{jk} \cdot \max(\bar{x}_j, \bar{x}_k).\end{aligned}$$

The second term depends only on $\bar{\mathbf{x}}^{-i}$ and is therefore constant with respect to changes in $\mathbf{x}^i$. The first term is precisely $u_i^{\text{Alt}}(\mathbf{x}^i, \bar{\mathbf{x}}^{-i})$ (where $\tilde{u}_i^{\text{Alt}}(\mathbf{y})$ is rewritten as $u_i^{\text{Alt}}(\mathbf{x})$). Hence, for any $\bar{\mathbf{x}}^i, \hat{\mathbf{x}}^i \in \mathcal{X}_i$,

$$\phi(\hat{\mathbf{x}}^i, \bar{\mathbf{x}}^{-i}) - \phi(\bar{\mathbf{x}}^i, \bar{\mathbf{x}}^{-i}) = u_i^{\text{Alt}}(\hat{\mathbf{x}}^i, \bar{\mathbf{x}}^{-i}) - u_i^{\text{Alt}}(\bar{\mathbf{x}}^i, \bar{\mathbf{x}}^{-i}),$$

which shows that ϕ is an exact potential function for the locally altruistic game. □

Proof of Corollary 2.12

Proof. By Theorem 2.11, the locally altruistic EBMC games are exact potential games with potential function $\phi(\mathbf{x})$, and $\bar{\mathbf{x}}_{\text{sw}}^*$ is an optimal solution maximizing $\phi(\mathbf{x})$ over $\mathbf{x} \in \mathcal{X}$. By Lemma 2.1 of Monderer and Shapley [105], which states that any maximizer of a potential function is a PNE, $\bar{\mathbf{x}}_{\text{sw}}^*$ is a PNE for locally altruistic EBMC games. □

Proof of Theorem 2.13.

Proof. Case 1: $\mathcal{J}^{\text{ng}} \subseteq \mathcal{J}^0$. Fix $i \in N$. Since vertices in $\mathcal{J}^0$ have no types, other players' selections in $\mathcal{J}^0$ create no relevant coverage benefit for player i in the selfish utility. Hence, for all $\mathbf{x}^i \in \mathcal{X}_i$,

$$u_i^{\text{self}}(\mathbf{x}^i, \bar{\mathbf{x}}_{\text{ng}}^{-i}) = u_i^{\text{self}}(\mathbf{x}^i, \mathbf{0}^{-i}).$$

By definition of $\bar{\mathbf{x}}_{\text{ng}}^i$,

$$\bar{\mathbf{x}}_{\text{ng}}^i \in \arg \max_{\mathbf{x}^i \in \mathcal{X}_i} u_i^{\text{self}}(\mathbf{x}^i, \mathbf{0}^{-i}),$$

so $\bar{\mathbf{x}}_{\text{ng}}^i$ is also a best response to $\bar{\mathbf{x}}_{\text{ng}}^{-i}$. Thus $\bar{\mathbf{x}}_{\text{ng}}$ is a PNE (for both $|\mathcal{L}| = 1$ and $|\mathcal{L}| \geq 2$).

Case 2: $\mathcal{J}^{\text{ng}} \subseteq \mathcal{J}^+$ and $|\mathcal{L}| = 1$. Fix $i \in N$. For any $\mathbf{x}^i \in \mathcal{X}_i$, overlap of covered edges implies

$$u_i^{\text{self}}(\mathbf{x}^i, \bar{\mathbf{x}}_{\text{ng}}^{-i}) \leq u_i^{\text{self}}(\mathbf{x}^i, \mathbf{0}^{-i}) + u_i^{\text{self}}(\mathbf{0}^i, \bar{\mathbf{x}}_{\text{ng}}^{-i}).$$

Since $\bar{\mathbf{x}}_{\text{ng}}^i$ maximizes $u_i^{\text{self}}(\cdot, \mathbf{0}^{-i})$,

$$u_i^{\text{self}}(\mathbf{x}^i, \bar{\mathbf{x}}_{\text{ng}}^{-i}) \leq u_i^{\text{self}}(\bar{\mathbf{x}}_{\text{ng}}^i, \mathbf{0}^{-i}) + u_i^{\text{self}}(\mathbf{0}^i, \bar{\mathbf{x}}_{\text{ng}}^{-i}).$$

Under $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$, the instance is bipartite as in Section 2.5.3, with each coverable edge having one endpoint in $\mathcal{J}^+$ and one in $\mathcal{J}^0$. With $\mathcal{J}^{\text{ng}} \subseteq \mathcal{J}^+$, there is no double counting of edges, so

$$u_i^{\text{self}}(\bar{\mathbf{x}}_{\text{ng}}^i, \mathbf{0}^{-i}) + u_i^{\text{self}}(\mathbf{0}^i, \bar{\mathbf{x}}_{\text{ng}}^{-i}) = u_i^{\text{self}}(\bar{\mathbf{x}}_{\text{ng}}^i, \bar{\mathbf{x}}_{\text{ng}}^{-i}).$$

Therefore,

$$u_i^{\text{self}}(\mathbf{x}^i, \bar{\mathbf{x}}_{\text{ng}}^{-i}) \leq u_i^{\text{self}}(\bar{\mathbf{x}}_{\text{ng}}^i, \bar{\mathbf{x}}_{\text{ng}}^{-i}) \quad \forall \mathbf{x}^i \in \mathcal{X}_i,$$

so $\bar{\mathbf{x}}_{\text{ng}}^i$ is a best response to $\bar{\mathbf{x}}_{\text{ng}}^{-i}$. Since i is arbitrary, $\bar{\mathbf{x}}_{\text{ng}}$ is a PNE for $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$. □

2.C Random Feasible-Start Generators for KPG and EBMC

We provide two randomized feasible-start generators for each IPG class studied in this chapter: (i) a *maximal* generator, which tends to produce strategies that are close to capacity-filling and thus resemble typical best responses; and (ii) a *full-support* generator, which assigns strictly positive probability to every feasible pure strategy. In our RRR-BRD implementation, we alternate between these two variants (e.g., using the maximal generator on odd-numbered restarts and the full-support generator on even-numbered restarts), although any mixture is admissible.

The design goals are: (a) every feasible profile in $\mathcal{X}$ has nonzero probability under the full-support generator (for the theoretical guarantees in Section 2.3.3); and (b) the generators remain lightweight (expected linear time) so that restarts are inexpensive.

Notation. Let $N = \{1, \dots, n\}$ be the set of players/counties. In KPG, let J be the set of items, $w_j^i \geq 0$ the weight of item $j \in J$ for player $i \in N$, and $\mathcal{B}_i \geq 0$ the capacity. A pure strategy for player i is $\mathbf{x}^i \in \{0, 1\}^{|J|}$ with $\sum_{j \in J} w_j^i x_j^i \leq \mathcal{B}_i$. In EBMC, let $\mathcal{J}_i$ denote the lakes in county i and $\mathcal{B}_i \in \mathbb{Z}_{\geq 0}$ its inspection budget; a pure strategy is $\mathbf{x}^i \in \{0, 1\}^{|\mathcal{J}_i|}$ with $\sum_{j \in \mathcal{J}_i} x_j \leq \mathcal{B}_i$.

Generators for Knapsack Problem Games (KPG)

Maximal generator (random-order greedy). This generator matches the implementation used in our code: for each player, we take a randomized ordering of items and perform a single greedy pass, adding an item whenever it fits in the remaining capacity.

Algorithm 2.14 MAXRANDOMGENERATION-KPG (random-order greedy)**Require:** Items J , capacities $(\mathcal{B}_i)_{i \in N}$, weights $(w_j^i)_{i \in N, j \in J}$ **Ensure:** Feasible profile $\bar{\mathbf{x}} \in \prod_{i \in N} \{\mathbf{x}^i \in \{0, 1\}^{|J|} : \sum_j w_j^i x_j^i \leq \mathcal{B}_i\}$

```

1: for each  $i \in N$  do
2:    $S \leftarrow \emptyset$ ,  $R \leftarrow \mathcal{B}_i$ 
3:    $\pi \leftarrow \text{RANDOMPERMUTATION}(J)$   $\triangleright$  encodes shuffle
4:   for each  $j \in \pi$  do
5:      $w \leftarrow w_j^i$ 
6:     if  $w \leq R$  then
7:        $S \leftarrow S \cup \{j\}$ ;  $R \leftarrow R - w$ 
8:
9:
10:  Set  $x_j^i \leftarrow \mathbf{1}\{j \in S\}$  for all  $j \in J$ 
11:
12: return  $\bar{\mathbf{x}} = (\mathbf{x}^i)_{i \in N}$ 

```

Algorithm 2.14 runs in $O(|N||J|)$ time and always returns a feasible strategy for each player, since items are only accepted when the remaining capacity R allows. Because all randomness sits in the permutation π , it yields a diverse collection of near-maximal solutions that typically resemble best responses.

Full-support generator (Bernoulli + rejection). The full-support generator uses a Bernoulli–rejection scheme. For each player i , we choose a Bernoulli parameter p_i that reflects the “density” of the knapsack, typically on the order of the mean fill ratio $\mathcal{B}_i/W_i$, where $W_i := \sum_{j \in J} w_j^i$:

Every feasible knapsack pattern for player i can be generated with positive probability in a single Bernoulli draw (because each coordinate is independent and $p_i \in (0, 1)$), and is accepted whenever it satisfies the capacity constraint. Thus, after accounting for the rejection step, MAXRANDOMGENERATION-KPG induces a full-support distribution over the feasible set. For moderate p_i and capacities, the expected number of rejection loops is small, so the expected runtime remains linear in $|J|$ per player.

In RRR-BRD, we alternate these two generators across restarts (e.g., MAXRANDOMGENERATION-

Algorithm 2.15 RANDOMGENERATION-KPG (full-support, expected linear time)

Require: Items J , capacities $(\mathcal{B}_i)_{i \in N}$, weights $(w_j^i)_{i \in N, j \in J}$
Ensure: Feasible profile $\bar{\mathbf{x}} \in \prod_{i \in N} \{\mathbf{x}^i \in \{0, 1\}^{|J|} : \sum_j w_j^i x_j^i \leq \mathcal{B}_i\}$

- 1: **for each** $i \in N$ **do**
- 2: $W_i \leftarrow \sum_{j \in J} w_j^i$
- 3: Choose p_i in $(0, 1)$, e.g. $p_i \leftarrow \mathcal{B}_i / W_i$
- 4: **repeat**
- 5: **for each** $j \in J$ **do**
- 6: Draw $x_j^i \sim \text{Ber}(p_i)$ independently
- 7:
- 8: **until** $\sum_{j \in J} w_j^i x_j^i \leq \mathcal{B}_i$ $\triangleright$ reject if capacity violated
- 9:
- 10: **return** $\bar{\mathbf{x}} = (\mathbf{x}^i)_{i \in N}$

KPG for odd restarts and RANDOMGENERATION-KPG for even restarts), combining capacity-filling starts with fully supported random perturbations.

Generators for EBMC

For EBMC, budgets are cardinality bounds, so we can work directly with subsets of lakes.

Maximal generator (uniform cardinality sampling). Here we sample a subset of lakes of size $b_i := \min\{\mathcal{B}_i, |\mathcal{J}_i|\}$ uniformly without replacement.

Algorithm 2.16 MAXRANDOMGENERATION-EBMC (uniform cardinality)

Require: County lakes $(\mathcal{J}_i)_{i \in N}$, budgets $(\mathcal{B}_i)_{i \in N}$
Ensure: Feasible profile with $\sum_{j \in \mathcal{J}_i} x_j \leq \mathcal{B}_i$ for all $i \in N$

- 1: **for each** $i \in N$ **do**
- 2: $b_i \leftarrow \min\{\mathcal{B}_i, |\mathcal{J}_i|\}$
- 3: $S \leftarrow \text{SAMPLEWITHOUTREPLACEMENT}(\mathcal{J}_i, b_i)$ $\triangleright$ uniform over $\binom{|\mathcal{J}_i|}{b_i}$
- 4: Set $x_j \leftarrow \mathbf{1}\{j \in S\}$ for all $j \in \mathcal{J}_i$
- 5:
- 6: **return** $\bar{\mathbf{x}} = (\mathbf{x}^i)_{i \in N}$

This generator runs in $O(|\mathcal{J}_i|)$ per county (via a shuffle-and-take routine) and samples uniformly among all maximal (cardinality- b_i) feasible strategies.

Full-support generator (Bernoulli + rejection). To obtain full support in EBMC, we again use a Bernoulli–rejection scheme. Here the natural density proxy is the ratio $\mathcal{B}_i/|\mathcal{J}_i|$:

Algorithm 2.17 RANDOMGENERATION-EBMC (full-support)

Require: County lakes $(\mathcal{J}_i)_{i \in N}$, budgets $(\mathcal{B}_i)_{i \in N}$
Ensure: Feasible profile with $\sum_{j \in \mathcal{J}_i} x_j \leq \mathcal{B}_i$ for all $i \in N$

- 1: **for each** $i \in N$ **do**
- 2: Choose p_i in $(0, 1)$, e.g. $p_i \leftarrow \mathcal{B}_i/|\mathcal{J}_i|$
- 3: **repeat**
- 4: **for each** $j \in \mathcal{J}_i$ **do**
- 5: Draw $x_j \sim \text{Ber}(p_i)$ independently
- 6:
- 7: **until** $\sum_{j \in \mathcal{J}_i} x_j \leq \mathcal{B}_i$
- 8:
- 9: **return** $\bar{\mathbf{x}} = (\mathbf{x}^i)_{i \in N}$

As in the KPG case, every feasible subset of $\mathcal{J}_i$ has positive probability under this procedure, and the expected number of rejection loops remains small for moderate p_i and budgets.

Remark. In summary, for both KPG and EBMC we alternate between maximal and full-support generators across restarts. Formally, let $\mu^{\max}$ and μ^{full} denote the restart laws on $\mathcal{X}$ induced when all restarts use the maximal generator and the full-support generator, respectively. If a PNE exists and the full-support generator is used (so the restart law is μ^{full}), then $q_{\mu^{\text{full}}}[R] > 0$ for all $R \geq 1$. In practice, however, the maximal generator often produces starting profiles that are closer to best responses and can empirically yield larger success probabilities ($q_{\mu^{\max}}[R] > q_{\mu^{\text{full}}}[R]$), even though these quantities are not computable in closed form. Alternating the two variants balances theoretical robustness (via the full-support generator) with the potential efficiency of more structured, capacity-filling starts.

2.D Extremely Small $q_\mu[R]$

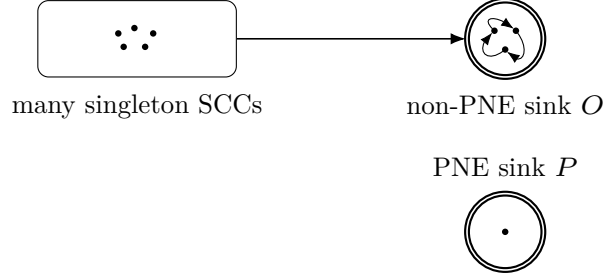


Figure 2.3: A condensation DAG showing that $q_\mu[R]$ can be extremely small.

Figure 2.3 highlights that the instance-dependent success probability $q_\mu[R]$ (2.4) in RRR-BRD can be extremely small even when a PNE exists. Assuming that the restart law μ has full support on the restart set and that R is sufficiently large, the diagram can be interpreted as the condensation DAG of the best-response state graph. In the schematic, O is an absorbing non-PNE sink SCC, P is an absorbing PNE sink SCC, and the remaining node(s) represent (possibly many) transient SCCs that flow into O . In particular, P may be a singleton and may be *unreachable* from any non-PNE start under best-response moves, so that a run succeeds only if the restart lands directly in P . Under full-support restart laws (e.g., uniform sampling over feasible profiles), the success probability $q_\mu[R]$ can therefore be made arbitrarily small by scaling the number of transient states (and hence their μ -mass) relative to P .

A knapsack-style IPG inducing vanishing $q_\mu[R]$. We now give a concrete IPG with a single knapsack constraint per player for which $q_\mu[R]$ decreases exponentially in M , matching the structure in Figure 2.3. Consider a two-player IPG ($N = \{1, 2\}$) with $m := M$ binary decisions indexed by $j \in \{1, \dots, M\}$. Each player i chooses a binary vector $\mathbf{x}^i \in \{0, 1\}^m$

subject to the single knapsack constraint

$$\sum_{j=1}^M x_j^i \leq M,$$

so every subset is feasible.

We use the standard IPG utility form

$$u_i(\mathbf{x}^i, \mathbf{x}^{-i}) = (\mathbf{d}^i)^\top \mathbf{x}^i + \sum_{p \in N \setminus \{i\}} (\mathbf{x}^p)^\top Q_p^i \mathbf{x}^i,$$

with $Q^i \equiv 0$ (no within-player quadratic term). Define the cyclic successor map on indices $g : \{1, \dots, M\} \rightarrow \{1, \dots, M\}$ by $g(j) = j + 1$ for $j = 1, \dots, M - 1$ and $g(M) = 1$.

Set linear coefficients $\mathbf{d}^1 = \mathbf{d}^2 = -c \mathbf{1}$ for some $c > 0$, and define the interaction matrices by the following nonzero entries (all others are 0):

$$Q_2^1[j, j] = T \quad (j = 1, \dots, M), \quad Q_1^2[g(j), j] = T \quad (j = 1, \dots, M),$$

for some $T > c$. Under these utilities, player 1's unique best response to $\mathbf{x}^2$ is to choose exactly the same subset $\mathbf{x}^1 = \mathbf{x}^2$, and player 2's unique best response to $\mathbf{x}^1$ is to choose the shifted subset $x_{g(j)}^2 = x_j^1$ for all j .

Identification of the sinks O and P and the transient region. The profile $(\mathbf{x}^1, \mathbf{x}^2) = (\mathbf{0}, \mathbf{0})$ is a PNE: each player obtains payoff 0, while any deviation to a nonzero subset yields negative payoff due to the linear cost. Thus the singleton $\{(\mathbf{0}, \mathbf{0})\}$ is the PNE sink P in Figure 2.3.

In contrast, no profile with $\mathbf{x}^1 \neq \mathbf{0}$ or $\mathbf{x}^2 \neq \mathbf{0}$ can be a PNE. If $\mathbf{x}^2 \neq \mathbf{0}$, then player 1's unique best response is $\mathbf{x}^1 = \mathbf{x}^2 \neq \mathbf{0}$. If $\mathbf{x}^1 \neq \mathbf{0}$, then player 2's unique best response is the

shifted subset $g(\mathbf{x}^1) \neq \mathbf{x}^1$, and hence $\mathbf{x}^2$ must be updated unless it already equals that shift. Restricting to the “active” profiles with $\mathbf{x}^1 \neq \mathbf{0}$ and $\mathbf{x}^2 \neq \mathbf{0}$, best-response updates cycle on the orbit induced by g , forming an absorbing strongly connected component O containing no PNE. All other non-PNE profiles are transient singleton SCCs that flow into O under improving best responses. This matches the schematic: a (possibly very large) transient region feeding into the non-PNE sink O , with a separate singleton PNE sink P .

Moreover, $(\mathbf{0}, \mathbf{0})$ is not reachable from any non-PNE start under improving best-response moves: when the opponent plays a nonzero subset, deviating to $\mathbf{0}$ yields payoff 0, whereas the prescribed best response yields strictly positive payoff because $T > c$. Thus, all non-PNE starts eventually enter O and remain there.

Implication for $q_\mu[R]$ under full-support restarts. Let the restart law μ have full support over the restart set, e.g., uniform over $\{0, 1\}^m \times \{0, 1\}^m$. In this instance, $P = \{(\mathbf{0}, \mathbf{0})\}$ is not reachable from any non-PNE start under improving best responses, whereas all other starts enter the non-PNE sink O and fail for any finite round cap R . Consequently,

$$q_\mu[R] = \mu(P) = \mu((\mathbf{0}, \mathbf{0})) = \frac{1}{2^{2M}},$$

which can be made arbitrarily small by increasing M . Equivalently, the expected number of restarts required to hit a PNE is $1/q_\mu[R] = 2^{2M}$, so RRR-BRD (Algorithm 2.4) can be exponentially weak in expectation on this instance.

Moreover, since the number of restarts until success is geometric, tail probabilities are non-negligible:

$$\Pr(R^* > r) = (1 - q_\mu[R])^r.$$

With $q_\mu[R] = 2^{-2M}$, we have

$$\Pr(R^* > 2^{2M}) = (1 - 2^{-2M})^{2^{2M}} \approx e^{-1},$$

so even at the mean scale 2^{2M} there remains a constant probability of requiring more restarts (and hence more work).

2.E Full Simulation Results

Tables [2.17–2.18](#) report full simulation results for the normal-form reductions and KPG instances. Most settings exhibit large $\tilde{q}_\mu[R]$ and small average rounds, while a minority of instances have much smaller $\tilde{q}_\mu[R]$, leading to larger ESB and longer runs; this instance dependence is most visible in the harder KPG configurations (notably type C).

Table 2.17: Full simulation results for normal-form reductions

n	k	inst	trials	successes	$\tilde{q}_\mu[R]$	avg_rounds_all	ESB ($\frac{nR}{q_\mu[R]}$)	Time (secs)
2	3	1	64	64	1.00	2.84	20.0	0.01
2	3	2	64	64	1.00	2.83	20.0	0.00
2	3	3	64	64	1.00	2.52	20.0	0.00
2	3	4	64	64	1.00	2.58	20.0	0.00
2	3	5	64	64	1.00	2.66	20.0	0.00
2	3	6	64	64	1.00	2.44	20.0	0.00
2	3	7	64	64	1.00	2.73	20.0	0.00
2	3	8	64	64	1.00	2.63	20.0	0.00
2	3	9	64	64	1.00	2.39	20.0	0.00
2	3	10	64	64	1.00	2.34	20.0	0.00
2	4	1	256	256	1.00	2.34	20.0	0.02
2	4	2	256	64	0.25	8.04	80.0	0.11
2	4	3	256	256	1.00	3.18	20.0	0.05
2	4	4	256	256	1.00	2.62	20.0	0.03
2	4	5	256	256	1.00	2.84	20.0	0.05
2	4	6	256	256	1.00	2.88	20.0	0.04
2	4	7	256	256	1.00	2.20	20.0	0.02
2	4	8	256	256	1.00	3.63	20.0	0.05
2	4	9	256	256	1.00	2.13	20.0	0.02
2	4	10	256	256	1.00	2.41	20.0	0.04
3	3	1	512	410	0.80	6.32	37.5	0.13
3	3	2	512	287	0.56	6.58	53.5	0.12
3	3	3	512	176	0.34	7.56	87.3	0.14
3	3	4	512	332	0.65	5.86	46.3	0.13
3	3	5	512	45	0.09	9.37	341.3	0.16
3	3	6	512	512	1.00	2.89	30.0	0.06
3	3	7	512	69	0.13	8.98	222.6	0.20
3	3	8	512	334	0.65	5.79	46.0	0.12
3	3	9	512	310	0.61	6.66	49.5	0.14
3	3	10	512	512	1.00	3.80	30.0	0.09
3	4	1	4096	4096	1.00	3.42	30.0	1.09
3	4	2	4096	2542	0.62	6.86	48.3	2.20
3	4	3	4096	612	0.15	9.31	200.8	2.95
3	4	4	4096	3173	0.77	6.04	38.7	1.92
3	4	5	4096	4096	1.00	3.41	30.0	1.16
3	4	6	4096	2542	0.62	6.47	48.3	2.10
3	4	7	4096	3566	0.87	4.65	34.5	1.59
3	4	8	4096	943	0.23	8.36	130.3	2.62
3	4	9	4096	4096	1.00	3.93	30.0	1.30
3	4	10	4096	3702	0.90	4.58	33.2	1.61

Table 2.18: Full simulation results for KPGs

type	n	items	budget	trials	successes	$\tilde{q}_\mu[R]$	avg_rounds_all	ESB ($\frac{nR}{q_\mu[R]}$)	Time (secs)
A	2	25	2	200	200	1.00	2.50	40.0	0.30
A	2	25	5	200	200	1.00	2.72	40.0	0.34
A	2	25	8	200	200	1.00	2.94	40.0	0.44
A	2	50	2	200	200	1.00	3.18	40.0	0.58
A	2	50	5	200	200	1.00	2.58	40.0	0.56
A	2	50	8	200	200	1.00	2.71	40.0	0.66
A	2	75	2	200	200	1.00	3.25	40.0	0.75
A	2	75	5	200	200	1.00	3.57	40.0	1.14
A	2	75	8	200	200	1.00	3.46	40.0	1.62
A	2	100	2	200	200	1.00	3.26	40.0	1.03
A	2	100	5	200	200	1.00	3.66	40.0	1.56
A	2	100	8	200	200	1.00	3.17	40.0	4.50
A	3	25	2	200	200	1.00	3.21	60.0	0.60
A	3	25	5	200	200	1.00	3.40	60.0	0.79
A	3	25	8	200	200	1.00	3.15	60.0	0.75
A	3	50	2	200	200	1.00	3.59	60.0	0.99
A	3	50	5	200	200	1.00	3.17	60.0	1.25
A	3	50	8	200	200	1.00	3.78	60.0	1.56
A	3	75	2	200	200	1.00	4.34	60.0	1.71
A	3	75	5	200	200	1.00	3.42	60.0	1.81
A	3	75	8	200	200	1.00	4.13	60.0	2.35
A	3	100	5	200	200	1.00	4.66	60.0	7.34
A	3	100	8	200	200	1.00	4.02	60.0	9.56
B	2	25	2	200	200	1.00	3.08	40.0	0.34
B	2	25	5	200	200	1.00	3.22	40.0	0.45
B	2	25	8	200	200	1.00	2.85	40.0	0.42
B	2	50	2	200	200	1.00	3.52	40.0	0.63
B	2	50	5	200	200	1.00	3.71	40.0	0.87
B	2	50	8	200	200	1.00	3.04	40.0	0.70
B	2	75	2	200	200	1.00	3.28	40.0	0.88
B	2	75	5	200	200	1.00	3.97	40.0	1.21
B	2	75	8	200	200	1.00	3.69	40.0	2.95
B	2	100	2	200	200	1.00	3.26	40.0	1.08
B	2	100	5	200	200	1.00	3.58	40.0	5.61
B	2	100	8	200	200	1.00	3.31	40.0	5.99
B	3	25	2	200	200	1.00	3.45	60.0	0.82
B	3	25	5	200	200	1.00	3.64	60.0	0.83
B	3	25	8	200	200	1.00	3.46	60.0	0.74
B	3	50	2	200	200	1.00	4.05	60.0	1.26
B	3	50	5	200	200	1.00	4.43	60.0	1.57
B	3	50	8	200	200	1.00	3.63	60.0	1.34
B	3	75	5	200	200	1.00	4.55	60.0	2.19
B	3	75	8	200	200	1.00	4.11	60.0	2.37
C	2	25	8	200	200	1.00	2.62	40.0	0.27
C	2	50	2	200	200	1.00	3.66	40.0	0.63
C	2	50	8	200	200	1.00	3.04	40.0	0.38
C	2	75	2	200	3	0.02	19.75	2666.7	5.36
C	2	75	8	200	74	0.37	13.85	108.1	2.92
C	2	100	2	200	156	0.78	7.53	51.3	2.59
C	3	25	2	200	1	0.01	19.92	12000.0	3.42

2.F Example for the BZR Algorithm

We exemplify how RRR-BRD, BZR, and ZR operate on a small two-player instance. In this example the performance gap is not pronounced, but the mechanics are transparent.

Player 1:
$$\max_{\mathbf{x}^1 \in \{0,1\}^3} 3x_1^1 + 2x_2^1 + 2x_3^1 - 4x_1^1x_1^2 - 5x_2^1x_2^2 - 3x_3^1x_3^2 \quad \text{s.t.} \quad x_1^1 + x_2^1 + x_3^1 \leq 2.$$

Player 2:
$$\max_{\mathbf{x}^2 \in \{0,1\}^3} 2x_1^2 + x_2^2 + 0x_3^2 + 5x_1^2x_1^1 + 2x_2^2x_2^1 + x_3^2x_3^1 \quad \text{s.t.} \quad x_1^2 + x_2^2 + x_3^2 \leq 2.$$

The game has a unique PNE $\hat{\mathbf{x}}_{\text{pne}}^* = \{(0, 0, 1), (1, 1, 0)\}$ with social welfare $\phi(\hat{\mathbf{x}}_{\text{pne}}^*) = 5$. We use $\mathbf{0}^N = \{(0, 0, 0), (0, 0, 0)\}$ as a starting profile.

RRR-BRD: Assume the (round-wise) random playing sequences are $[1, 2] \rightarrow [2, 1] \rightarrow [1, 2]$. Starting from $\mathbf{0}^N$, RRR-BRD follows the BR path

$$\{(0, 0, 0), (0, 0, 0)\} \xrightarrow{1} \{(1, 1, 0), (0, 0, 0)\} \xrightarrow{2} \{(1, 1, 0), (1, 1, 0)\} \xrightarrow{1} \{(0, 0, 1), (1, 1, 0)\},$$

and the final round $[1, 2]$ certifies this profile via the end-of-round no-change test.

BZR: In our implementation, BZR first calls the separation oracle at an incumbent integer solution. Suppose the MIP solver produces $\{(0, 0, 0), (0, 0, 0)\}$. This profile is not a PNE: player 1 has BR $(1, 1, 0)$ and player 2 has BR $(1, 1, 0)$, yielding two violated equilibrium inequalities (EIs), which are added:

$$5 - 4x_1^2 - 5x_2^2 \leq 3x_1^1 + 2x_2^1 + 2x_3^1 - 4z_1 - 5z_2 - 3z_3, \quad 3 + 5x_1^1 + 2x_2^1 \leq 2x_1^2 + x_2^2 + 0x_3^2 + 5z_1 + 2z_2 + z_3.$$

Since $\{(0, 0, 0), (0, 0, 0)\}$ is not a PNE, BZR then invokes RRR-BRD and finds the PNE

$\{(0, 0, 1), (1, 1, 0)\}$ under $[1, 2] \rightarrow [2, 1] \rightarrow [1, 2]$. Along this BRD trajectory, player 1's BRs are $(1, 1, 0)$ and $(0, 0, 1)$, and player 2's BR is $(1, 1, 0)$; thus player 1's BR $(0, 0, 1)$ provides an additional inequality:

$$2 - 3x_3^2 \leq 3x_1^1 + 2x_2^1 + 2x_3^1 - 4z_1 - 5z_2 - 3z_3.$$

After adding these inequalities, BZR resolves and terminates once the separation oracle confirms that the incumbent $\{(0, 0, 1), (1, 1, 0)\}$ (maximizing social welfare) is indeed the best PNE.

ZR: ZR first checks $\{(1, 0, 1), (1, 1, 0)\}$ (social welfare = 9), generating the violated EI from player 1's BR $(0, 0, 1)$:

$$2 - 3x_3^2 \leq 3x_1^1 + 2x_2^1 + 2x_3^1 - 4z_1 - 5z_2 - 3z_3.$$

Next it evaluates $\{(1, 1, 0), (1, 0, 1)\}$ (social welfare = 8), generating two violated EIs from player 1's BR $(0, 1, 0)$ and player 2's BR $(1, 1, 0)$:

$$2 - 5x_2^2 \leq 3x_1^1 + 2x_2^1 + 2x_3^1 - 4z_1 - 5z_2 - 3z_3,$$

$$3 + 5x_1^1 + 2x_2^1 \leq 2x_1^2 + x_2^2 + 0x_3^2 + 5z_1 + 2z_2 + z_3.$$

Finally, $\{(0, 0, 1), (1, 1, 0)\}$ is tested and confirmed as the best PNE.

Comparison: BZR requires one separation call at the initial incumbent, one RRR-BRD run, and three EIs in total (two from the initial separation and one from the BRD trajectory), resulting in two separation calls overall (including the final call that certifies the best PNE). ZR requires three separation calls and generates three EIs. This gap can widen sub-

stantially in large-scale IPGs where many separation rounds may be needed. We include this example for *illustrative* purposes; in the large-scale IPG instances studied in this chapter, the performance gap between BZR and ZR is often substantial.

2.G Exponential Separation Between ZR and BZR

This section gives a simple worst-case construction showing an exponential separation between ZR and BZR when *maximizing social welfare over PNEs* (i.e., the welfare-maximizing equilibrium problem). We use the n -fold product of the classical Prisoner's Dilemma and the standard welfare $\phi = u_1 + u_2$ to illustrate that there can be exponentially many high-welfare non-equilibria, forcing welfare-guided equilibrium search to reject many candidates before certifying the optimal PNE. (We use $O(\cdot)$ for upper bounds and $\Theta(\cdot)$ when the bound is tight up to constant factors.)

Construction. Two players choose $\mathbf{x}, \mathbf{y} \in \{0, 1\}^n$. Each coordinate $j \in \{1, \dots, n\}$ is an independent Prisoner's Dilemma, where 0 = Cooperate and 1 = Defect, with per-coordinate payoffs shown in Table 2.19.

Table 2.19: Per-coordinate payoffs for the Prisoner's Dilemma component game in $\mathcal{P}_n$.

	$y_j = 0$ (C)	$y_j = 1$ (D)
$x_j = 0$ (C)	(3, 3)	(0, 5)
$x_j = 1$ (D)	(5, 0)	(1, 1)

Note. (0 = cooperate, 1 = defect)

Structure. Defection strictly dominates cooperation in each coordinate for both players. Hence the unique PNE is $(\mathbf{1}, \mathbf{1})$ with $\phi(\mathbf{1}, \mathbf{1}) = 2n$, while the social optimum is $(\mathbf{0}, \mathbf{0})$ with

$\phi(\mathbf{0}, \mathbf{0}) = 6n$. Moreover, every profile other than $(\mathbf{1}, \mathbf{1})$ has strictly larger welfare, so exactly $4^n - 1$ profiles satisfy $\phi > 2n$.

Complexity gap. ZR must reject all $4^n - 1$ non-PNE profiles with $\phi > 2n$ before it can certify that $(\mathbf{1}, \mathbf{1})$ is the optimal PNE, yielding $\Theta(4^n)$ separation-oracle calls in the worst case. By contrast, BZR finds the PNE in $O(1)$ best-response steps (player 1 best-responds to $\mathbf{1}$, then player 2 best-responds to $\mathbf{1}$). Verification is still exponential and in fact $\Theta(2^n)$ in this construction: because defection is dominant, equilibrium inequalities are essentially non-restrictive and do not cut off high-welfare non-PNE profiles, so BZR must rule out $(\mathbf{1}, \mathbf{y})$ for each $\mathbf{y} \neq \mathbf{1}$. There are $2^n - 1$ such $\mathbf{y}$, and for each one player 2's best response to $\mathbf{1}$ is $\mathbf{1} \neq \mathbf{y}$, hence $(\mathbf{1}, \mathbf{y})$ is not a PNE. Thus verification requires $\Theta(2^n)$ checks and the overall separation is $\Theta(2^n)$.

Table 2.20: Worst-case complexity comparison on $\mathcal{P}_n$: ZR versus BZR.

Algorithm	Finding	Verification	Total
ZR	–	(included)	$\Theta(4^n)$
BZR	$O(1)$	$\Theta(2^n)$	$\Theta(2^n)$
Gap:			$\Theta(2^n)$

Takeaway. This dominant-strategy example shows that certifying the welfare-optimal PNE can be exponentially harder than finding a PNE.

2.H Illustration: A Two-Player Selfish EBMC Game

Figure 2.4 illustrates the interplay between two selfish players, each controlling three vertices and selecting one under a budget constraint of one. The bold circles and the dashed lines

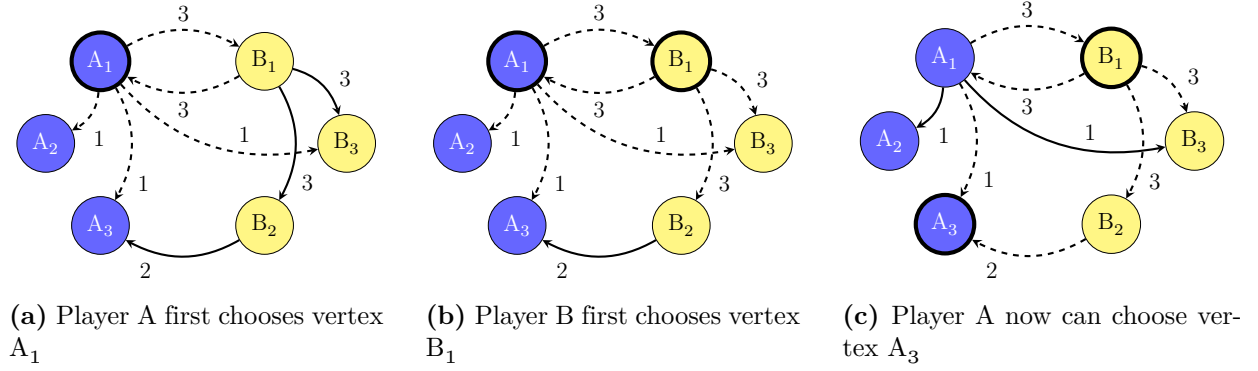


Figure 2.4: The interactions in the selfish game with two players.

represent the selected vertices and the edges covered by the choices of players, respectively. For example, in Figure 2.4a, edges from B_1 to A_1 are covered at vertex A_1 , while those from B_1 to B_2 are not covered. Assume that the vertices differ in status between two task types. Following standard BRD, Player A first chooses vertex A_1 . Player B benefits from this choice because the edge (A_1, B_1) is now covered. Subsequently, Player B chooses vertex B_1 , covering additional arcs. This in turn activates the edge (B_1, A_1) , allowing Player A to consider selecting vertex A_3 as a new best response. Player B maintains the same choice. At this point, the game reaches a PNE, as neither player has an incentive to deviate from their current strategy. This example illustrates how one player's best response may incidentally benefit the other, even in a selfish setting. However, these benefits are not intentional but rather emergent consequences of self-interested decisions. Importantly, mutual gains observed during the BRD do not guarantee monotonic increases in utility for all players. For example, Player B benefits from Player A's initial choice of A_1 , but the subsequent choice of A_3 does not influence Player B's utility, so that it continues to prefer vertex B_1 . From Player B's perspective, Player A's earlier strategy (choosing A_1 alone) yields a higher selfish objective value than the final strategy (choosing A_3).

2.I EBMC as an IPG: Reformulation and Complexity Implications

The objective functions of the locally altruistic and selfish EBMC games can be reformulated in the standard (quadratic) IPG form by eliminating the edge variables $\mathbf{y}$ and expressing utilities purely in terms of vertex variables $\mathbf{x}$. Throughout, vertex decisions are binary ($x_j \in \{0, 1\}$), and edge coverage is defined by $y_{jk} = \max\{x_j, x_k\}$. Hence, for every arc (j, k) we have the equation $y_{jk} = x_j + x_k - x_j x_k$. Using this equation, for each player $i \in N$ we can write the EBMC utility as

$$u_i(\mathbf{x}^i, \mathbf{x}^{-i}) = (\mathbf{x}^i)^\top Q^i \mathbf{x}^i + \sum_{p \in N \setminus \{i\}} (\mathbf{x}^p)^\top Q_p^i \mathbf{x}^i + (\mathbf{d}^i)^\top \mathbf{x}^i + c_i(\mathbf{x}^{-i}),$$

where Q^i captures within-player quadratic terms, Q_p^i captures cross-player interactions (rows indexed by variables in $\mathbf{x}^p$ and columns by variables in $\mathbf{x}^i$), and $\mathbf{d}^i$ captures linear terms in $\mathbf{x}^i$. The last summation collects all terms that depend only on opponents' decisions $\mathbf{x}^{-i}$. Since $c_i(\mathbf{x}^{-i})$ does not depend on $\mathbf{x}^i$, it is an additive constant from player i 's perspective and therefore does not affect best responses or equilibria.

Coefficient construction. The reformulation follows by expressing the three core utility components $\sum_{(j,k) \in \mathcal{A}_i} w_{jk} y_{jk}$, $\sum_{(j,k) \in \delta_+(\mathcal{I}_i)} w_{jk} y_{jk}$, $\sum_{(j,k) \in \delta_-(\mathcal{I}_i)} w_{jk} y_{jk}$ in terms of $\mathbf{x}$ via $y_{jk} = x_j + x_k - x_j x_k$. Below we describe how each arc set contributes to the coefficients in $u_i(\mathbf{x}^i, \mathbf{x}^{-i})$. (As usual, if one stores Q^i as a symmetric matrix, then a term $-w_{jk} x_j x_k$ may be split across (j, k) and (k, j) to avoid double counting.)

- **Internal arcs.** For an internal arc $(j, k) \in \mathcal{A}_i$ (so $j, k \in \mathcal{I}_i$), the contribution $w_{jk}(x_j + x_k - x_j x_k)$ adds w_{jk} to the j -th and k -th entries of $\mathbf{d}^i$, and adds $-w_{jk}$ to the (j, k) entry of Q^i (representing the bilinear term $-w_{jk} x_j x_k$).

- **Outgoing arcs.** For an outgoing arc $(j, k) \in \delta_+(\mathcal{J}_i)$ with $j \in \mathcal{J}_i$ and $k \in \mathcal{J}_p$ for some $p \neq i$, the contribution $w_{jk}(x_j + x_k - x_j x_k)$ adds w_{jk} to the j -th entry of $\mathbf{d}^i$ (the x_j term), adds the bilinear coefficient $-w_{jk}$ to the (k, j) entry of Q_p^i (row indexed by x_k , column indexed by x_j), and adds the opponent-only linear term $w_{jk}x_k$ into $c_i(\mathbf{x}^{-i})$.
- **Incoming arcs.** For an incoming arc $(j, k) \in \delta_-(\mathcal{J}_i)$ with $j \in \mathcal{J}_p$ for some $p \neq i$ and $k \in \mathcal{J}_i$, the contribution $w_{jk}(x_j + x_k - x_j x_k)$ adds w_{jk} to the k -th entry of $\mathbf{d}^i$ (the x_k term), adds the bilinear coefficient $-w_{jk}$ to the (j, k) entry of Q_p^i (row indexed by x_j , column indexed by x_k), and adds the opponent-only linear term $w_{jk}x_j$ into $c_i(\mathbf{x}^{-i})$.

Applying these transformations to the appropriate arc sets yields an explicit representation of EBMC games in the standard (quadratic) IPG format. In particular, the interaction structure of EBMC, captured by the family $\{Q_p^i\}_{p \neq i}$, mirrors that of quadratic IPGs (qIPGs).

Finally, although $u_i(\mathbf{x}^i, \mathbf{x}^{-i})$ includes the term $c_i(\mathbf{x}^{-i})$, this term depends only on $\mathbf{x}^{-i}$ and therefore does not affect player i 's best response; it can be safely ignored when computing best responses or separating equilibrium inequalities.

Complexity considerations. Both the selfish and locally altruistic EBMC games are instances of quadratic integer programming games (qIPGs): each player solves a binary optimization problem over an individual feasible set $\mathcal{X}_i$, and utilities admit the qIPG representation. Moreover, any term depending only on opponents' decisions can be absorbed into $c_i(\mathbf{x}^{-i})$ and therefore does not affect best responses or equilibria. Carvalho et al. [35] show that deciding whether a general IPG admits a pure Nash equilibrium (PNE) is Σ_2^P -complete; consequently, the PNE-existence problem restricted to EBMC games lies in Σ_2^P . In the locally altruistic EBMC game, we establish that the game is an exact potential game and hence a PNE always exists (each $\mathcal{X}_i$ is finite and nonempty). For the selfish EBMC game, PNE

existence is not guaranteed; while the decision problem lies in Σ_2^P , establishing Σ_2^P -hardness (and thus completeness) for the EBMC subclass would require a dedicated reduction, which we leave for future work.

2.J Counterexamples for the Existence of PNE

This section presents counterexamples for the existence of a PNE in the EBMC games $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$ and $\mathcal{G}_{|\mathcal{L}|\geq 2}^{\text{Self}}$. For $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$, we identify a counterexample using modest integer weights between 1 and 10 with a total of 9 vertices and two players with budgets 2 and 1. Table 2.21 shows that there is a unilateral selfish deviation for any strategy profile. We demonstrate the existence of a selfish deviation for any strategy profile rather than showing cycles, as the latter case does not conclusively indicate the absence of a PNE.

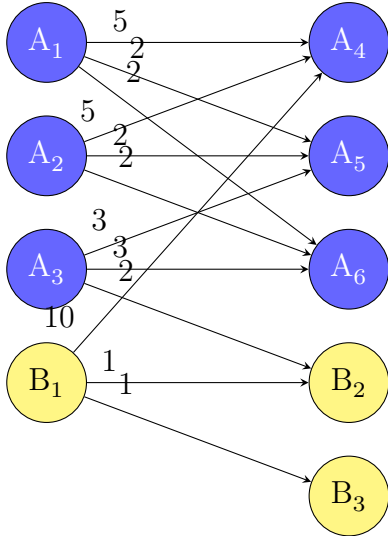


Figure 2.5: No PNE for $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$.

Table 2.21: Strategy Profiles and Selfish Deviations.

Strategy profiles	Selfish unilateral deviation
$x_{A_3} = 1, \mathbf{x}^B \neq (1, 0, 0)$	$\hat{\mathbf{x}}^B = (1, 0, 0)$
$x_{A_3} \neq 1, \mathbf{x}^B \neq (0, 1, 0)$	$\hat{\mathbf{x}}^B = (0, 1, 0)$
$x_{B_1} = 1, \mathbf{x}^A \neq (1, 1, 0, 0, 0, 0)$	$\hat{\mathbf{x}}^A = (1, 1, 0, 0, 0, 0)$
$x_{B_1} \neq 1, \mathbf{x}^A \neq (0, 0, 1, 1, 0, 0)$	$\hat{\mathbf{x}}^A = (0, 0, 1, 1, 0, 0)$

This table displays all potential strategy profiles for players A and B. A has a budget of 2 while B has a budget of 1. Each row describes a set of strategy profiles. The first row contains the strategy profile $\mathbf{x}^A = (0, 0, 1, 1, 0, 0)$ and $\mathbf{x}^B = (0, 1, 0)$. In this case, player B would deviate to $\hat{\mathbf{x}}^B = (1, 0, 0)$. In another scenario, if $\mathbf{x}^A = (1, 0, 1, 0, 0, 0)$ and $\mathbf{x}^B = (0, 1, 0)$, possible deviations are $\hat{\mathbf{x}}^B = (1, 0, 0)$ for player B and $\hat{\mathbf{x}}^A = (0, 0, 1, 1, 0, 0)$ for player A.

Compared to $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$, counterexamples in $\mathcal{G}_{|\mathcal{L}|\geq 2}^{\text{Self}}$ are more readily identifiable. An example is presented in Figure 2.6, where the vertices A_1 and A_2 are of type-1, and the vertices B_1 and B_2 are of type-2. Assuming that each player has only one budget, no PNE exists; for every

possible strategy profile, there is a selfish unilateral deviation, as detailed in Table 2.22.

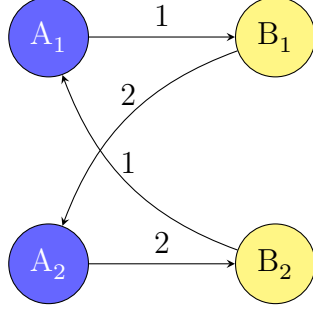


Figure 2.6: No PNE for $\mathcal{G}_{|\mathcal{L}|\geq 2}^{\text{Self}}$.

Table 2.22: Strategy Profiles and Selfish Deviations.

Strategy profiles	Selfish unilateral deviation
$x_{A_1} = 1, \mathbf{x}^B \neq (0, 1)$	$\hat{\mathbf{x}}^B = (0, 1)$
$x_{A_1} \neq 1, \mathbf{x}^B \neq (1, 0)$	$\hat{\mathbf{x}}^B = (1, 0)$
$x_{B_1} = 1, \mathbf{x}^A \neq (1, 0)$	$\hat{\mathbf{x}}^A = (1, 0)$
$x_{B_1} \neq 1, \mathbf{x}^A \neq (0, 1)$	$\hat{\mathbf{x}}^A = (0, 1)$

This table displays all potential strategy profiles for players A and B. For instance, if $\mathbf{x}^A = (1, 0)$ and $\mathbf{x}^B = (1, 0)$, player B would deviate to $\hat{\mathbf{x}}^B = (0, 1)$.

2.K RRR-BRD Results

We present the full results of the RRR-BRD algorithm using different initial strategy profiles in various problem settings. Specifically, we report results for selfish EBMC games with $|\mathcal{L}| = 1$ and $|\mathcal{L}| \geq 2$, as well as KPG instances with interaction types (A), (B), and (C). The metrics and computational settings used in these experiments are described in Section 2.7.1.

Table 2.23: RRR-BRD results with different initial strategy profiles for selfish EBMC games $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$.

n	BG	RRR-BRD (ϕ)		Price of Stability		References from BZR		Time (secs)		Round	
		init: $\mathbf{0}^N$	init: $\tilde{\mathbf{x}}_{\text{sw}}^{300}$	$\widetilde{\text{OSW}}$	$\widetilde{\text{POS}}$	BZR(ϕ)	BZR(Bound)	init: $\mathbf{0}^N$	init: $\tilde{\mathbf{x}}_{\text{sw}}^{300}$	init: $\mathbf{0}^N$	init: $\tilde{\mathbf{x}}_{\text{sw}}^{300}$
2	0.3	19.2	19.2	19.2	1.000	19.2	19.2	0.0	0.1	2	1
	0.5	21.6	22.3	22.3	1.000	22.3	22.3	0.1	2.3	2	1
	0.8	22.6	22.6	24.9	1.102	23.3	23.3	0.1	0.4	2	2
3	0.3	22.0	27.6	27.6	1.000	27.6	27.6	0.2	2.4	2	1
	0.5	38.8	42.1	42.1	1.000	42.1	42.1	0.2	6.9	2	1
	0.8	53.1	53.1	53.1	1.000	53.1	53.1	0.1	0.2	2	1
5	0.3	83.5	90.4	90.4	1.000	90.4	90.4	0.4	13.8	2	1
	0.5	85.0	100.6	100.6	1.000	100.6	101.2	0.5	301.0	2	1
	0.8	146.9	156.7	156.7	1.000	156.7	157.4	0.5	301.1	2	1
8	0.3	130.1	130.1	149.9	1.152	134.1	134.2	1.6	5.3	2	3
	0.5	169.1	252.2	252.2	1.000	252.2	260.2	1.6	302.8	2	1
	0.8	269.9	341.4	381.3	1.117	349.2	353.4	1.4	232.6	2	3
10	0.3	308.1	317.2	317.2	1.000	317.2	317.3	2.6	304.5	2	1
	0.5	289.8	361.7	361.7	1.000	361.7	372.4	1.6	302.8	2	1
	0.8	412.2	539.8	539.8	1.000	538.1	549.3	2.0	303.8	2	1
15	0.3	630.4	725.4	725.4	1.000	725.4	745.9	4.7	307.8	2	1
	0.5	576.9	576.9	857.5	1.487	708.4	730.1	6.1	23.7	2	3
	0.8	886.3	1212.2	1219.7	1.006	1216.5	1249.2	5.7	311.3	2	3
20	0.3	656.1	904.5	904.5	1.000	904.5	922.5	10.4	319.2	2	1
	0.5	1089.0	1398.8	1529.1	1.093	1282.1	1472.8	11.1	319.8	2	1
	0.8	1317.6	1701.4	2415.5	1.420	2024.4	2207.2	13.1	323.9	2	3
25	0.3	1419.5	1765.0	1765.0	1.000	1765.0	1819.5	18.0	328.9	2	1
	0.5	2140.7	2597.5	2597.7	1.000	2597.4	2767.9	19.0	334.7	2	2
	0.8	2246.8	3294.0	3312.8	1.006	3307.4	3541.1	18.9	336.4	2	3
30	0.3	1780.2	2448.2	2448.3	1.000	2448.1	2556.0	26.1	345.6	2	2
	0.5	2704.9	3263.8	3263.8	1.000	3260.8	3482.4	29.8	349.1	2	1
	0.8	3724.0	4919.7	4919.7	1.000	4910.4	5399.1	28.1	346.4	2	1

Table 2.24: RRR-BRD results with different initial strategy profiles for selfish EBMC games $\mathcal{G}_{|\mathcal{L}| \geq 2}^{\text{Self}}$.

n	BG	RRR-BRD (ϕ)		Price of Stability		References from BZR		Time (secs)		Round	
		init: $\mathbf{0}^N$	init: $\tilde{\mathbf{x}}_{\text{sw}}^{300}$	$\widetilde{\text{OSW}}$	$\widetilde{\text{POS}}$	BZR(ϕ)	BZR(Bound)	init: $\mathbf{0}^N$	init: $\tilde{\mathbf{x}}_{\text{sw}}^{300}$	init: $\mathbf{0}^N$	init: $\tilde{\mathbf{x}}_{\text{sw}}^{300}$
2	0.3	51.4	51.4	56.3	1.096	51.4	51.4	0.4	1.6	3	3
	0.5	83.8	83.8	86.6	1.033	83.8	86.0	0.8	58.8	3	2
	0.8	92.1	91.4	95.2	1.034	92.8	94.3	1.0	5.6	3	2
3	0.3	115.0	115.0	139.7	1.215	115.0	115.0	0.7	12.1	4	4
	0.5	178.6	178.3	200.3	1.122	178.6	196.7	0.7	301.5	4	4
	0.8	252.2	252.2	253.7	1.006	252.2	257.8	0.8	301.8	4	6
5	0.3	380.0	380.0	399.8	1.052	380.0	380.0	1.7	303.5	3	3
	0.5	557.6	557.6	577.3	1.035	557.6	618.5	1.7	303.1	3	3
	0.8	650.1	650.1	661.9	1.018	650.1	677.5	2.1	303.9	4	4
8	0.3	862.1	862.1	995.8	1.155	862.1	980.4	5.2	309.4	4	4
	0.5	1214.5	1214.5	1387.9	1.143	1214.5	1535.1	5.6	309.2	4	4
	0.8	1661.5	1661.5	1721.0	1.036	1661.5	1771.3	6.1	309.3	4	6
10	0.3	1265.2	1265.2	1521.6	1.203	1265.2	1526.5	8.0	319.6	3	13
	0.5	2084.2	2084.2	2253.9	1.081	2083.2	2635.7	8.7	315.5	4	5
	0.8	2612.3	2612.3	2685.2	1.028	2612.3	2757.3	9.4	318.6	5	9
15	0.3	2916.4	2916.4	3440.2	1.180	2916.4	3523.7	19.5	329.0	4	4
	0.5	4476.8	4476.8	5030.1	1.124	4476.8	5750.7	29.3	337.9	6	5
	0.8	-	-	6206.3	-	-	6428.7	402.9	687.6	200	200
20	0.3	5373.6	5373.6	6284.2	1.169	5373.6	6483.1	44.9	366.4	5	4
	0.5	-	-	9019.4	-	-	11452.5	874.9	1195.1	200	200
	0.8	10149.3	10149.7	10510.8	1.036	10149.7	10858.2	45.7	377.6	4	6
25	0.3	8023.9	8023.9	9502.9	1.184	8023.9	17369.6	51.8	394.3	3	4
	0.5	12640.7	12640.7	13936.0	1.102	12640.7	17424.1	77.7	417.1	4	4
	0.8	-	-	16806.1	-	-	17521.5	1789.4	1949.1	200	200
30	0.3	11151.4	11151.4	13520.2	1.212	11151.4	24217.3	119.6	460.2	5	4
	0.5	17342.4	17342.4	16969.0	0.978	17342.4	25103.4	146.5	503.8	6	6
	0.8	-	-	24180.9	-	-	25028.6	2910.7	3240.5	200	200

Table 2.25: RRR-BRD results with different initial strategy profiles for Type A KPG.

n	BG	RRR-BRD (ϕ)		Price of Stability		References from BZR		Time (secs)		Round	
		init: $\mathbf{0}^N$	init: $\tilde{\mathbf{x}}_{\text{sw}}^{300}$	$\widetilde{\text{OSW}}$	$\widetilde{\text{POS}}$	BZR(ϕ)	BZR(Bound)	init: $\mathbf{0}^N$	init: $\tilde{\mathbf{x}}_{\text{sw}}^{300}$	init: $\mathbf{0}^N$	init: $\tilde{\mathbf{x}}_{\text{sw}}^{300}$
2	0.2	18.4	18.5	18.7	1.011	18.5	18.5	0.1	0.6	3	2
	0.5	17.6	18.3	18.3	1.002	18.3	18.3	0.0	0.4	3	2
	0.8	8.1	8.2	8.4	1.022	8.3	8.3	0.0	0.4	3	2
3	0.2	37.2	37.5	37.8	1.010	37.6	37.6	0.3	3.1	4	3
	0.5	31.8	32.4	32.8	1.011	32.7	32.7	0.1	2.0	3	3
	0.8	14.5	14.8	15.2	1.027	14.8	14.8	0.0	0.4	4	2
5	0.2	129.2	131.3	131.3	1.000	131.3	131.3	0.9	47.0	4	1
	0.5	59.6	60.0	61.4	1.024	60.6	61.0	0.8	33.5	7	3
	0.8	27.0	27.2	29.3	1.079	27.3	27.3	0.1	2.0	4	3
8	0.2	289.8	294.8	295.1	1.001	295.1	295.4	4.0	301.1	8	2
	0.5	110.5	112.1	116.7	1.041	112.4	114.0	0.9	235.1	6	5
	0.8	85.6	86.6	87.8	1.013	86.9	87.6	0.3	25.8	7	4
10	0.2	500.5	504.6	508.7	1.008	505.5	506.1	6.1	302.5	8	3
	0.5	255.0	260.4	262.8	1.009	260.9	263.0	1.0	302.3	7	4
	0.8	139.5	141.6	143.9	1.016	141.2	142.9	0.5	301.3	7	5
15	0.2	1050.2	1057.4	1058.8	1.001	1058.1	1060.8	6.5	302.8	8	3
	0.5	583.1	599.7	604.2	1.007	600.4	605.2	1.9	304.1	9	6
	0.8	246.4	254.2	257.7	1.014	255.3	258.3	0.8	302.4	8	6
20	0.2	1663.5	1688.0	1689.9	1.001	1689.5	1693.9	8.6	303.2	10	4
	0.5	1103.4	1120.6	1126.2	1.005	1120.4	1129.0	2.2	302.8	13	7
	0.8	598.6	610.2	612.8	1.004	610.8	615.2	1.2	302.6	15	4
25	0.2	2597.6	2626.4	2628.1	1.001	2627.4	2637.1	8.3	306.1	7	3
	0.5	1789.9	1825.8	1833.3	1.004	1827.5	1838.4	3.7	307.2	17	8
	0.8	690.1	706.2	711.5	1.008	706.6	714.5	1.4	303.8	11	5
30	0.2	3311.2	3334.8	3351.2	1.005	3331.6	3356.8	15.1	307.4	10	4
	0.5	2490.0	2528.8	2535.3	1.003	2528.7	2544.7	3.3	305.2	14	6
	0.8	1070.2	1095.9	1104.6	1.008	1095.7	1109.7	1.4	304.4	8	6

Table 2.26: RRR-BRD results with different initial strategy profiles for Type B KPG.

n	BG	RRR-BRD (ϕ)		Price of Stability		References from BZR		Time (secs)		Round	
		init: $\mathbf{0}^N$	init: $\tilde{\mathbf{x}}_{\text{sw}}^{300}$	$\widetilde{\text{OSW}}$	$\widetilde{\text{POS}}$	BZR(ϕ)	BZR(Bound)	init: $\mathbf{0}^N$	init: $\tilde{\mathbf{x}}_{\text{sw}}^{300}$	init: $\mathbf{0}^N$	init: $\tilde{\mathbf{x}}_{\text{sw}}^{300}$
2	0.2	8.0	8.2	8.3	1.018	8.2	8.2	0.0	0.1	3	2
	0.5	14.1	14.2	14.4	1.016	14.2	14.2	0.2	0.3	3	3
	0.8	18.8	19.0	19.1	1.005	19.1	19.1	0.2	0.3	3	3
3	0.2	14.8	15.4	15.9	1.035	15.6	15.6	0.0	0.3	5	3
	0.5	28.3	28.7	29.4	1.025	28.9	29.1	0.1	2.8	5	4
	0.8	39.0	39.8	40.3	1.012	40.0	40.1	0.9	2.6	4	2
5	0.2	39.6	39.6	40.9	1.033	40.4	40.8	0.2	5.8	6	5
	0.5	76.7	77.9	78.8	1.012	78.1	78.8	0.6	19.5	9	5
	0.8	106.9	107.7	108.7	1.009	108.3	108.6	4.0	24.9	10	3
8	0.2	87.7	91.1	92.6	1.017	91.6	92.7	0.4	253.4	5	4
	0.5	185.2	187.5	189.9	1.013	188.6	189.9	1.8	184.4	13	7
	0.8	267.6	272.3	273.1	1.003	272.8	273.2	4.5	302.5	6	4
10	0.2	132.5	135.4	136.9	1.011	136.0	137.0	0.1	300.6	10	3
	0.5	281.0	286.8	289.0	1.008	287.6	289.4	0.7	302.0	7	4
	0.8	423.2	426.1	427.8	1.004	426.9	428.3	5.6	302.9	9	4
15	0.2	279.2	285.8	289.5	1.013	287.0	290.5	0.6	302.5	7	7
	0.5	627.1	641.3	644.0	1.004	642.7	645.4	1.1	302.1	9	6
	0.8	954.7	963.8	965.2	1.001	963.8	966.5	6.6	303.1	9	3
20	0.2	479.9	486.6	491.2	1.009	488.7	493.1	0.7	301.9	10	4
	0.5	1108.2	1121.6	1125.5	1.004	1123.8	1129.0	1.0	302.0	9	4
	0.8	1661.3	1671.9	1672.9	1.001	1670.7	1677.3	8.4	302.7	11	2
25	0.2	753.5	759.8	763.5	1.005	761.4	767.5	0.9	303.4	9	4
	0.5	1696.0	1717.7	1723.7	1.004	1719.9	1728.6	4.0	304.2	20	8
	0.8	2586.8	2608.5	2614.3	1.002	2610.0	2622.9	12.1	306.5	13	5
30	0.2	1055.2	1085.3	1088.8	1.003	1085.2	1094.7	2.3	305.8	12	14
	0.5	2409.3	2460.4	2465.8	1.002	2460.3	2473.3	3.8	305.1	13	5
	0.8	3729.8	3745.0	3750.8	1.002	3749.0	3764.0	17.0	310.2	11	6

Table 2.27: RRR-BRD results with different initial strategy profiles for Type C KPG.

n	BG	RRR-BRD (ϕ)		Price of Stability		References from BZR		Time (secs)		Round	
		init: $\mathbf{0}^N$	init: $\tilde{\mathbf{x}}_{\text{sw}}^{300}$	$\widetilde{\text{OSW}}$	$\widetilde{\text{POS}}$	BZR(ϕ)	BZR(Bound)	init: $\mathbf{0}^N$	init: $\tilde{\mathbf{x}}_{\text{sw}}^{300}$	init: $\mathbf{0}^N$	init: $\tilde{\mathbf{x}}_{\text{sw}}^{300}$
2	0.2	4.7	4.7	4.7	1.007	4.7	4.7	0.0	0.1	3	2
	0.5	7.8	7.8	7.8	1.007	7.8	7.8	0.0	0.1	3	2
	0.8	8.4	8.4	8.4	1.008	8.4	8.4	0.1	0.1	3	2
3	0.2	7.1	7.1	7.2	1.012	7.1	7.1	0.0	0.3	4	3
	0.5	10.3	10.3	10.4	1.011	10.3	10.3	0.1	0.3	4	2
	0.8	11.5	11.6	11.8	1.024	11.6	11.6	0.1	0.1	5	3
5	0.2	10.1	10.1	10.4	1.025	10.3	10.3	0.1	0.7	3	3
	0.5	14.2	14.1	14.6	1.028	14.3	14.4	0.5	1.6	4	3
	0.8	13.6	13.7	15.0	1.095	13.7	13.7	0.3	0.4	4	3
8	0.2	15.6	15.6	16.2	1.034	15.8	16.0	0.1	11.4	4	3
	0.5	15.6	15.6	17.6	1.122	15.8	16.5	0.5	1.9	6	4
	0.8	15.6	15.5	18.5	1.192	15.7	16.6	0.2	1.3	5	4
10	0.2	16.3	16.4	17.5	1.070	16.7	17.3	0.1	13.1	6	10
	0.5	16.7	16.7	20.3	1.212	16.9	18.3	6.2	16.4	85	126
	0.8	16.1	16.1	20.3	1.262	16.3	18.0	0.3	4.5	6	5
15	0.2	19.6	19.6	21.8	1.114	19.8	21.5	0.5	224.2	8	17
	0.5	15.9	16.1	23.0	1.435	16.5	19.8	0.4	72.6	10	7
	0.8	16.3	16.4	23.0	1.398	16.8	19.8	0.3	42.8	9	8
20	0.2	20.5	20.7	24.5	1.185	20.9	24.0	4.3	304.7	77	54
	0.5	-	-	24.7	-	-	22.0	6.4	308.5	200	200
	0.8	16.8	16.8	25.5	1.519	17.1	23.1	5.0	303.7	131	50
25	0.2	20.6	20.4	26.2	1.272	20.7	26.4	15.1	303.9	196	20
	0.5	-	-	25.7	-	-	24.9	10.2	313.0	200	200
	0.8	-	-	26.4	-	-	25.4	11.5	315.3	200	200
30	0.2	20.6	-	27.7	1.350	20.5	40.8	17.7	320.2	196	200
	0.5	15.9	15.7	27.3	1.720	16.3	33.0	4.6	305.5	74	33
	0.8	15.6	-	27.6	1.772	15.8	30.4	2.9	323.8	78	200

2.L BZR Full Results

We present the full BZR results of the EBMC $|\mathcal{L}| \geq 2$ games, KPG types (A) and (C). The metrics and computational settings used in these experiments are described in Section 2.7.1.

Table 2.28: Comparison of BZR and ZR for selfish EBMC games $\mathcal{G}_{|\mathcal{L}| \geq 2}^{\text{Self}}$.

n	BG	# PNEs Found		Best PNE (ϕ)		Dual Bounds		Price of Stability		Time (secs)				# EI Cuts	
		# _{pne(ZR)}	# _{pne(BZR)}	ZR	BZR	ZR	BZR	$\overline{\text{OSW}}$	$\overline{\text{POS}}$	T(ZR ^{1st})	T(ZR)	T(BZR ^{1st})	T(BZR)	# _{cuts(ZR)}	# _{cuts(BZR)}
2	0.3	1	1	51.4	51.4	51.4	51.4	56.3	1.096	9.7	9.8	0.8	12.8	12	19
	0.5	0	1	-	83.8	85.7	86.0	86.6	1.033	-	1800.3	1.4	1800.1	11	20
	0.8	0	6	-	92.8	94.1	94.3	95.2	1.027	-	1800.1	2.1	1800.1	14	26
3	0.3	1	1	115.0	115.0	115.0	115.0	139.7	1.215	1423.4	1423.7	1.9	1040.1	18	29
	0.5	0	2	-	178.6	197.9	196.7	200.3	1.122	-	1800.4	2.3	1800.5	15	29
	0.8	0	1	-	252.2	257.7	257.8	253.7	1.006	-	1800.3	1.9	1800.4	15	32
5	0.3	1	1	380.0	380.0	380.0	380.0	399.8	1.052	489.2	665.9	5.7	642.9	28	49
	0.5	0	1	-	557.6	626.7	618.5	577.3	1.035	-	1800.3	5.6	1800.0	25	43
	0.8	0	1	-	650.1	671.1	677.5	661.9	1.018	-	1800.7	5.4	1800.5	25	57
8	0.3	0	1	-	862.1	975.9	980.4	995.8	1.155	-	1800.1	14.8	1800.1	40	70
	0.5	0	1	-	1214.5	1527.4	1535.1	1387.9	1.143	-	1800.1	14.4	1800.1	40	75
	0.8	0	1	-	1661.5	1770.3	1771.3	1721.0	1.036	-	1800.1	14.8	1800.2	40	78
10	0.3	0	1	-	1265.2	1517.8	1526.5	1521.6	1.203	-	1800.1	24.9	1800.1	50	95
	0.5	0	1	-	2083.2	2635.2	2635.7	2253.9	1.082	-	1800.1	23.2	1800.1	50	93
	0.8	0	1	-	2612.3	2755.8	2757.3	2685.2	1.028	-	1800.3	25.7	1800.3	50	100
15	0.3	0	1	-	2916.4	3501.5	3523.7	3440.2	1.180	-	1800.3	56.7	1800.1	75	131
	0.5	0	1	-	4476.8	5728.7	5750.7	5030.1	1.124	-	1800.3	58.6	1800.2	75	137
	0.8	0	0	-	-	6423.5	6428.7	6206.3	-	-	1800.3	-	1800.3	75	150
20	0.3	0	1	-	5373.6	6483.6	6483.1	6284.2	1.169	-	1800.4	118.5	1800.3	100	178
	0.5	0	0	-	-	10639.6	11452.5	9019.4	-	-	1800.4	-	1800.4	100	200
	0.8	0	1	-	10149.7	10856.7	10858.2	10510.8	1.036	-	1800.4	117.4	1800.2	99	190
25	0.3	0	1	-	8023.9	9787.5	17369.6	9502.9	1.184	-	1800.4	166.9	1800.8	125	210
	0.5	0	1	-	12640.7	17424.1	17424.1	13936.0	1.102	-	1800.6	176.5	1800.5	125	236
	0.8	0	0	-	-	17512.8	17521.5	16806.1	-	-	1800.4	-	1800.5	125	175
30	0.3	0	1	-	11151.4	13751.2	24217.3	13520.2	1.212	-	1800.6	284.1	1801.1	150	267
	0.5	0	1	-	17342.4	25103.4	25103.4	16969.0	0.978	-	1800.8	288.5	1800.9	150	272
	0.8	0	0	-	-	25019.9	25028.6	24180.9	-	-	1800.6	-	1800.7	150	180

Note. Boldfaced values indicate better solutions found within the time limit, tighter dual bounds, or more PNEs.

Table 2.29: Comparison of BZR and ZR for Type A KPG.

n	BG	# PNEs Found		Best PNE (ϕ)		Dual Bounds		Price of Stability		Time (secs)				# EI Cuts	
		# _{pne} (ZR)	# _{pne} (BZR)	ZR	BZR	ZR	BZR	$\widetilde{\text{OSW}}$	$\widetilde{\text{POS}}$	T(ZR ^{1st})	T(ZR)	T(BZR ^{1st})	T(BZR)	# _{cuts} (ZR)	# _{cuts} (BZR)
2	0.2	1	3	18.5	18.5	18.5	18.5	18.7	1.011	1.1	1.7	0.1	2.5	2031	2040
	0.5	1	9	18.3	18.3	18.3	18.3	18.3	1.002	1.1	1.5	0.1	1.2	155	116
	0.8	1	10	8.3	8.3	8.3	8.3	8.4	1.007	2.7	3.1	0.1	2.3	671	657
3	0.2	5	9	37.6	37.6	37.6	37.6	37.8	1.006	12.0	14.8	1.1	9.6	1058	1009
	0.5	2	131	32.7	32.7	32.7	32.7	32.8	1.003	958.7	1646.6	0.2	174.2	3394	805
	0.8	0	61	-	14.8	14.9	14.8	15.2	1.025	-	1800.2	0.2	339.1	2739	1167
5	0.2	3	8	131.3	131.3	131.3	131.3	131.3	1.000	7.7	81.5	1.8	47.9	58	78
	0.5	0	93	-	60.6	61.0	61.0	61.4	1.014	-	1800.3	0.9	1800.1	2892	1730
	0.8	1	18	27.3	27.3	27.4	27.3	29.3	1.072	1784.9	1800.1	0.5	183.8	3982	3857
8	0.2	0	17	-	295.1	295.6	295.4	295.1	1.000	-	1800.7	5.0	1800.3	1657	200
	0.5	0	35	-	112.4	114.2	114.0	116.7	1.038	-	1800.4	1.5	1800.3	4833	4680
	0.8	0	49	-	86.9	87.7	87.6	87.8	1.010	-	1800.5	0.6	1800.9	571	649
10	0.2	0	8	-	505.5	506.5	506.1	508.7	1.006	-	1800.3	6.2	1800.6	2604	2157
	0.5	0	36	-	260.9	263.1	263.0	262.8	1.007	-	1800.9	2.6	1801.0	583	608
	0.8	0	54	-	141.2	143.0	142.9	143.9	1.019	-	1800.8	1.1	1800.6	1584	1653
15	0.2	0	13	-	1058.1	1061.3	1060.8	1058.8	1.001	-	1800.5	12.9	1801.5	344	254
	0.5	0	31	-	600.4	605.3	605.2	604.2	1.006	-	1800.8	11.3	1801.5	461	755
	0.8	0	35	-	255.3	258.3	258.3	257.7	1.009	-	1800.7	9.1	1801.6	325	822
20	0.2	0	13	-	1689.5	1694.4	1693.9	1689.9	1.000	-	1801.6	12.9	1800.6	279	372
	0.5	0	15	-	1120.4	1129.0	1129.0	1126.2	1.005	-	1800.7	3.8	1800.8	215	514
	0.8	0	14	-	610.8	615.5	615.2	612.8	1.003	-	1800.6	3.0	1800.8	348	395
25	0.2	0	8	-	2627.4	2637.5	2637.1	2628.1	1.000	-	1800.9	17.3	1801.1	309	355
	0.5	0	10	-	1827.5	1838.6	1838.4	1833.3	1.003	-	1800.7	5.6	1801.0	290	447
	0.8	0	12	-	706.6	714.5	714.5	711.5	1.007	-	1801.4	3.9	1800.9	332	547
30	0.2	0	7	-	3331.6	3356.9	3356.8	3351.2	1.006	-	1800.7	26.3	1802.9	197	397
	0.5	0	9	-	2528.7	2545.1	2544.7	2535.3	1.003	-	1800.8	8.4	1801.3	357	475
	0.8	0	11	-	1095.7	1109.9	1109.7	1104.6	1.008	-	1800.9	4.7	1801.0	312	586

Note. Boldfaced values indicate better solutions found within the time limit, tighter dual bounds, or more PNEs.

Table 2.30: Comparison of BZR and ZR for Type C KPG.

n	BG	# PNEs Found		Best PNE (ϕ)		Dual Bounds		Price of Stability		Time (secs)				# EI Cuts	
		# _{pne} (ZR)	# _{pne} (BZR)	ZR	BZR	ZR	BZR	$\widetilde{\text{OSW}}$	$\widetilde{\text{POS}}$	T(ZR ^{1st})	T(ZR)	T(BZR ^{1st})	T(BZR)	# _{cuts} (ZR)	# _{cuts} (BZR)
2	0.2	1	1	4.7	4.7	4.7	4.7	4.7	1.007	0.4	0.6	0.1	0.5	4075	4085
	0.5	3	14	7.8	7.8	7.8	7.8	7.8	1.007	2.3	3.3	0.1	3.7	4159	4196
	0.8	1	2	8.4	8.4	8.4	8.4	8.4	1.008	0.6	0.8	0.1	0.9	3567	3583
3	0.2	1	3	7.1	7.1	7.1	7.1	7.2	1.012	3.6	3.9	0.1	1.8	4385	4407
	0.5	2	5	10.3	10.3	10.3	10.3	10.4	1.011	39.1	51.2	0.3	12.3	4442	4451
	0.8	2	4	11.6	11.6	11.6	11.6	11.8	1.024	1.3	1.4	0.1	1.6	5629	5647
5	0.2	0	14	-	10.3	10.3	10.3	10.4	1.012	-	1800.4	0.3	243.4	4198	4274
	0.5	0	20	-	14.3	14.5	14.4	14.6	1.023	-	1800.6	0.5	1800.5	4986	5144
	0.8	2	33	13.7	13.7	13.7	13.7	15.0	1.092	235.0	446.0	0.8	177.6	4119	4227
8	0.2	0	19	-	15.8	16.1	16.0	16.2	1.024	-	1800.8	0.4	1801.0	2713	2856
	0.5	0	34	-	15.8	16.8	16.5	17.6	1.108	-	1800.9	1.0	1800.5	2267	2612
	0.8	0	146	-	15.7	16.8	16.6	18.5	1.183	-	1800.5	0.7	1800.5	3033	4072
10	0.2	0	19	-	16.7	17.4	17.3	17.5	1.051	-	1800.5	0.7	1800.4	1559	1749
	0.5	0	49	-	16.9	18.7	18.3	20.3	1.196	-	1800.7	6.0	1800.6	1564	2377
	0.8	0	156	-	16.3	18.6	18.0	20.3	1.244	-	1800.5	1.1	1800.8	1775	3841
15	0.2	0	18	-	19.8	21.7	21.5	21.8	1.104	-	1800.5	22.3	1800.4	1608	1831
	0.5	0	39	-	16.5	20.3	19.8	23.0	1.397	-	1800.3	21.7	1800.3	1651	2302
	0.8	0	37	-	16.8	20.2	19.8	23.0	1.366	-	1800.3	21.4	1800.4	1633	2362
20	0.2	0	11	-	20.9	24.1	24.0	24.5	1.170	-	1800.6	4.4	1800.4	120	340
	0.5	0	0	-	21.8	22.0	24.7	-	-	-	1800.2	-	1800.2	160	240
	0.8	0	13	-	17.1	22.7	23.1	25.5	1.492	-	1800.2	3.7	1800.3	200	352
25	0.2	0	7	-	20.7	25.9	26.4	26.2	1.267	-	1800.3	10.1	1800.4	150	360
	0.5	0	0	-	24.7	24.9	25.7	-	-	-	1800.3	-	1800.6	175	250
	0.8	0	0	-	-	25.5	25.4	26.4	-	-	1800.3	-	1800.5	175	250
30	0.2	0	2	-	20.5	29.3	40.8	27.7	1.355	-	1800.5	14.5	1800.8	150	526
	0.5	0	11	-	16.3	27.5	33.0	27.3	1.675	-	1800.5	8.7	1800.9	180	719
	0.8	0	18	-	15.8	27.9	30.4	27.6	1.747	-	1800.4	5.6	1801.5	210	891

Note. Boldfaced values indicate better solutions found within the time limit, tighter dual bounds, or more PNEs.

2.M Approximate-PNE

This appendix reports *approximate* equilibrium certificates for the instances in which our algorithms did not report an *exact* pure Nash equilibrium (PNE) within the prescribed computational limits. The goal is not to claim that a PNE exists in these instances, but rather to quantify the stability of the best-observed strategy profiles against unilateral deviations.

For each “no-PNE-reported” instance (both EBMC games and KPGs), we run RRR-BRD with a single restart and a fixed round cap: $L = 1$ and $R = 20$. Let $\mathbf{x}^{(r)}$ denote the strategy profile at the *end* of round $r \in \{1, \dots, 20\}$ (i.e., after all players have played once in that round under a randomized play order). We store these end-of-round profiles, yielding at most 20 candidate profiles per instance.

For any candidate profile $\mathbf{x}^{(r)}$, we compute a multiplicative approximation factor as follows. For each player $i \in N$, we solve the best-response problem against $\mathbf{x}^{-i}$ and obtain a best response $\hat{\mathbf{x}}^i \in \arg \max_{\mathbf{x}^i \in X_i} u_i(\mathbf{x}^i, \mathbf{x}^{-i})$. Assuming $u_i(\mathbf{x}^i, \mathbf{x}^{-i}) > 0$ for all feasible profiles, we define

$$\alpha_i(\mathbf{x}^{(r)}) := \frac{u_i(\hat{\mathbf{x}}^i, (\mathbf{x}^{(r)})^{-i})}{u_i(\mathbf{x}^{(r),i}, (\mathbf{x}^{(r)})^{-i})}, \quad \alpha(\mathbf{x}^{(r)}) := \max_{i \in N} \alpha_i(\mathbf{x}^{(r)}).$$

Then $\mathbf{x}^{(r)}$ is an $\alpha(\mathbf{x}^{(r)})$ -approximate PNE in the multiplicative sense, and $\alpha(\mathbf{x}^{(r)}) = 1$ if and only if $\mathbf{x}^{(r)}$ is an exact PNE. Among the 20 end-of-round candidates, we identify the profile with the smallest approximation factor,

$$\alpha^* := \min_{r=1, \dots, 20} \alpha(\mathbf{x}^{(r)}),$$

and report: (i) the best round r^* attaining α^* , (ii) the time taken for this RRR-BRD run, and (iii) the social welfare (global objective value) evaluated at $\mathbf{x}^{(r^*)}$.

Tables 2.31 and 2.32 summarize these results for EBMC games and KPGs, respectively.

Throughout, “no exact PNE was found within the time limit” records the outcome of a computation and is not a claim that the instance admits no PNE.

Table 2.31: Approximate PNE for the EBMC game instances in which no exact PNE was found within the time limit

Type	n	BG	Approx BRD (ϕ)	Best α	Time (secs)	Best Round
$ \mathcal{L} \geq 2$	15	0.8	6043.8	1.000003	106.3	11
$ \mathcal{L} \geq 2$	20	0.5	8292.9	1.000003	210.4	3
$ \mathcal{L} \geq 2$	25	0.8	16255.9	1.000002	393.8	19
$ \mathcal{L} \geq 2$	30	0.8	23509.4	1.000004	639.2	8

Table 2.32: Approximate PNE for the KPG instances in which no exact PNE was found within the time limit

Type	n	BG	Approx BRD (ϕ)	Best α	Time (secs)	Best Round
C	20	0.5	15.2	1.003140	0.9	9
C	25	0.5	14.5	1.003591	1.1	19
C	25	0.8	15.9	1.005714	1.2	20

2.N County-level Utility Values

Table 2.33 presents the complete set of county-level utility values under different strategy profiles, as introduced in Table 2.10 of the main manuscript.

Table 2.33: County-level utility values using different strategy profiles.

County	$u_i^{\text{Self}}(\bar{\mathbf{x}}_{\text{ng}})$	$u_i^{\text{Self}}(\hat{\mathbf{x}}_{\text{pne}}^1)$	$u_i^{\text{Self}}(\hat{\mathbf{x}}_{\text{pne}}^*)$	$u_i^{\text{Self}}(\hat{\mathbf{x}}_{\text{sw}}^{3600})$	County	$u_i^{\text{Self}}(\bar{\mathbf{x}}_{\text{ng}})$	$u_i^{\text{Self}}(\hat{\mathbf{x}}_{\text{pne}}^1)$	$u_i^{\text{Self}}(\hat{\mathbf{x}}_{\text{pne}}^*)$	$u_i^{\text{Self}}(\hat{\mathbf{x}}_{\text{sw}}^{3600})$
Aitkin	23746	24956	24982	29013	Martin	24	24	24	24
Anoka	11292	12466	12495	15723	McLeod	7194	7479	7503	8933
Becker	24219	26489	26497	31712	Meeker	8866	9423	9428	9964
Beltrami	18720	20049	20061	23756	Mille Lacs	1778	1818	1818	1886
Benton	1571	1688	1688	1779	Morrison	5243	5460	5466	6184
Big Stone	4555	4827	4827	5664	Mower	43	43	43	48
Blue Earth	4712	4924	4932	5439	Murray	3481	3576	3590	3906
Brown	2338	2436	2436	2805	Nicollet	32	32	32	32
Carlton	5393	5808	5813	6910	Nobles	2328	2396	2396	2520
Carver	6664	7306	7331	8211	Norman	8	8	8	8
Cass	40151	42450	42500	48926	Olmsted	587	596	598	852
Chippewa	60	60	60	60	Otter Tail	43479	47839	47863	58200
Chisago	5463	5923	5930	6827	Pennington	22	22	22	22
Clay	2075	2481	2485	2893	Pine	6061	6313	6321	7088
Clearwater	8484	9210	9235	10484	Pipestone	38	73	73	78
Cook	28157	29949	30031	34442	Polk	5584	6031	6037	6906
Cottonwood	2713	2824	2824	3212	Pope	10157	11020	11032	13728
Crow Wing	38044	39727	39789	46254	Ramsey	6995	7726	7761	9372
Dakota	6958	7400	7406	8664	Redwood	177	177	177	177
Dodge	7	7	7	7	Renville	837	866	871	1056
Douglas	16827	18127	18136	22158	Rice	4496	4839	4847	5076
Faribault	251	251	251	251	Rock	0	0	0	0
Freeborn	1172	1172	1172	1172	Roseau	234	234	234	234
Goodhue	770	845	845	879	Saint Louis	78691	81719	81770	91172
Grant	6635	7389	7391	8817	Scott	4983	5249	5253	6381
Hennepin	21136	22442	22474	27024	Sherburne	5077	5557	5564	6698
Hubbard	22780	24426	24444	29543	Sibley	2390	2388	2388	2429
Isanti	3635	3876	3884	4227	Stearns	20200	21544	21554	26076
Itasca	51214	53588	53640	61367	Steele	372	372	372	372
Jackson	3061	3167	3171	3474	Stevens	2938	3340	3382	4142
Kanabec	3697	3917	3927	4634	Swift	1287	1461	1461	1778
Kandiyohi	15551	16438	16450	18723	Todd	9108	9832	9842	11661
Kittson	254	254	254	254	Traverse	1427	1466	1466	1523
Koochiching	2217	2278	2284	2314	Wabasha	331	331	331	331
Lac qui Parle	1331	1376	1376	1405	Wadena	1834	1950	1950	2289
Lake	30194	31584	31640	36702	Waseca	2940	3074	3080	3089
Lake of the Woods	1009	1009	1009	1009	Washington	11252	11946	11968	15845
Le Sueur	6804	7135	7157	7929	Watsonwan	946	955	955	1114
Lincoln	4221	4292	4292	4501	Wilkin	94	102	102	128
Lyon	2752	2925	2939	3314	Winona	1561	1579	1579	1444
Mahnomen	4177	4424	4428	6012	Wright	22047	23782	23805	29464
Marshall	4585	5076	5078	6081	Yellow Medicine	829	858	858	938

Note. Counties for which the inequalities regarding utility value comparisons do not hold are highlighted in bold.

Chapter 3

Nash-informed Planning for Public-Resource Coordination: An Application to Aquatic Invasive Species Prevention

Manuscript Note. This chapter is based in part on the following manuscript:

H. Lee, R. Hildebrand, W. Cai, A. Kinsley, N. Phelps, and İ. E. Büyüктаhtakın, “Nash-informed Planning for Public-Resource Coordination: An Application to Aquatic Invasive Species Prevention.” Working paper.

The manuscript has been adapted and expanded for inclusion in this dissertation; notation, exposition, and organization have been revised for consistency with the remaining chapters. Code and data are available at <https://github.com/HyunwooLee0429/nash-informed-planning>.

3.1 Introduction

A common challenge in public-sector operations is to improve system-wide outcomes when authority is decentralized. A state environmental agency, for example, may wish to raise

the effectiveness of inspection programs across dozens of counties whose budgets, staffing decisions, and lake selections are autonomous. The agency can anticipate the collective outcome that emerges when each county acts in its own interest, but it cannot mandate that counties choose differently, and it cannot redesign the institutional structure (county budgets, jurisdictional boundaries, statutory authority) that shapes the game in the first place. The agency’s instrument is, instead, the recommendation: an allocation that counties may voluntarily adopt in lieu of the equilibrium they would otherwise reach. This chapter studies when such voluntary recommendations can substantially raise welfare, and the structural conditions under which they do so without requiring any county to be worse off. For convenience, all acronyms used throughout this chapter are collected in Section [3.A](#).

3.1.1 A new direction of analysis

Prior work divides by what it takes as given. Algorithmic game theory (AGT) and the integer programming game (IPG) literature take a game and produce an equilibrium; mechanism design takes a target outcome and produces a game whose equilibrium implements it. We extend the first: taking the game and its equilibria as given, we add a downstream step that produces a welfare-improving recommendation under equilibrium-benchmarked utility guarantees. The interaction structure and utility functions are unchanged. The only principle borrowed from mechanism design is individual rationality (IR), which functions in our framework as a structural device: the IR constraint releases the equilibrium fixity of each player’s action and re-admits that action as a decision variable, but only within a region that preserves the player’s equilibrium utility. The act of releasing the equilibrium action variable, conditional on the IR guarantee, is what makes voluntary post-equilibrium re-optimization coherent as a procedure. We refer to this approach as *Nash-informed planning*. Figure [3.1](#) illustrates the three steps.

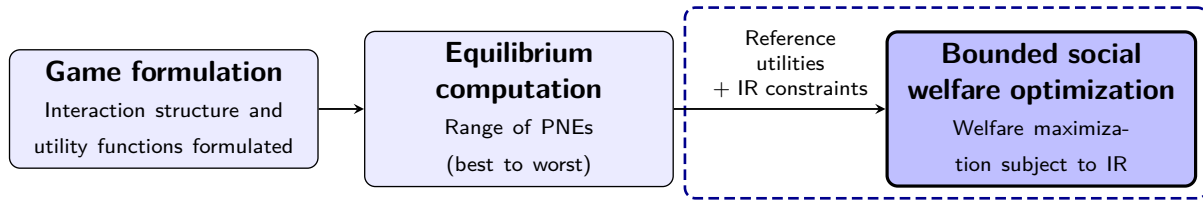


Figure 3.1: The three steps of Nash-informed planning. The dashed box encloses the contribution of this chapter: the reference-utility construction (the second arrow) and the bounded social welfare optimization (the third box). The first two boxes assemble the game and compute its pure Nash equilibria (PNEs) using existing methods.

The framework targets complete-information non-cooperative games in which each player’s strategy is a discrete decision (resource allocation, scheduling, and the like) that we model compactly as IPGs. The sum of player utilities defines social welfare; its maximum is the *social optimum*. The equilibrium-computation step uses dedicated algorithms to find a representative range of Pure Nash Equilibria (PNEs), from the welfare-best to the welfare-worst; algorithmic game theory then quantifies how far these PNEs lie from the social optimum [127]. Even the best PNE may sacrifice nontrivial welfare. Our Nash-informed planning step closes this gap by constructing equilibrium-based individual participation rationality constraints from the computed equilibria (utility lower bounds that preserve each player’s equilibrium-derived benchmark) and solving a bounded social welfare optimization that improves welfare while removing utility-based participation objections relative to the modeled equilibrium fallback.

We motivate and test the framework on a case study of aquatic invasive species (AIS) prevention in Minnesota. The state has 87 counties and 9,182 lakes, of which 506 were infested with at least one AIS type as of 2018; 41 counties have documented inspection programs, and 38 are strategically active at the 100% baseline under the model’s positive-integer-budget threshold. Boat movements link infested and uninfested lakes across county lines, so each county’s inspection choices generate spillovers to other counties whose lakes

receive incoming traffic. The state planner seeks to maximize inspection coverage of risky boat movements across the state, while each county aims to maximize coverage at its own lakes, with objectives aligned in direction but differing in jurisdictional scope.

The framework’s institutional assumptions (fixed county budgets, information-sharing through state channels, and a coordinator without coercive authority) were informed by sustained engagement with county and state stakeholders, such as the Minnesota Department of Natural Resources (DNR), rather than imposed for modeling convenience.

3.1.2 Related literature

We review four streams of work related to Nash-informed planning.

Stream 1: IPGs and equilibrium computation. IPGs are non-cooperative games in which each player solves an integer program under complete information [37, 96]. The framework captures decision settings with inherently combinatorial strategy spaces: knapsack [36, 58], kidney exchange [34, 36], and the edge-weighted budgeted maximum coverage games studied here [98]. Algorithms for IPGs focus primarily on equilibrium computation: the zero-regret (ZR) algorithm of Dragotto and Scatamacchia [58] uses a cutting-plane framework to compute the socially best PNE, and Lee et al. [98] study the use of random-restart best-response dynamics (BRD) in IPGs, both theoretically and computationally, and show how to incorporate BRD into ZR to scale to large instances, yielding the BZR algorithm.

Stream 2: Inefficiency of equilibria in operations management (OM). Quantifying the inefficiency of decentralized equilibria is a long-standing question in both algorithmic game theory (AGT) and operations management (OM). The Price of Stability (POS) and Price of Anarchy (POA) [52, 127] measure the welfare ratio of the socially best and worst

equilibria to the social optimum, and Filos-Ratsikas et al. [69] characterize the Pareto frontier between these two metrics. Perakis [118] characterizes POA bounds under nonlinear and asymmetric costs in transportation and supply-chain settings; Perakis and Roels [119] quantify POA in supply chains under price-only contracts; Cachon [30] and Bernstein and Federgruen [15] analyze POA and POS in inventory and capacity games, and the cost of decentralization relative to the centralized optimum is quantified in scheduling [20] and warehouse-logistics [55] settings. Christodoulou and Koutsoupias [46] establish theoretical bounds for network congestion games. Where POA and POS quantify decentralization’s inefficiency, this chapter asks how much of it a planner can remove while keeping participation voluntary—the question the rest of this chapter takes up.

Stream 3: Coordination through game redesign and equilibrium selection. One major approach to coordinating decentralized agents is to alter the game itself. *Mechanism design* [89, 97] constructs a game whose equilibrium implements a target outcome, with computational tractability [110] or money-free [121] variants developed in subsequent work; recent OM applications include kidney exchange [9]. *Supply-chain coordination contracts* alter payoffs through buyback, revenue-sharing, and related transfer-based instruments [15, 30, 31, 120]. Our framework keeps the game fixed and intervenes only through the recommendation. In AIS control and management, counties can rarely justify resource transfers to their stakeholders without a guarantee of improved local outcomes, so we take the institutional structure (payoffs, action spaces, and allocation rules) as given and introduce no transfers. The one element we adopt from mechanism design is individual rationality, which enters as a structural device rather than as part of a designed game form.

A natural alternative is the *correlated equilibrium* (CE) of Aumann [12, 13], a distribution over joint strategies in which following a mediator’s private recommendation is incentive-

compatible. Correlated equilibrium fits a different institutional setting than ours: a state-level environmental agency issues a single public plan and cannot administer the private, randomized recommendations a CE requires. The deterministic-CE and computational-tractability comparisons [8, 116] are deferred to Section 3.C, which contrasts BSW with the welfare-best CE directly. We explore two analytical connections—to the *price of fairness* of Bertsimas et al. [16, 17] and to Nash [1950] bargaining—in Sections 3.2.3 and 3.2.5.

Stream 4: Public-sector OM, with focus on AIS prevention. Aquatic invasive species (AIS) prevention is an operationally significant environmental problem; Minnesota’s AIS Prevention Aid program appropriates \$10 million annually to counties for aids payable through 2026 under Minnesota Statutes §477A.19. The optimization-based AIS literature models prevention as a single-planner problem: locating inspection stations [81], bi-level state–county allocation [82], and cross-jurisdictional collaboration structures [94]. Lee et al. [98] formalized it as an EBMC game and identified welfare-improving coordination beyond equilibrium as future work; the present paper picks up there. This work also connects to environmental and closed-loop operations, which study environmental externalities in problems such as remanufacturing and reverse-channel design [10, 11, 138], surveyed by Drake and Spinler [59]. More broadly, recent public-sector and humanitarian operations work coordinates scarce resources across decentralized providers: NGO-operated rural water systems under limited repair budgets [146] and government-to-government food procurement via Nash bargaining [106], settings that, like AIS prevention, pair public objectives with autonomous local execution.

3.1.3 Contributions and plan

Our contributions are summarized as follows.

First, we introduce *Nash-informed planning*, a framework that takes the equilibrium of a decentralized integer programming game as input to a downstream welfare-improving planning step under individual rationality (Section 3.1.1): taking the game as given by institutional structure, it computes equilibria and issues a welfare-improving recommendation that removes utility-based participation objections relative to the modeled equilibrium fallback, improving social welfare while preserving each planner’s equilibrium utility.

Second, we formalize how equilibrium information enters the planning step. From a representative range of PNEs obtained by running BZR in both welfare-maximizing and welfare-minimizing directions, we construct utility lower bounds that serve as IR constraints in a bounded social welfare optimization. We define the price of individual rationality (PIR) and show that this welfare cost is bounded by the price of stability in any IPG under the socially best-PNE reference. Three optimization models span increasing coordination levels from decentralized budgets to full pooling, with policy-constrained variants (opt-out, minimum-budget retention, association-based pooling) capturing institutional constraints.

Third, we evaluate the framework in a real-world-driven Minnesota AIS planning setting that combines actual lake and county structure, observed infestation records, county-level budget rules, and survey-calibrated estimates of lake-to-lake boat movements. Across four budget scenarios, the unconstrained social optimum already leaves specific counties below their equilibrium-derived participation benchmark; the Nash-informed plan removes these concentrated county-level deficits while sacrificing at most 0.15% of statewide welfare (a realized PIR within 0.15% of 1). The operational contribution is thus not that individual rationality is cheap in the aggregate, but that it converts a nearly optimal plan some counties would reject into one that preserves every county’s equilibrium utility; a structural explanation, rooted in the hub-and-spillover structure of the boat-movement network, indicates when this holds.

Fourth, we complement the Minnesota study with synthetic and benchmark instances from EBMC, KPG [58], and KEG [36] to examine whether the computational regularity extends beyond the motivating application. Across 28 non-trivial instances the strict ordering $1 \leq \text{PIR} < \text{POS}$ holds empirically, with realized PIR not exceeding 1.050.

Plan of the chapter. Section 3.2 introduces the Nash-informed planning framework. Section 3.3 specializes it to EBMC games. Section 3.4 explains the PIR-near-one regularity. Section 3.5 reports the Minnesota computational study, Section 3.6 demonstrates cross-class generalization (EBMC random, KPG, KEG), and Section 3.7 distills the managerial implications. Section 3.8 concludes.

3.2 Nash-informed Planning Framework

This section develops the Nash-informed planning framework for general IPGs, in two phases: an equilibrium-analysis phase (Sections 3.2.1 and 3.2.2) and a Nash-informed planning phase that constructs reference utilities and solves bounded social-welfare optimization (Sections 3.2.3 and 3.2.4).

The model, up front. The contribution is a single optimization. Given an IPG with social welfare $\phi(\mathbf{x}) = \sum_{i \in N} u_i(\mathbf{x})$, the Nash-informed planner solves

$$\max_{\mathbf{x}} \phi(\mathbf{x}) \quad \text{s.t.} \quad \mathbf{x} \in \tilde{\mathcal{X}}, \quad u_i(\mathbf{x}) \geq u_i^\dagger \quad \forall i \in N, \quad (3.1)$$

The constraints $u_i(\mathbf{x}) \geq u_i^\dagger$ are *equilibrium-based individual participation rationality constraints* (IR constraints), where $u_i^\dagger$ is player i 's *reference utility*, an equilibrium-derived utility benchmark constructed from the computed PNEs (Section 3.2.3), and $\tilde{\mathcal{X}}$ encodes the

coordination regime (Section 3.2.4). The objective is the planner's, the constraints are the players'. Because $u_i^\dagger$ is an equilibrium-derived benchmark rather than an exogenous fairness parameter, every feasible point preserves each player's equilibrium-derived utility floor, so the optimum is welfare-improving and participation-compatible relative to the reference utilities. Its welfare cost, the price of individual rationality (PIR), is bounded by the price of stability (Proposition 3.3) and stays within 0.15% of the optimum in our case study (Section 3.5). The rest of this section develops the two phases behind (3.1): an equilibrium-analysis phase using existing algorithms (ZR, BZR), and the Nash-informed phase (new here) that builds $u_i^\dagger$ and solves the model.

3.2.1 General IPG setting and notation

An IPG is a non-cooperative simultaneous game under complete information, where each player's feasible strategies are defined by integer programming constraints. Strategic interactions are modeled through multilinear terms in players' objective functions, most commonly bilinear ones. While the original definition of Köppe et al. [96] allows for mixed-integer strategy sets, we focus on pure-integer sets, which are sufficient for modeling EBMC games. Formally, an IPG is the tuple $G = (N, (\mathcal{X}_i)_{i \in N}, (u_i)_{i \in N})$, where $N := \{1, \dots, n\}$ is the set of players, $\mathcal{X}_i = \{\mathbf{x}^i \in \mathbb{Z}^{m_i} : A^i \mathbf{x}^i \leq \mathbf{b}^i\}$ is the pure-integer strategy set for player i (with rational A^i , $\mathbf{b}^i$, dimension m_i), and $u_i(\mathbf{x}^i, \mathbf{x}^{-i}) : \mathcal{X}_1 \times \dots \times \mathcal{X}_n \rightarrow \mathbb{R}$ is player i 's utility, parametrized by the strategies $\mathbf{x}^{-i}$ of the other players. The joint strategy space is $\mathcal{X} := \mathcal{X}_1 \times \dots \times \mathcal{X}_n$.

The utility function takes the form

$$u_i(\mathbf{x}^i, \mathbf{x}^{-i}) = (\mathbf{x}^i)^\top Q^i \mathbf{x}^i + (\mathbf{d}^i)^\top \mathbf{x}^i + \sum_{p \in N: p \neq i} (\mathbf{x}^p)^\top Q_p^i \mathbf{x}^i, \quad (3.2)$$

where Q^i is upper triangular, $\mathbf{d}^i$ is a linear coefficient vector, and the cross-terms $(\mathbf{x}^p)^\top Q_p^i \mathbf{x}^i$ capture pairwise inter-player interactions through reciprocally bilinear terms. Higher-order multilinear extensions accommodate trilinear and beyond, as in the length-three KEG of Section 3.H.

In an IPG, each player solves a best-response problem $\max_{\mathbf{x}^i \in \mathcal{X}_i} u_i(\mathbf{x}^i, \mathbf{x}^{-i})$. A strategy profile $\hat{\mathbf{x}} = (\hat{\mathbf{x}}^1, \dots, \hat{\mathbf{x}}^n)$ is a *pure Nash equilibrium* (PNE) if $u_i(\hat{\mathbf{x}}^i, \hat{\mathbf{x}}^{-i}) \geq u_i(\mathbf{x}^i, \hat{\mathbf{x}}^{-i})$ for every $i \in N$ and every $\mathbf{x}^i \in \mathcal{X}_i$. This chapter focuses on PNEs, since mixed Nash equilibria are difficult to implement in real public-sector settings such as AIS prevention. We define the social welfare function as $\phi(\mathbf{x}) = \sum_{i \in N} u_i(\mathbf{x}^i, \mathbf{x}^{-i})$, and the socially optimal solution as $\bar{\mathbf{x}}_{\text{sw}}^* \in \arg \max_{\mathbf{x} \in \mathcal{X}} \phi(\mathbf{x})$. Let $\mathcal{S}_{\text{pne}}$ denote the set of PNEs. The *best PNE* $\hat{\mathbf{x}}_{\text{pne}}^* \in \arg \max\{\phi(\mathbf{x}) : \mathbf{x} \in \mathcal{S}_{\text{pne}}\}$ and the *worst PNE* $\hat{\mathbf{x}}_{\text{pne}}^0 \in \arg \min\{\phi(\mathbf{x}) : \mathbf{x} \in \mathcal{S}_{\text{pne}}\}$ define the welfare range. The Price of Stability $\text{POS} := \phi(\bar{\mathbf{x}}_{\text{sw}}^*)/\phi(\hat{\mathbf{x}}_{\text{pne}}^*)$ and the Price of Anarchy $\text{POA} := \phi(\bar{\mathbf{x}}_{\text{sw}}^*)/\phi(\hat{\mathbf{x}}_{\text{pne}}^0)$ measure the inefficiency of the best and worst equilibria, respectively. Larger values indicate that equilibria are farther from the social optimum.

3.2.2 Equilibrium analysis phase

This phase assembles the game and computes its equilibria using existing components (Figure 3.1, first two boxes). Welfare is benchmarked against the social optimum, the optimal value of

$$\max_{\mathbf{x} \in \mathcal{X}} \phi(\mathbf{x}) = \max_{\mathbf{x} \in \mathcal{X}} \sum_{i \in N} u_i(\mathbf{x}). \quad (3.3)$$

The contrast between this social objective and each player's best-response problem $\max_{\mathbf{x}^i \in \mathcal{X}_i} u_i(\mathbf{x}^i, \bar{\mathbf{x}}^{-i})$ captures the tension between system-level efficiency and individual incentives. Strategic interaction is encoded through the MIP-linearizable multilinear terms or through auxiliary variables in a lifted formulation.

Computing pure Nash equilibria. The ZR algorithm [58] computes the best PNE through a cutting-plane framework, and Lee et al. [98] study best-response dynamics (BRD) for IPGs and show how to incorporate a random-restart BRD primal heuristic into ZR, yielding the BZR algorithm. BZR scales to instances with $n \geq 10$ players and produces multiple PNEs along the cutting-plane trajectory; we adopt BZR as our default. To compute the worst PNE, we run BZR with the welfare-minimization objective. Running BZR in both directions yields a set of equilibria covering the welfare range, which is the input to the Nash-informed phase.

3.2.3 Construction of reference utility values

The Nash-informed phase begins by translating computed equilibria into *reference utility values* that serve as individual rationality (IR) benchmarks. We construct IR constraints of the form

$$u_i(\mathbf{x}) \geq u_i^\dagger, \quad \forall i \in N, \quad (3.4)$$

where $u_i^\dagger$ is a fixed *reference utility*, a scalar floor for player i . We outline three principled rules for defining $u_i^\dagger$ and the resulting bounds. Only rule (i) below evaluates u_i at a single common strategy profile; rules (ii) and (iii) assemble $u_i^\dagger$ coordinate-by-coordinate and generally correspond to no single profile, which is why we treat it as a scalar rather than as u_i evaluated at a reference profile.

(i) A specific equilibrium as reference. A single PNE can serve as the reference; examples include the unique PNE (when verifiable) and the socially best or worst PNE. We focus on the socially best PNE $\hat{\mathbf{x}}_{\text{pne}}^*$, setting $u_i^\dagger := u_i(\hat{\mathbf{x}}_{\text{pne}}^*)$: if it is also individually best for every player, it is the payoff-dominant equilibrium of Harsanyi [86], and selecting the best

PNE as reference parallels the POS benchmarking perspective [52, 127].

(ii) Average across the collected equilibrium set. When BZR collects multiple PNEs $\mathcal{S}_{\text{pne}}$, the reference can be the per-player mean utility:

$$u_i^\dagger = \frac{1}{|\mathcal{S}_{\text{pne}}|} \sum_{\mathbf{x} \in \mathcal{S}_{\text{pne}}} u_i(\mathbf{x}), \quad \forall i \in N. \quad (3.5)$$

This expectation-based benchmark reflects uncertainty in equilibrium selection, akin to probabilistic equilibrium frameworks [86]. A potential issue is that the averaged utility profile may not correspond to any feasible PNE, so the bounded social welfare optimization may be infeasible.

(iii) Conservative minimum across the collected equilibrium set. A robust alternative takes the per-player minimum across the same collected set:

$$u_i^\dagger = \min_{\mathbf{x} \in \mathcal{S}_{\text{pne}}} u_i(\mathbf{x}), \quad \forall i \in N, \quad (3.6)$$

which guarantees feasibility (the worst PNE satisfies these bounds). This coordinatewise minimum is a conservative equilibrium-derived floor; because its components may come from different PNEs, it need not be the payoff of any single equilibrium, so we read it as a threat-point-style floor rather than a literal Nash [1950] bargaining solution.

Application order. We apply the rules in cascade: (i) the socially best PNE when it is individually dominant (the payoff-dominant case of Harsanyi [86]); otherwise (ii) as the default; and (iii), always feasible, if BSW is infeasible under (ii). Since BZR collects both welfare-extreme PNEs, (ii) and (iii) reduce to the average and minimum of the two extremes

when $|\mathcal{S}_{\text{pne}}| = 2$.

Two practical caveats apply. Since the IPG instances we consider are large-scale, the BZR algorithm typically runs under a time limit and may not certify zero MIP gap. The collected PNE set $\mathcal{S}_{\text{pne}}$ is a sample, not exhaustive. A small KPG-style instance illustrating how this rule choice drives the comparison between BSW and the welfare-best correlated equilibrium is given in Section 3.C.

3.2.4 Bounded social welfare (BSW) optimization

Given reference utilities, the Nash-informed planner solves the bounded social welfare (BSW) optimization

$$\max_{\mathbf{x}} \{ \phi(\mathbf{x}) \mid \mathbf{x} \in \tilde{\mathcal{X}}, \mathbf{x} \in \mathcal{R} \}, \quad (3.7)$$

where $\mathcal{R} := \{ \mathbf{x} : u_i(\mathbf{x}) \geq u_i^\dagger, \forall i \in N \}$ is the *individually rational region* and $\tilde{\mathcal{X}}$ is a modified feasibility set that captures the level of coordination available.

Three configurations of $\tilde{\mathcal{X}}$ correspond to different coordination regimes. Let $\mathbf{x} = (\mathbf{x}^1, \dots, \mathbf{x}^n)$ denote the joint strategy and $m := \sum_{i \in N} m_i$. In the *standard non-cooperative* setting, $\tilde{\mathcal{X}}$ is the original joint strategy set:

$$\tilde{\mathcal{X}} := \{ \mathbf{x} \in \mathbb{Z}^m : A^i \mathbf{x}^i \leq b^i, \forall i \in N \}. \quad (3.8)$$

Under *full cooperation* (the grand coalition), per-player resource constraints are pooled, assuming the A^i and b^i are conformable in a shared resource space (in the EBMC instantiation of Section 3.3, these reduce to scalar per-county budgets):

$$\tilde{\mathcal{X}} := \left\{ \mathbf{x} \in \mathbb{Z}^m : \sum_{i \in N} A^i \mathbf{x}^i \leq \sum_{i \in N} b^i \right\}. \quad (3.9)$$

A more general *coalitional* setting allows partial budget pooling within coalitions. Let G index possible coalitions; let $z_{i,g} \in \{0, 1\}$ indicate whether player i is assigned to coalition g ; and let $\mathbf{x}'_g$ be the player- i , coalition- g decision, with $\bar{\mathbf{x}}^i$ a componentwise upper bound on $\mathcal{X}_i$. The feasibility set $\tilde{\mathcal{X}}$ is then

$$\mathbf{x}' \in \mathbb{Z}^{m \cdot |G|}, \quad z \in \{0, 1\}^{|N| \cdot |G|}, \quad (3.10a)$$

$$\sum_{i \in N} A^i \mathbf{x}'_g \leq \sum_{i \in N} b^i \cdot z_{i,g}, \quad \forall g \in G, \quad (3.10b)$$

$$\mathbf{x}'_g \leq \bar{\mathbf{x}}^i z_{i,g}, \quad \forall i \in N, g \in G, \quad (3.10c)$$

$$\sum_{g \in G} z_{i,g} = 1, \quad \forall i \in N. \quad (3.10d)$$

By construction, (3.10) relaxes (3.8), and (3.9) relaxes (3.10), so enlarging the feasible region increases attainable social welfare; Section 3.2.5 makes these orderings precise. Section 3.3 specializes the formulations to EBMC games.

The region $\mathcal{R}$ is the single device that carries the Nash-informed content: across the three coordination regimes only $\tilde{\mathcal{X}}$ changes, while $\mathcal{R}$ is held fixed. This makes the contrast with the classical approach explicit. Algorithmic game theory restricts attention to the equilibrium set and reports the best PNE, $\max\{\phi(\mathbf{x}) : \mathbf{x} \in \mathcal{S}_{\text{pne}}\}$; Nash-informed planning instead optimizes over $\tilde{\mathcal{X}} \cap \mathcal{R}$, relaxing the equilibrium-*stability* requirement to a *participation* requirement. Under rule (i), the selected socially best PNE lies in $\mathcal{R}$ because it attains its own reference utilities; under rule (iii), every PNE in the collected set satisfies the coordinatewise minimum floors, so $\mathcal{S}_{\text{pne}} \subseteq \mathcal{R}$. Under rule (ii), the coordinatewise average utility vector need not correspond to a feasible profile, and BSW feasibility must therefore be checked. When BSW is feasible, it optimizes over a tractable region that (under rules (i) and (iii)) contains the equilibria and can strictly improve on the best PNE while remaining participation-compatible.

3.2.5 Structural guarantees

Setting $\tilde{\mathcal{X}}$ to the decentralized (3.8), coalitional (3.10), and pooled (3.9) configurations in the bounded model (3.7) yields the *bounded social welfare* (BSW), *bounded coalition* (BC), and *bounded grand coalition* (BGC) models, with optimal values $z^{\text{BSW}}, z^{\text{BC}}, z^{\text{BGC}}$; dropping the individually rational region $\mathcal{R}$ gives their IR-free counterparts, the *social welfare* (SW) and *grand coalition* (GC) models, with values $z^{\text{SW}}, z^{\text{GC}}$. Let $\bar{\mathbf{x}}_{\text{sw}}^*$ and $\bar{\mathbf{x}}_{\text{bsw}}$ denote optimal solutions of the SW and BSW models; $\bar{\mathbf{x}}_{\text{sw}}^*$ is the social optimum. Write $\phi^\dagger := \sum_{i \in N} u_i^\dagger$ for the sum of per-player floors, a scalar reference-welfare level, not welfare evaluated at any profile (under rules (ii)/(iii) no feasible profile realizes it). Throughout this subsection we assume BSW is feasible and that the welfare values entering PIR, POS, and POA are positive. Proofs of the results in this subsection are given in Section 3.B.

Proposition 3.1. *The decentralized and pooled models satisfy: (i) feasible sets $P_{\text{BSW}} \subseteq P_{\text{SW}} \subseteq P_{\text{GC}}$; and (ii) objective values $\phi^\dagger \leq z^{\text{BSW}} \leq z^{\text{SW}} \leq z^{\text{GC}}$.*

Definition 3.2. The *Price of Individual Rationality* is

$$\text{PIR} := \frac{\phi(\bar{\mathbf{x}}_{\text{sw}}^*)}{\phi(\bar{\mathbf{x}}_{\text{bsw}})} = \frac{z^{\text{SW}}}{z^{\text{BSW}}},$$

the welfare the planner sacrifices by guaranteeing each player at least its reference utility.

PIR has the same mathematical form as the *price of fairness* of Bertsimas et al. [16, 17]: $1 - 1/\text{PIR}$ equals the relative welfare loss, the fractional drop in social welfare imposed by restricting the feasible region to a fairness scheme U , when the IR-feasible region is treated as the fairness scheme. Our setting differs in two respects: the side constraints are equilibrium-derived utility floors rather than exogenous fairness criteria, so the constraint set itself is an output of the equilibrium-computation phase; and the relevant comparison bound is POS

rather than a worst-case constant over a problem class. In bargaining language, BSW is the *utilitarian* selection $\max \sum_i u_i$ from the IR-feasible bargaining set (S, d) introduced in Section 3.2.3.

Proposition 3.3. *PIR ≥ 1 . Moreover, $\text{PIR} \leq \text{POS}$ holds when the reference utilities are constructed either from the socially best PNE $\hat{\mathbf{x}}_{\text{pne}}^*$ (rule (i) of Section 3.2.3) or as the per-player minimum across the collected equilibrium set $\mathcal{S}_{\text{pne}}$ (rule (iii)).*

The ordering $\text{PIR} \leq \text{POS}$ holds because the socially best PNE remains feasible for BSW under rules (i) and (iii). Under the average-PNE rule, a player’s average equilibrium utility may exceed its utility at the socially best PNE, so that PNE need not be BSW-feasible; any $\text{PIR} \leq \text{POS}$ ordering observed under rule (ii) is therefore empirical rather than guaranteed by Proposition 3.3.

Proposition 3.4. *The bounded coalitional models satisfy: (i) feasible sets (after projection) $P_{\text{BSW}} \subseteq \text{proj}_{\mathbf{x}} P_{\text{BC}} \subseteq P_{\text{BGC}} \subseteq P_{\text{GC}}$; and (ii) objective values $\phi^\dagger \leq z^{\text{BSW}} \leq z^{\text{BC}} \leq z^{\text{BGC}} \leq z^{\text{GC}}$.*

Remark 3.5 (Scope). The framework and the price-of-individual-rationality bound (Proposition 3.3) use only a welfare objective and a feasible equilibrium reference; integrality of the strategy sets is never invoked in the proof, so both extend to any complete-information game with these ingredients. We instantiate on integer programming games because the equilibrium–optimum gap the bound concerns is structurally richer in combinatorial settings. Recent algorithms (ZR, BZR) make equilibria of these games computable at the scale the applications demand [58, 98].

3.3 EBMC Games and Application Models

This section specializes the Nash-informed planning framework to EBMC games, the structure governing public-sector inspection problems such as AIS prevention. The Minnesota operational context (Section 3.3.1) motivates the modeling assumptions used throughout.

3.3.1 Operational context and modeling assumptions

The EBMC specialization of Nash-informed planning is motivated by, and tailored to, the institutional structure of Minnesota’s AIS prevention program. We summarize that structure here and make explicit the modeling assumptions it implies; the EBMC formulation in the subsequent subsections is constructed to match it.

Institutional setting. Minnesota has 87 counties; 41 have a documented county inspection program, of which 38 have a positive integer inspection budget at the 100% baseline under our budget conversion. The AIS Prevention Aid program is administered by the State of Minnesota, which sets the funding formula based on the number of boat ramps and parking spots at public water access sites within each county (Minnesota Statutes §477A.19); counties independently decide how to use their allocations within statutory requirements. Analytical resources, most notably the AIS Explorer decision-support platform developed by the Minnesota Aquatic Invasive Species Research Center (MAISRC), are available to facilitate intra- and inter-county prioritization planning [2, 93]. County planners may share intended inspection sites at one-on-one and group coordination meetings organized by the Minnesota Department of Natural Resources (DNR); state-level researchers and managers provide analytical tools and play a convening role, respectively, while counties act independently with open information sharing among participants. The operational reality is therefore one of

coordination without authority: counties retain decision authority, while state and research partners provide support.

Modeling assumptions. This setting motivates three modeling assumptions that we adopt throughout the EBMC application.

(A1) Complete information. Each strategic county is assumed to know the intended inspection plans of the other strategic counties at the planning stage. This approximates the information-sharing environment created by these coordination meetings and allows the decentralized phase to be modeled as a complete-information IPG.

(A2) Decentralized integer programming game structure. Each county solves an integer program whose strategy set is constrained by its own budget. Inspection at each lake is modeled as a binary decision (inspect or not); although fractional allocations of inspector-hours exist in practice, we adopt the binary convention of prior AIS inspection optimization work [81, 94, 98]. County utilities interact through the cross-jurisdictional structure of the boat-movement network.

(A3) Analytical authority without mandatory authority. The state planner can compute equilibria, evaluate alternative allocations, and *recommend* coordinated plans, but cannot mandate county compliance. Recommendations are voluntary, which is what makes individual rationality a binding feature of any implementable plan.

3.3.2 EBMC formulation

EBMC games [98] model decentralized inspection problems on a directed network. The notation is summarized in Table 3.1.

Table 3.1: Notation for EBMC games

Sets			
N	Counties, $N = \{1, \dots, n\}$	$\mathcal{A}$ (resp. $\mathcal{A}_i$)	Arcs (boat movements); $\mathcal{A}_i$ internal to $\mathcal{J}_i$
$\mathcal{J}$ (resp. $\mathcal{J}_i$)	Lakes in the state (resp. in county i)	$\delta_-(\mathcal{J}_i)$	Arcs entering $\mathcal{J}_i$
L	Aquatic invasive species		
Parameters			
n_{jk}	Number of boats moving from j to k	w_{jk}	Risky boats, $w_{jk} = b_{jk} \cdot n_{jk}$
f_{jl}	Binary indicator that lake j is infested w/ l	$\mathcal{B}_i$	Inspection budget for county i
b_{jk}	Per-boat infestation risk, $b_{jk} = f_{jl}(1 - f_{kl})$	$\mathcal{B}$	Total budget, $\mathcal{B} = \sum_{i \in N} \mathcal{B}_i$
Variables			
x_j	Binary; 1 if lake j is selected for inspection	y_{jk}	Binary; 1 if boat movement (j, k) is inspected
Coalition notation $(G, x'_{j,g}, z_{i,g})$ is introduced with the BC model in Section 3.3.3.			

State-level optimization (SW and GC). The social welfare (SW) model maximizes statewide arc coverage under individual county budgets:

$$\max_{\mathbf{x}, \mathbf{y}} \sum_{(j,k) \in \mathcal{A}} w_{jk} y_{jk} \quad (3.11a)$$

$$\text{s.t. } y_{jk} \leq x_j + x_k, \quad \forall (j, k) \in \mathcal{A}, \quad (3.11b)$$

$$\sum_{j \in \mathcal{J}_i} x_j \leq \mathcal{B}_i, \quad \forall i \in N, \quad (3.11c)$$

$$x_j, y_{jk} \in \{0, 1\}. \quad (3.11d)$$

The grand coalition (GC) model pools budgets:

$$\max_{\mathbf{x}, \mathbf{y}} \phi(\mathbf{x}) \quad \text{s.t.} \quad \sum_{j \in \mathcal{J}} x_j \leq \mathcal{B}, \quad (3.11b), (3.11d). \quad (3.12)$$

The SW model first appears in Lee et al. [98]; pooled or state-county coordination formulations are related to Haight et al. [82], Kinsley et al. [94]. SW provides the social optimum under individual budgets; GC is an idealized upper bound under full pooling. The displayed models impose only $y_{jk} \leq x_j + x_k$; because all arc weights are nonnegative and the objective maximizes covered weight, an optimal solution can always be selected with $y_{jk} = \max\{x_j, x_k\}$, which for binary $\mathbf{x}$ is the logical OR of the endpoints. We adopt this

convention throughout. EBMC formulations also admit equivalent reformulations using only $\mathbf{x}$ -variables via reciprocally bilinear terms, which we suppress here for clarity.

County-level optimization. County i optimizes its own utility, taking other counties' decisions as fixed. Let $\mathcal{A}_i^+ := \mathcal{A}_i \cup \delta_-(\mathcal{J}_i)$ denote the arcs counted by county i (internal arcs plus all arcs entering $\mathcal{J}_i$). The county-level model is

$$\max_{\mathbf{x}, \mathbf{y}} u_i(\mathbf{x}) = \sum_{(j,k) \in \mathcal{A}_i^+} w_{jk} y_{jk}, \quad (3.13a)$$

$$\text{s.t. } y_{jk} \leq x_j + x_k, \quad \forall (j,k) \in \mathcal{A}_i^+, \quad (3.13b)$$

$$\sum_{j \in \mathcal{J}_i} x_j \leq \mathcal{B}_i, \quad (3.13c)$$

$$x_j = \bar{x}_j^{-i}, \quad \forall j \in \mathcal{J}_{-i}, \quad (3.13d)$$

$$x_j, y_{jk} \in \{0, 1\}. \quad (3.13e)$$

Constraint (3.13d) fixes other counties' decisions; the objective sums weighted arcs incoming to or internal to county i . The sum of player utilities equals social welfare: $\sum_{i \in N} u_i(\mathbf{x}) = \phi(\mathbf{x})$ [Corollary 1 of 98].

Game formulation and equilibrium computation. Lee et al. [98] formulate EBMC games in which each county solves its county-level model, and inter-county interactions are captured implicitly through Constraint (3.13d). To compute PNEs, we run BZR (Section 3.2.2) in both welfare-maximizing and welfare-minimizing directions, yielding a sample of equilibria covering the welfare range.

3.3.3 Three Nash-informed optimization models for EBMC

We now apply the Nash-informed phase to EBMC games. For reference utility construction (Section 3.2.3), we use the equilibria collected by BZR to build the IR constraints (3.4). We then formulate three bounded models corresponding to the three configurations of $\tilde{\mathcal{X}}$ in Section 3.2.4.

Bounded Social Welfare Model (BSW). The bounded social welfare model retains the original decentralized budget structure and optimizes total welfare subject to IR:

$$\max_{\mathbf{x}, \mathbf{y}} \phi(\mathbf{x}) = \sum_{(j,k) \in \mathcal{A}} w_{jk} y_{jk} \quad (3.14a)$$

$$\text{s.t. } y_{jk} \leq x_j + x_k, \quad \forall (j, k) \in \mathcal{A}, \quad (3.14b)$$

$$\sum_{j \in \mathcal{J}_i} x_j \leq \mathcal{B}_i, \quad \forall i \in N, \quad (3.14c)$$

$$u_i(\mathbf{x}) \geq u_i^\dagger, \quad \forall i \in N, \quad (3.14d)$$

$$x_j \in \{0, 1\}, y_{jk} \in \{0, 1\}, \quad \forall j, (j, k). \quad (3.14e)$$

Bounded Grand Coalition Model (BGC). The bounded grand coalition model pools all county budgets under the planner while retaining IR:

$$\max_{\mathbf{x}, \mathbf{y}} \phi(\mathbf{x}) \text{ s.t. } \sum_{j \in \mathcal{J}} x_j \leq \mathcal{B}, \quad (3.14b), (3.14d), (3.14e). \quad (3.15)$$

Bounded Coalition Model (BC). The bounded coalition model represents an intermediate level of coordination in which counties may form coalitions and pool budgets within each coalition. Let G denote the set of possible coalitions, $x'_{j,g} \in \{0, 1\}$ indicate that lake j is selected and assigned to coalition g , and $z_{i,g} \in \{0, 1\}$ indicate that county i is part of

coalition g :

$$\max_{\mathbf{x}', \mathbf{y}, \mathbf{z}} \phi(\mathbf{x}') = \sum_{(j,k) \in \mathcal{A}} w_{jk} y_{jk}, \quad (3.16a)$$

$$\text{s.t. } y_{jk} \leq \sum_{g \in G} (x'_{j,g} + x'_{k,g}), \quad \forall (j,k) \in \mathcal{A}, \quad (3.16b)$$

$$\sum_{g \in G} z_{i,g} = 1, \quad \forall i \in N, \quad (3.16c)$$

$$\sum_{i \in N} \sum_{j \in \mathcal{J}_i} x'_{j,g} \leq \sum_{i \in N} \mathcal{B}_i z_{i,g}, \quad \forall g \in G, \quad (3.16d)$$

$$x'_{j,g} \leq z_{i,g}, \quad \forall j \in \mathcal{J}_i, \forall i \in N, \forall g \in G, \quad (3.16e)$$

$$u'_i(\mathbf{x}') \geq u_i^\dagger, \quad \forall i \in N, \quad (3.16f)$$

$$x'_{j,g}, y_{jk}, z_{i,g} \in \{0, 1\}. \quad (3.16g)$$

The county- i utility under coalition-based strategies is $u'_i(\mathbf{x}') = \tilde{u}_i(\mathbf{y}) = \sum_{(j,k) \in \mathcal{A}_i^+} w_{jk} y_{jk}$, where $y_{jk} = \max\{\sum_{g \in G} x'_{j,g}, \sum_{g \in G} x'_{k,g}\}$. Constraints (3.16c) assign each county to exactly one coalition; (3.16d) pools budgets within each coalition; (3.16e) links lake selections to coalition membership. Budget pooling in BC and BGC is a relaxation of the planner's feasible set, not a contract that modifies payoffs: utility functions and the strategic interaction structure remain unchanged.

3.3.4 Instantiation

The EBMC social-welfare, grand-coalition, and bounded models above are the concrete instances of the general models of Section 3.2.4, obtained by taking each county's integer program as the player's strategy set. The structural guarantees therefore apply directly: the feasible-set and objective orderings of Propositions 3.1 and 3.4, and the bound $\text{PIR} \leq \text{POS}$ of Proposition 3.3. Section 3.5 reports the realized PIR (Definition 3.2) for the Minnesota study.

3.3.5 Policy-constrained BC variants

The default BC model permits arbitrary budget pooling within coalitions, which leaves the planner’s recommendation space large but may produce reallocations that real institutions cannot adopt, for instance, sharp drops in a county’s own-lake spending or coalitions that ignore existing administrative groupings. We therefore consider additional constraints that capture common policy-relevant restrictions on coalition formation and budget reallocation, and evaluate the resulting welfare-implementability tradeoffs in the computational study (Section 3.5).

We add four policy-relevant constraint families: (i) *minimum-budget retention* (each county keeps at least $\mathcal{B}_i - \alpha$ units on its own lakes); (ii) *opt-out* (unwilling counties stay singletons with their own budget); (iii) *coalition size/number* limits; and (iv) *association-based coalitions* (predefined groups such as the DNR’s four regions [56]); full formulations are in the accompanying code. In Section 3.5 we evaluate five tailored models: BC_1 (opt-out), BC_2 (size limits), and BC_3 (association-based), each also imposing minimum-budget retention with $\alpha = 2$, plus NVC and BC_NVC based on the Network Volume-based Coalition of Kinsley et al. [94] (without and with IR, respectively). Among the bounded models, BSW fits when counties retain full budget control, BGC is the voluntary-participation welfare upper bound under full pooling, and the BC variants provide a middle pathway, recovering a substantial fraction of BGC’s gains at far smaller budget shifts.

3.4 Structural Sources of the Realized PIR

This section explains why the realized PIR is small. Section 3.4.1 gives a general IPG argument supported by the empirical evidence in Sections 3.5 and 3.6. Section 3.4.2 attempts

a structural specialization to EBMC games through an externality decomposition and a per-county sufficient condition for $\text{PIR} = 1$ (Proposition 3.7); further analytical attempts, loose in different ways, are collected in Section 3.D.

3.4.1 General IPG argument

An IPG’s joint strategy set $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$ is combinatorially large (astronomically so in the EBMC, KPG, and KEG instances we study), while the PNE set $\mathcal{S}_{\text{pne}}$ is much smaller. Because BSW searches beyond the PNE set, it may identify high-welfare non-equilibrium profiles that satisfy the equilibrium-derived utility floors; Sections 3.5 and 3.6 show that such profiles arise consistently in the studied instances.

Under rule (i), every BSW solution $\bar{\mathbf{x}}_{\text{bsw}}$ weakly Pareto-improves the selected reference PNE, since it satisfies $u_i(\bar{\mathbf{x}}_{\text{bsw}}) \geq u_i^\dagger = u_i(\hat{\mathbf{x}}_{\text{pne}}^*)$ for all i via the IR constraints; under rules (ii) and (iii), BSW instead satisfies coordinatewise equilibrium-derived floors that need not correspond to, or Pareto-improve, any single PNE. This is a guarantee on utility *levels*, not on stability: whenever BSW strictly improves welfare ($z^{\text{BSW}} > \phi(\hat{\mathbf{x}}_{\text{pne}}^*)$), the recommendation $\bar{\mathbf{x}}_{\text{bsw}}$ is not itself a PNE—were it one, its welfare could not exceed that of the best PNE—so some player could profitably deviate from it. The framework’s welfare gains therefore arise precisely where the plan is not self-enforcing; voluntary adoption rests on commitment to the joint plan, evaluated against the fallback equilibrium, rather than on deviation-proofness. The empirical question (how small is the residual gap $\phi(\bar{\mathbf{x}}_{\text{sw}}^*)/\phi(\bar{\mathbf{x}}_{\text{bsw}})$, the realized PIR) is answered for the Minnesota network in Section 3.5 and across additional IPG classes in Section 3.6.

Analytically, $\text{PIR} \leq \text{POS}$ (Proposition 3.3) is the only universal bound under rules (i) and (iii), and is not guaranteed under rule (ii); we are not aware of a tighter constant bound for

the EBMC class. We formalize the mechanism through a *deficit-and-repair* reading. For each player i , the deficit $d_i := [u_i^\dagger - u_i(\bar{\mathbf{x}}_{\text{sw}}^*)]_+$ measures how far SW falls below i 's benchmark, and the aggregate repair cost is $C^{\text{IR}} := z^{\text{SW}} - z^{\text{BSW}}$, so that

$$\text{PIR} = \frac{z^{\text{SW}}}{z^{\text{BSW}}} = \frac{z^{\text{SW}}}{z^{\text{SW}} - C^{\text{IR}}}.$$

The deficits identify which IR constraints bind, but PIR is determined by both their size and the welfare cost of repairing them. It remains close to one when the deficits are small or can be eliminated through substitutions of nearly equal social value. It can be substantially larger when SW asks a player to sacrifice considerable private utility for benefits accruing mainly to others and no low-cost substitute exists, a pattern most likely under tight indivisible budgets, strongly one-directional externalities, and unique high-value social decisions with little private value to the implementing player. The deficit-and-repair interpretation applies across IPG classes, although the source of inexpensive repair differs by application—redundant knapsack choices, alternative exchange cycles, or overlapping coverage decisions may leave high-welfare IR-feasible alternatives outside the PNE set. At the 100% Minnesota budget, $C^{\text{IR}} = 708,397 - 708,176 = 221$, about 0.03% of SW welfare. Two analytical bounds and a degenerate boundary construction (an inactive $\mathcal{B}_i = 0$ county makes PIR unbounded, outside our convention) are deferred to Section 3.D. The remainder develops the EBMC externality structure behind the small repair cost.

3.4.2 EBMC arc structure and externality multiplier

We now attempt a structural specialization to EBMC games to see how far EBMC-specific structure can carry the general argument. The specialization isolates one mechanism (cross-county spillover) that admits a clean per-county sufficient condition; we will see in Section 3.5

that this condition is satisfied by only a minority of counties, indicating that the closed-form route captures only part of why the realized PIR is small. Each directed arc $(j, k) \in \mathcal{A}$ is counted by exactly one county, the county containing the destination k . Hence $\sum_{i \in N} u_i(\mathbf{x}) = \phi(\mathbf{x})$, that is, social welfare equals the sum of individual utilities (Proposition 1 of Lee et al. 98). For each lake j , let W_j denote its total incident arc weight. For lakes $j \in \mathcal{J}_i$, we further distinguish the *private weight* $W_j^{\text{priv}} \leq W_j$ that is counted by the owning county i : namely, all arcs entering j and arcs leaving j to other lakes in $\mathcal{J}_i$. The difference $W_j - W_j^{\text{priv}}$ is the externality from i 's viewpoint, namely arcs that leave j for destinations in other counties.

The structural property that drives the rest of this section is that, in EBMC games, every risky arc originates from an infested lake.

Observation 3.6 (Infested lakes are the sole sources of risky arcs). *In any EBMC game, every arc with positive weight originates at an infested lake: by the EBMC weight definition $w_{jk} = f_{jl}(1 - f_{kl})n_{jk}$, $w_{jk} > 0$ requires $f_{jl} = 1$ for some AIS type $l \in L$.*

?? 3.6 does not by itself guarantee that infested lakes carry the largest W_j ; an uninfested lake that is the destination of many high-weight incoming arcs could in principle also be high-weight. Whether infested lakes *empirically* dominate depends on how sparse they are relative to uninfested ones. We refer to lakes with the largest W_j as *hub lakes*, and verify in Section 3.5 that in the Minnesota network, hubs are predominantly infested.

Decentralized play and centralized welfare maximization differ in how they treat the externality $W_j - W_j^{\text{priv}}$. The social welfare optimizer internalizes it; the decentralized PNE does not. Define the *externality multiplier* for lake $j \in \mathcal{J}_i$ as $\beta_j := W_j / W_j^{\text{priv}}$, which measures how much the social benefit of inspecting j exceeds the private benefit to the owning county. For an uninfested lake, positive-weight outgoing arcs are absent by ?? 3.6, so its incident risky-arc weight is entirely private and $\beta_j = 1$; an infested non-hub lake may still have positive

outgoing risky arcs, so its multiplier need not equal one. For source hubs with substantial cross-county outgoing risk, $\beta_j > 1$ because the social value of inspection includes external benefits omitted from the owning county's objective.

The social planner internalizes these externalities and may therefore prioritize such source hubs more strongly than decentralized counties do; a county acting alone sees only W_j^{priv} and may instead favor non-hubs whose incoming arcs are already covered by neighbors' hubs “for free.” Best-response dynamics partially correct this, so the resulting PNE mixes hub and non-hub selections while the social optimum is hub-heavy.

The hub-heavy structure of the SW solution interacts with the IR constraints through two complementary channels. First, when $\mathbf{x}^*$ selects an infested hub in some county $i' \neq i$, the destination county i receives “free” coverage of arcs entering $\mathcal{J}_i$ from that hub—a *pure spillover channel* that contributes to $u_i(\mathbf{x}^*)$ without any own-county selection. Second, the remaining slack $u_i(\mathbf{x}^*) - s_i$ comes from county i 's own-lake selections under $\mathbf{x}^*$, which cover arcs internal to $\mathcal{J}_i$ —an *own-coverage channel* that clears the reference floor without relying on pure spillover. Proposition 3.7 identifies a sufficient condition under which external-source spillovers alone satisfy a county's reference utility; in the Minnesota instances it holds for only a minority of active counties, so spillovers and own-county selections jointly clear the floors (Section 3.5). The near-one PIR in Minnesota thus appears to arise from the low cost of repairing the few county-level deficits left by SW: high-impact source selections benefit several jurisdictions, many also provide value to their owning counties, and overlapping coverage permits alternative portfolios with similar statewide value. This favorable structure need not hold when spillovers are strongly one-directional, budgets are tight, and no substitute selection preserves both local utility and statewide coverage.

The pure spillover channel is captured as follows. For each county $i \in N$, let $s_i := \sum_{(j,k) \in \delta_-(\mathcal{J}_i): x_j^*=1} w_{jk}$ denote the total weight of arcs entering $\mathcal{J}_i$ whose external source is

selected under the SW solution.

Proposition 3.7. *If $s_i \geq u_i^\dagger$ for every $i \in N$, then the SW solution $\mathbf{x}^*$ is IR-feasible, is therefore feasible for BSW, and $\text{PIR} = 1$. The proof is given in Section 3.B.*

Proposition 3.7 converts the pure spillover channel into a per-county inequality that can be evaluated directly from the SW solution and the reference equilibrium. The condition is sufficient but conservative: pure spillover alone may be small while $u_i(\mathbf{x}^*) \geq u_i^\dagger$ still holds via the own-coverage channel. Section 3.5 reports the empirical count of counties satisfying the condition outright across budget scenarios, which provides direct evidence on the relative weights of the two channels.

3.5 Computational Study

This section reports computational evidence for the Nash-informed planning framework on the Minnesota AIS prevention case study. Section 3.5.1 describes the data. Section 3.5.2 computes the welfare range across PNEs. Section 3.5.3 presents the baseline results at 100% budget, including welfare across all models, summary metrics, and county-level patterns. Section 3.5.4 evaluates the policy-constrained BC variants. Section 3.5.5 examines budget sensitivity together with the realized PIR across budget scenarios.

3.5.1 Data and calibration

The Minnesota dataset, provided by MAISRC, reflects 2018 infestation status and uses the boat-movement generation scheme of Kao et al. [90]. We consider four AIS types (zebra mussels, starry stonewort, Eurasian watermilfoil, and spiny water fleas), each inducing a

corresponding set of risky edges. We consider the 41 counties with a documented inspection program (delegation agreements in the Minnesota DNR’s annual county-level inspection records for 2017–2023). Of these, 38 have an integer inspection budget $\mathcal{B}_i \geq 1$ at baseline (100%) funding and act as strategic players; the number with a positive integer budget ranges from 32 to 39 across the 50%–125% scenarios as scaled allocations cross the one-unit threshold. Externalities from non-participating counties remain in the network, but the non-cooperative game is played only among counties with $\mathcal{B}_i \geq 1$. The Minnesota study is not a generic synthetic-network experiment: it preserves the state’s lake geography, county ownership structure, infestation records, and county inspection-budget structure. Lake-to-lake movement volumes are survey-based estimates rather than complete direct observations, so the network is best described as survey-calibrated (empirically calibrated) rather than fully observed or synthetic. The analysis provides operational decision-support evidence for the Minnesota setting, while current deployment would require updated infestation and movement data and reoptimization.

Optimization models require a comparable county-level budget unit. We express each county’s inspection capacity in inspector-hours and standardize via the conversion ($\#$ inspectors) $\approx$ (total inspection hours)/500, where 500 hours represents a rough full-time seasonal inspector workload (about three months, June–August). When the DNR records list lake entries without an explicit county field, we match lakes to counties by name, with a consistent rule for boundary cases.

Table 3.12 (in Section 3.E) summarizes the dataset: infested lakes are far fewer than uninfested lakes, yet their average incident weight is roughly $22\times$ higher—the structural premise of Section 3.4.2. Among the 200 lakes with the largest $w(j)$, only three are non-infested.

Computational environment. All experiments throughout this chapter, including the Minnesota case study (Section 3.5) and the cross-class generalization study (Section 3.6), are run on an Intel Core Ultra 7 265F processor (2.40 GHz) with 32 GB of RAM. The optimization models are implemented in Python (3.10+) and solved with Gurobi 13.0 using 16 threads. Time limits vary by experiment and are stated where each set of runs is introduced. The implementation, the instance generators, and the scripts that reproduce every table and figure in this chapter are available at <https://github.com/HyunwooLee0429/nash-informed-planning>.

3.5.2 Computing the range of social welfare

We run BZR [98] in both welfare-maximizing and welfare-minimizing directions to compute PNEs (3,600-second limit per direction). At the 100% budget BZR identifies two distinct PNEs, with worst- and best-PNE welfare 664,909 and 665,588 against the social optimum 708,397, giving $POA = 1.07$ and $POS = 1.06$ (Table 3.13 in Section 3.E).

For the Minnesota dataset, BZR identifies two distinct PNEs at the 100% budget. BPNE is also individually best (every county’s utility under BPNE is at least as high as under any other discovered PNE), so we adopt BPNE as the reference equilibrium $\hat{\mathbf{x}}_{\text{pne}}^*$ throughout. Per the BZR design [98], certification typically incurs a residual master-MIP dual gap for $n \geq 5$; we therefore report the welfare value of the best PNE found within the time limit. Where BZR does not certify global optimality, we report the best and worst PNE *found* and treat the resulting POS and POA as incumbent-based; the theoretical definitions in Section 3.2.1 refer to the exact best and worst PNE.

3.5.3 Baseline results (Minnesota, 100% budget)

This subsection examines the Minnesota baseline in depth: welfare across all models, summary inefficiency and improvement metrics, and county-level utility and budget patterns. All Nash-informed models are solved with a 7,200-second time limit.

Welfare and lake selection. Table 3.2 jointly summarizes welfare, computation time, MIPGAP, and the number of selected infested lakes (out of 466 total) across all models. A clear pattern emerges: as the decision environment shifts from non-cooperative behavior—the non-game baseline (NG), in which each county optimizes in isolation with no strategic interaction, and the reference PNE (RPNE)—through Nash-informed refinement (BSW) to pooled-budget coordination (BC, BGC, GC), social welfare increases and the solutions allocate more inspection effort to infested lakes. The number of selected infested lakes rises from 165 (NG) to 220 (RPNE), to 302–303 (BSW/SW), and to 410–417 under the coalitional models. The bounded models do not coincide with their IR-free counterparts in general; in the baseline without additional coalition-specific constraints, BC and BGC happen to return identical objective values, in which case the BGC solution is feasible for BC and certifies BC optimality. A visualization of Table 3.2 is provided in Figure 3.5.

Table 3.2: Welfare and selected lakes across models (Minnesota, 100% budget)

Models	NG	RPNE	BSW	SW	BC	BGC	GC
$\phi(\mathbf{x})$	596,535	665,588	708,176	708,397	773,960	773,960	774,638
Time (s)	93.8	7,202.0	141.8	61.6	4,738.3	134.5	54.5
MIPGAP	0.0%	—	0.0%	0.0%	0.0%	0.0%	0.0%
Sel. Inf. (Uninf.)	165 (256)	220 (201)	302 (119)	303 (118)	410 (11)	410 (11)	417 (4)

Intuitively, purely non-cooperative counties stay conservative, prioritizing incoming risk to their own jurisdictions and thus selecting more uninfested lakes than system welfare would favor. Enforcing individual rationality and expanding coordination relaxes this stance, shift-

ing selections toward the high-incident-weight infested lakes.

Key metrics. Table 3.3 reports six summary metrics organized into two families. The *ratio* family quantifies inefficiency or IR cost: POA and POS measure the welfare gap of the worst and best PNE relative to SW; $\text{PIR} := \phi(\bar{\mathbf{x}}_{\text{sw}}^*)/\phi(\bar{\mathbf{x}}_{\text{bsw}})$ (Definition 3.2) measures the cost of IR under individual budgets, and $\text{PIR}^{\text{pool}} := \phi(\bar{\mathbf{x}}_{\text{gc}})/\phi(\bar{\mathbf{x}}_{\text{bgc}})$ is its pooled-budget analog. The *improvement* family quantifies the value of coordination above SW: $\text{PI_BC} := (\phi(\bar{\mathbf{x}}_{\text{bc}}) - \phi(\bar{\mathbf{x}}_{\text{sw}}^*))/\phi(\bar{\mathbf{x}}_{\text{sw}}^*)$ and analogously PI_BGC for BGC. While POA and POS are around 1.07 and 1.06, both PIR and PIR^{pool} are essentially 1 (1.00 and 1.0009 respectively), indicating that the welfare cost of the participation rationality constraints is small in both budget regimes. The improvement metrics show that allowing coalition-based coordination yields 9.3% welfare gains beyond SW.

Table 3.3: Headline metrics for Nash-informed and coalition models

	n	POA	POS	PIR	PIR^{pool}	PI_BC	PI_BGC
MN	38	1.07	1.06	1.00	1.0009	9.3%	9.3%

County-level utility and budget patterns. Per-county utility values and budget reallocation outcomes are reported in Table 3.14 (Section 3.E). Utility gains are broadly shared under both BSW and BC, with BC yielding a larger median improvement and a longer upper tail (Figure 3.6). The systemwide gain is therefore not obtained at any county’s expense: all 38 strategically active counties improve strictly relative to RPNE under both BSW and BC, and none declines, which is guaranteed by construction under any IR-constrained model (Figure 3.6). A small number of counties exhibit near-zero changes, consistent with cases where the RPNE utility bounds are tight and there is limited room for welfare-improving reallocation.

The Minnesota case study reveals systematic budget-reallocation patterns under coalition-based coordination. Counties with relatively excessive initial budgets, those with large budget-per-infested-lake ratios, tend to receive reduced budgets under BC/BGC, while counties facing higher infestation burdens receive more. This reallocation pattern highlights potential inefficiencies in the original budget allocations and motivates the policy-constrained BC variants studied next.

3.5.4 Tailored BC variants

We evaluate the five tailored variants defined in Section 3.3.5—the retention-constrained BC_1 (opt-out), BC_2 (coalition-size limits), and BC_3 (DNR district-based), plus the Network Volume-based NVC (without IR) and BC_NVC (with IR) of Kinsley et al. [94]—each solved with a 7,200-second time limit.

Every tailored model improves on the RPNE baseline, but by varying amounts. The default BC attains the highest welfare at the largest budget shift (Sum of $|\Delta BG|$, the total absolute across-county budget change, of 228 units); the three tailored BC models (BC_1–BC_3) deliver substantial gains while cutting budget reallocation by roughly 2.7–3.9 $\times$, and the two-coalition NVC and BC_NVC variants yield more modest gains.

Across counties, the bounded models with IR deliver positive utility gains for all 38 active counties by construction, whereas NVC—the only variant solved without IR constraints—leaves part of the distribution below zero, so some counties end up worse off than at RPNE (Figure 3.3, Section 3.E). This is the practical content of the IR constraint: it converts a plan that is better on aggregate into one that no county has a reason to refuse. The default BC imposes the sharpest budget cuts on over-resourced counties (Ratio > 100%), while BC_1–BC_3 move less under the minimum-retention rule (Table 3.10, Figure 3.7).

3.5.5 Budget sensitivity and PIR across budget scenarios

The Minnesota baseline corresponds to the 100% budget scenario. State allocations may shift over time, so we examine three additional budget scenarios (50%, 75%, and 125% of the current budget) and report the welfare results and PIR values across all four scenarios. The 50% stress test is especially timely: Minnesota’s AIS Prevention Aid appropriation is scheduled to fall from \$10 million for aids payable through 2026 to \$5 million beginning in 2027 (Minnesota Statutes §477A.19), a 50% reduction. We scale modeled county budgets proportionally for sensitivity analysis rather than as a one-to-one forecast of county spending.

At each budget level BZR finds multiple distinct PNEs (three at 50% and 75%, two at 100% and 125%), and BPNE is individually best in every scenario, so we adopt it as $\hat{\mathbf{x}}_{\text{pne}}^*$. Table 3.11 (Section 3.E) reports welfare and selected-lake counts across the four scenarios: both increase monotonically from NG to GC and with available budget, with diminishing marginal gains at higher funding (the 50% $\rightarrow$ 75% increase exceeds 100% $\rightarrow$ 125%), as the highest-impact opportunities are captured first (Figure 3.8).

Realized PIR across budgets. Table 3.4 reports the realized PIR across the four budget scenarios, under both individual and pooled budgets. The values are striking: $\text{PIR} \leq 1.0015$ for individual budgets and $\text{PIR}^{\text{pool}} \leq 1.0013$ for pooled budgets across all scenarios, so BSW recovers at least $1/1.0015 \approx 99.85\%$ of the SW upper bound. The PIR-near-one regularity therefore persists not only at the baseline budget but across a $\pm 50\%$ range of plausible funding levels. The qualitative explanation, grounded in the externality decomposition of EBMC games, is in Section 3.4.2.

Proposition 3.7 isolates the pure spillover channel of Section 3.4.2; we evaluate the condition on the Minnesota data. The count of counties satisfying $s_i \geq u_i^\dagger$ ranges from 13 of 32

active counties at the 50% budget down to 4 of 38 at the 100% budget and 4 of 39 at the 125% budget (the active-county count rises with the budget as more counties clear the $\mathcal{B}_i \geq 1$ threshold). The pure spillover channel is therefore the *minority* route through which IR is satisfied; for the remaining counties, $\mathbf{x}^*$'s own-county hub selections cover enough of county i 's own arcs, given the cross-county spillover landscape under $\mathbf{x}^*$, to close the IR deficit without requiring pure spillover alone. The realized PIR (within 0.15% of 1 across all scenarios) therefore reflects the low cost of repairing these county-level deficits: for most counties the SW plan's own-lake selections, together with cross-county spillovers, already clear the reference floor, and where a small deficit remains it can be closed by substitute selections of nearly equal social value.

Table 3.4: Realized PIR across budget scenarios

Metric	Budget			
	50%	75%	100%	125%
PIR	1.0015	1.0003	1.0003	1.0003
PIR ^{pool}	1.0013	1.0013	1.0009	1.0004

Although PIR is close to one, the constrained model is not redundant. The unconstrained social optimum, evaluated county by county, leaves specific counties below their equilibrium-derived reference utilities: writing the deficit as $d_i = [u_i^\dagger - u_i(\bar{\mathbf{x}}_{\text{sw}}^*)]_+$, a small set of counties—chiefly Isanti and Waseca—has $d_i > 0$ under SW in every budget scenario, and BSW reoptimizes the plan to restore each to at least its reference utility (Table 3.5; per-county detail in Table 3.8).

Although the aggregate SW–BSW welfare difference is small, the SW solution leaves specific counties below their equilibrium-derived benchmarks; BSW eliminates these concentrated deficits while sacrificing at most 0.15% of statewide welfare. The IR constraints therefore serve not to recover aggregate welfare but to convert a nearly optimal plan that some counties

Table 3.5: Counties left below their reference utility by the SW solution

Budget	# counties below IR under SW	Max. shortfall	Mean shortfall	PIR
50%	1	6.00%	6.00%	1.0015
75%	3	5.25%	1.99%	1.0003
100%	1	4.87%	4.87%	1.0003
125%	2	3.75%	2.03%	1.0003

Shortfall = $100 (u_i^\dagger - u_i(\bar{\mathbf{x}}_{\text{sw}}^*)) / u_i^\dagger$, over counties with $u_i(\bar{\mathbf{x}}_{\text{sw}}^*) < u_i^\dagger$ and $u_i^\dagger > 0$; full county-level results are in Section 3.E.

may reject into one that preserves every county’s equilibrium utility. The result is robust to the reference rule: because the socially best PNE is individually dominant, the average and minimum floors lie weakly below its utilities, all three rules are feasible, and PIR stays within 1.0003–1.0015 under each (Table 3.9).

3.6 Generalization Across IPG Classes

A natural question raised by the Minnesota case study (Section 3.5) is whether the realized PIR-near-one regularity is special to the EBMC–Minnesota combination or whether it reflects a broader property of Nash-informed planning. We evaluate the framework on three additional IPG instantiations spanning different game-theoretic structures.

Game classes. We consider three additional classes: (i) *synthetic EBMC random instances* with independently sampled networks and county sizes; (ii) the *Knapsack Problem Game (KPG)* of Dragotto and Scatamacchia [58], Lee et al. [98], testing Type B (positive interactions) and Type C (negative interactions in $[-20, 0]$); and (iii) the *Kidney Exchange Game (KEG)* of Carvalho et al. [34, 36], a cycle-packing game extended to length-three exchanges, where only the length-three case is non-trivial (length two has $\text{POS} = 1$) and trilinear payoff terms arise (Section 3.H). Together these span coverage, knapsack, and cycle-packing struc-

tures; linear, bilinear, and trilinear interactions; and player-count versus graph-size scaling, testing the framework outside the Minnesota network.

Experimental setup. For each instance we run BZR [98] in both welfare-maximizing and welfare-minimizing directions to obtain the best and worst PNEs found within the time limit, construct the reference utility per Section 3.2, and solve the BSW model. We use a 600-second limit per solve and 16 Gurobi threads; instance sizes (n up to 20 for EBMC random and KPG; $|V|$ up to 200 with three partition strategies for KEG) and full per-instance results and methodology are in Sections 3.F to 3.H.

Table 3.6: Headline PIR/POS/POA across three IPG classes

Game class	Scaling	n_{runs}	PIR	POS	POA	PIR < POS
EBMC random	$n_{\text{counties}} \in [2, 20]$	7	1.009	1.133	1.144	7/7
KPG Type B	$n_{\text{players}} \in [2, 20]$	7	1.000	1.008	1.053	7/7
KPG Type C	$n_{\text{players}} \in [2, 15]$	6	1.000	1.133	1.148	6/6
KEG (non-trivial)	$ V \in [60, 200]$	8	1.000	1.196	1.342	8/8
Pooled	<i>three game classes</i>	28	1.002	1.119	1.179	28/28

Class rows are means across instances within each game class; the pooled row is the mean across all 28 non-trivial instances. “Non-trivial” denotes instances with $\text{POS} > 1$; trivial KEG cases (best PNE already coincides with SW) are reported separately in Section 3.H.

KPG Type C at $n = 20$ produced no PNE within the time budget and is excluded (see Section 3.G). Where BZR did not certify global optimality, POS and POA are incumbent-based.

Findings. Table 3.6 reports headline statistics. Three observations stand out. First, the ordering $1 \leq \text{PIR} < \text{POS}$ holds strictly in every non-trivial instance across all three game classes (28/28). Because 15 of the 28 instances (all 13 KPG cases and 2 EBMC-random cases) use the average-PNE reference rule, for which Proposition 3.3 does not guarantee $\text{PIR} \leq \text{POS}$, the ordering in those cases is empirical rather than theorem-backed, and POS and POA are incumbent-based wherever BZR did not certify global optimality. Second, when measured

to three decimal places, PIR equals 1.000 in 100% of KPG and KEG instances and in 43% of EBMC random instances; the remaining EBMC cases stay within $1.000 \leq \text{PIR} \leq 1.050$. That $\text{PIR} = 1.000$ despite $\text{POS} > 1$ in these instances means the social optimum already clears every player’s reference floor; the floors are not vacuous—the same instances have inefficient equilibria—but the SW plan happens to satisfy them at no welfare cost. Third, the welfare recovery ($\text{POS} - \text{PIR}$) that the framework achieves is largest precisely where the underlying inefficiency is largest: KPG Type C reaches $\text{POS} = 1.430$ at $n_{\text{players}} = 15$ with $\text{PIR} = 1.000$ (a 43 percentage-point recovery), and a KEG instance at $|V| = 100$ exhibits $\text{POS} = 1.708$ with $\text{PIR} = 1.000$ (a 70.8 percentage-point recovery). In the tested instances, the largest welfare recoveries occur where the underlying equilibrium inefficiency is largest.

A few boundary instances (KPG Type C at $n_{\text{players}} = 20$, KEG at $|V| \geq 300$) are cases in which BZR did not certify a PNE within the time budget, reflecting the Σ_2^P -completeness of PNE existence in general IPGs [35]; details are in Sections 3.G and 3.H.

The general landscape remains open. PIR is, by construction, the ratio of two integer-program optimal values, $\phi(\bar{\mathbf{x}}_{\text{sw}}^*)/\phi(\bar{\mathbf{x}}_{\text{bsw}})$, and characterizing such ratios in closed form across IPG classes is fundamentally difficult. The instances we examine span a range of structures (coverage, knapsack, cycle-packing), interaction signs (positive, mixed, negative), and player counts (from 2 to 38), and the realized PIR remains within 1.050 of 1 across all of them. We do not claim universality: IPG classes with structural features we have not tested (more antagonistic large- n games, games with diffuse rather than concentrated externalities, or game classes outside the IPG family altogether) could in principle exhibit substantially larger PIR. Tighter analytical characterization of the realized PIR across IPG classes is a natural direction for future work.

3.7 Managerial and Policy Implications

Nash-informed planning applies to settings with three institutional features: decentralized agents with fixed budget authority, a coordinator who can compute equilibria and recommend but cannot compel, and enough information-sharing to make complete-information modeling defensible. Minnesota’s AIS program has all three. We organize the implications around four decisions a state planner can act on.

Participation repair is the first message. The unconstrained statewide optimum is nearly efficient but not automatically acceptable to every county. Across the four funding scenarios only one to three counties fall below their equilibrium benchmark under SW, and BSW removes these concentrated objections while preserving at least 99.85% of SW welfare ($\text{PIR} \leq 1.0015$; Table 3.4). At baseline this raises state-level welfare by 6.4% (from 665,588 to 708,176) and expands infested-lake coverage from 220 to 302, while respecting each county’s budget envelope, requiring no inter-county transfer, and remaining compatible with existing county budget authority, conditional on a credible commitment arrangement. The practical value of Nash-informed planning is therefore targeted participation repair, not a large sacrifice of statewide protection.

Inspection priority and sequencing. Under tight budgets, capacity should first be directed to high-impact infested source lakes whose outgoing movements expose several counties: at baseline the reference PNE selects 220 infested and 201 uninfested lakes, BSW 302 and 119, and the pooled models 410 and 11. As funding expands and the major sources are covered, selected uninfested destinations and lower-priority lakes re-enter the plan. Coordination therefore changes not only total coverage but the composition and sequencing of inspection investment.

Strategic decentralization versus budget fragmentation. At baseline, moving from the refer-

ence PNE to SW raises welfare by 42,809, whereas allowing coalition-based budget coordination adds a further 65,563. The larger remaining gain comes from moving inspection capacity across county budget boundaries, not merely from replacing equilibrium behavior—so improving the funding formula and enabling targeted pooling may matter as much as persuading counties to coordinate their lake choices.

An implementation ladder. The variants differ in what they require of the institutions, and only the first rung is directly implementable under existing county authority. BSW leaves every county’s budget envelope intact and moves only lake selections, so it needs no new instrument beyond a credible commitment to the recommended plan. The BC family moves budget across county lines and therefore requires voluntary coalition formation together with a cross-county funding instrument such as a joint-powers agreement and the corresponding board approvals. BGC and GC pool the entire statewide appropriation and would require a change to the AIS Prevention Aid allocation formula, so we report them as benchmarks for the value of fiscal coordination rather than as recommendations. With that caveat the variants support a ladder: begin with BSW when county budget authority is fixed; use retention-constrained or institution-based coalitions (BC_1 opt-out, BC_2 coalition-size limits, and BC_3 DNR district-based) when limited pooling is feasible, which recover 59–74% of the coordination gain at 58–86 units of cross-county budget shift versus 228 under unrestricted coordination; and treat BGC/GC as benchmarks for the value of broader fiscal coordination. The BC solution also diagnoses budget misalignment—counties with high budget-per-infested-lake ratios tend to lose budget under coordination while under-resourced counties tend to gain—a signal usable in budget-formula discussions even where coordination itself is politically infeasible.

Participation, not deviation-proofness. Under the complete-information model, the computed PNE is the stable decentralized fallback against which counties evaluate the recom-

mendation; because BSW guarantees each county at least its equilibrium payoff, no county has a utility-based reason to reject it in favor of reverting to that equilibrium. The guarantee concerns participation, not unilateral deviation: a BSW plan preserves equilibrium utility conditional on commitment but need not be self-enforcing as a one-shot game. In Minnesota commitment is institutionally plausible—plans are observable, planning recurs annually, the DNR provides coordination infrastructure, and deviations can affect future coordination and funding discussions. We therefore interpret BSW as a participation-compatible recommendation rather than a deviation-proof mechanism.

Robustness to funding cuts. Coordination also buffers budget cuts: even at half funding, 50% BSW selects 170 infested lakes for inspection—more than the fully-funded uncoordinated baseline (100% NG, 165). These conclusions hold across all four budget scenarios, so neither recommendation requires precise budget calibration; the guidance is nonetheless observational, and for a new network instance-specific computation remains the safest verification.

Real-world-driven evidence. The Minnesota study preserves actual lake and county structure, infestation records, county budget rules, and survey-calibrated movement estimates; the synthetic and benchmark instances of Section 3.6 test generalization. Current operational deployment would require updated infestation and movement estimates and reoptimization.

I1: Where the framework should generalize beyond AIS prevention. The three institutional features above—decentralized fixed-budget agents, a recommender without coercive authority, and sufficient information-sharing—recur across regional health coordination, cross-jurisdictional environmental management, and biosecurity. When they hold, the general IPG argument of Section 3.4.1 applies, and the cross-class evidence of Section 3.6 indicates that the PIR-near-one regularity persists across coverage, knapsack, and cycle-

packing structures. The regularity is reinforced where externality flow concentrates through a small set of high-impact sources (Section 3.4.2), as in cross-jurisdictional wildfire prevention or regional vaccine deployment.

I2: Limitations of the EBMC abstraction in real planning. Two modeling abstractions in the EBMC instantiation warrant acknowledgment. First, the coverage function is binary: selecting a lake fully covers all incident arcs, ignoring graduation in inspection effort across access points, staffing, and time. Second, each lake is treated as a single decision unit, ignoring multiple access points; lakes with several entry points could be modeled as small networks within the larger boat-movement graph. Both abstractions are inherited from the prior AIS inspection optimization literature [81, 94, 98] and are acceptable simplifications that preserve the network’s overall externality structure.

3.8 Conclusion

This chapter develops Nash-informed planning, a framework that takes the equilibrium of a decentralized integer programming game as input to a downstream welfare-improving planning step under individual rationality. We define the price of individual rationality (PIR), the welfare cost of imposing the equilibrium-based participation rationality constraints, and show it is bounded by the price of stability. Empirically it stays near 1 across the Minnesota AIS case study and 28 non-trivial instances of three further IPG classes.

For practitioners in Minnesota and analogous public-sector settings, the practical message is that state and county planning objectives, often presumed to be in tension, can be reconciled at negligible welfare cost—at most 0.15% of the unconstrained statewide optimum ($\text{PIR} \leq 1.0015$ relative to SW across all four budget scenarios). The cross-class empirical evidence

suggests that this regularity extends to other decentralized public-resource networks with concentrated externality flow.

A sharper analytical characterization of the realized PIR across IPG classes is the natural next direction. In public-sector settings where institutional structure is fixed and the planner can recommend but not mandate, the equilibrium of the decentralized game is not the end of analysis: it is the input to a planning step that, in our evidence, preserves every participant's equilibrium utility at essentially no aggregate welfare cost (PIR between 1.0003 and 1.0015 against the SW benchmark across the four Minnesota budget scenarios). Implementation still requires a credible commitment arrangement among the participants: individual rationality guarantees that no participant is worse off than its fallback, but it does not by itself make the plan self-enforcing.

Appendix

3.A Acronyms and Notation

Table 3.7 lists the acronyms used throughout this chapter.

3.B Proofs

Proof. Proof of Proposition 3.1. (i) $P_{\text{BSW}} = P_{\text{SW}} \cap \mathcal{R} \subseteq P_{\text{SW}}$ because BSW adds the individual rationality constraints $u_i(\mathbf{x}) \geq u_i^\dagger$ to SW. $P_{\text{SW}} \subseteq P_{\text{GC}}$ because the pooled feasibility set (3.9) relaxes the per-player constraints of (3.8). (ii) Summing the IR constraints $u_i(\mathbf{x}) \geq u_i^\dagger$ over i gives $z^{\text{BSW}} = \sum_i u_i(\bar{\mathbf{x}}_{\text{bsw}}) \geq \sum_i u_i^\dagger = \phi^\dagger$; the remaining inequalities follow

Table 3.7: Acronyms used throughout this chapter

Acronym	Definition
<i>General concepts</i>	
IPG	Integer Programming Game
AGT	Algorithmic Game Theory
PNE	Pure Nash Equilibrium
BPNE	Best PNE (welfare-maximizing)
WPNE	Worst PNE (welfare-minimizing)
POS	Price of Stability
POA	Price of Anarchy
IR	Individual Rationality
PIR	Price of Individual Rationality
PIR ^{pool}	Pooled-budget PIR
CE	Correlated Equilibrium
<i>Application domain</i>	
OM	Operations Management
AIS	Aquatic Invasive Species
MN	Minnesota
DNR	Minnesota Department of Natural Resources
MAISRC	Minnesota Aquatic Invasive Species Research Center
<i>Equilibrium-computation algorithm</i>	
ZR	Zero-Regret algorithm
BZR	Best-response-incorporated Zero-Regret algorithm
<i>IPG classes studied</i>	
EBMC	Edge-weighted Budgeted Maximum Coverage game
KPG	Knapsack Problem Game
KEG	Kidney Exchange Game
<i>Optimization models</i>	
NG	Non-Game baseline (each county plays in isolation; no game-theoretic interaction)
RPNE	Reference PNE (the equilibrium adopted as the IR baseline)
SW	Social Welfare model (state-level, IR-free)
GC	Grand Coalition model (pooled-budget, IR-free)
BSW	Bounded Social Welfare model (SW with IR)
BC	Bounded Coalition model (coalition pooling with IR)
BGC	Bounded Grand Coalition model (full pooling with IR)
NVC	Network Volume-based Coalition structure

Acronyms are introduced once near their first use in the body; this E-companion section provides a single point of reference.

from feasible-set inclusion and the common objective ϕ . This lower bound is rule-agnostic; under rules (i) and (iii), $\hat{\mathbf{x}}_{\text{pne}}^*$ is itself BSW-feasible (it meets its own floor), so the sharper $z^{\text{BSW}} \geq \phi(\hat{\mathbf{x}}_{\text{pne}}^*)$ also holds. $\square$

Proof. Proof of Proposition 3.3. Lower bound: $P_{\text{BSW}} \subseteq P_{\text{SW}}$ (Proposition 3.1) gives $\phi(\bar{\mathbf{x}}_{\text{sw}}^*) \geq \phi(\bar{\mathbf{x}}_{\text{bsw}})$, hence $\text{PIR} \geq 1$. Upper bound: when the IR bound uses $\hat{\mathbf{x}}_{\text{pne}}^*$, the profile $\hat{\mathbf{x}}_{\text{pne}}^*$ itself is feasible for BSW (the IR constraint $u_i(\mathbf{x}) \geq u_i(\hat{\mathbf{x}}_{\text{pne}}^*)$ is satisfied at $\mathbf{x} = \hat{\mathbf{x}}_{\text{pne}}^*$), so $\phi(\bar{\mathbf{x}}_{\text{bsw}}) \geq \phi(\hat{\mathbf{x}}_{\text{pne}}^*)$. Since $\text{POS} = \phi(\bar{\mathbf{x}}_{\text{sw}}^*)/\phi(\hat{\mathbf{x}}_{\text{pne}}^*)$,

$$\text{PIR} = \frac{\phi(\bar{\mathbf{x}}_{\text{sw}}^*)}{\phi(\bar{\mathbf{x}}_{\text{bsw}})} \leq \frac{\phi(\bar{\mathbf{x}}_{\text{sw}}^*)}{\phi(\hat{\mathbf{x}}_{\text{pne}}^*)} = \text{POS}.$$

The same upper bound $\text{PIR} \leq \text{POS}$ also holds under rule (iii) of Section 3.2.3, the per-player minimum across the collected equilibrium set $\mathcal{S}_{\text{pne}}$. Since $\hat{\mathbf{x}}_{\text{pne}}^* \in \mathcal{S}_{\text{pne}}$, the reference utility $u_i^\dagger = \min_{\mathbf{x} \in \mathcal{S}_{\text{pne}}} u_i(\mathbf{x}) \leq u_i(\hat{\mathbf{x}}_{\text{pne}}^*)$ for every player i , so the IR constraints are no more restrictive than under the best-PNE reference; the profile $\hat{\mathbf{x}}_{\text{pne}}^*$ remains feasible for BSW, and the same argument yields $\text{PIR} \leq \text{POS}$. Under rule (ii) (per-player average), in contrast, the reference utility can exceed $u_i(\hat{\mathbf{x}}_{\text{pne}}^*)$ for individual players when $\hat{\mathbf{x}}_{\text{pne}}^*$ is not individually dominant, in which case the IR constraints become more restrictive than those derived from the best PNE alone and $\text{PIR} \leq \text{POS}$ is not guaranteed. $\square$

Proof. Proof of Proposition 3.4. We project P_{BC} onto the $\mathbf{x}$ -space. The assignment and linking constraints of (3.10) place each player in exactly one coalition, so $\sum_{g \in G} \mathbf{x}'_g = \mathbf{x}^i$ for every $i \in N$; substituting into the coalitional resource constraints reduces them to $\sum_{i \in C} A^i \mathbf{x}^i \leq \sum_{i \in C} b^i$ for each active coalition C . These relax the per-player constraints of BSW and tighten the fully pooled constraint of BGC, so $P_{\text{BSW}} \subseteq \text{proj}_{\mathbf{x}} P_{\text{BC}} \subseteq P_{\text{BGC}}$; and $P_{\text{BGC}} \subseteq P_{\text{GC}}$ because BGC adds the IR constraints to GC. Objective inequalities follow from feasible-set inclusion and the rule-agnostic bound $\phi^\dagger \leq z^{\text{BSW}}$ of Proposition 3.1. $\square$

Proof. Proof of Proposition 3.7. For each $i \in N$ and each arc $(j, k) \in \delta_-(\mathcal{J}_i)$ with $x_j^* = 1$, choose the optimal coverage-variable representation associated with $\mathbf{x}^*$ so that $y_{jk} = \max\{x_j^*, x_k^*\}$, which is valid because all arc weights are nonnegative. Hence $x_j^* = 1$ implies $y_{jk}(\mathbf{x}^*) = 1$, so the arc is covered under $\mathbf{x}^*$. Summing over such arcs and using $\delta_-(\mathcal{J}_i) \subseteq \mathcal{A}_i^+$,

$$u_i(\mathbf{x}^*) = \sum_{(j,k) \in \mathcal{A}_i^+} w_{jk} y_{jk}(\mathbf{x}^*) \geq \sum_{\substack{(j,k) \in \delta_-(\mathcal{J}_i): \\ x_j^*=1}} w_{jk} = s_i.$$

The hypothesis $s_i \geq u_i^\dagger$ then gives $u_i(\mathbf{x}^*) \geq u_i^\dagger$ for every i , so $\mathbf{x}^* \in \mathcal{R}$ (the IR-feasible region) and $\mathbf{x}^*$ is feasible for the BSW model. Since the BSW objective is the same as SW, $z^{\text{BSW}} = \phi(\mathbf{x}^*) = \phi(\bar{\mathbf{x}}_{\text{sw}}^*)$, and $\text{PIR} = \phi(\bar{\mathbf{x}}_{\text{sw}}^*)/z^{\text{BSW}} = 1$. $\square$

3.C Stylized comparison of BSW with correlated equilibria

The reference-rule cascade of Section 3.2.3 constructs a deterministic, per-player utility floor; this differs structurally from a stochastic mediator recommendation, and the rule choice itself drives the comparison with the welfare-best correlated equilibrium (CE; Section 3.1.2). A small Knapsack Problem Game instance (Section 3.6) with $n = 2$ players and $m = 2$ items illustrates the contrast.

Instance. Each player $p \in \{1, 2\}$ chooses $x_k^p \in \{0, 1\}$ for $k \in \{1, 2\}$ subject to the knapsack constraint $x_1^p + x_2^p \leq 1$ (at most one item per player). Standalone profits are $\pi_1^p = 2$ and $\pi_2^p = 7$; pairwise interactions are $c_1^{pq} = 4$ (item 1 synergy) and $c_2^{pq} = -7$ (item 2 conflict). Each player's utility is the standard KPG form $u_p(\mathbf{x}) = \sum_k \pi_k^p x_k^p + \sum_{q \neq p} \sum_k c_k^{pq} x_k^p x_k^q$. The knapsack constraint reduces each player's strategy set to {item 1, item 2, nothing}; “nothing”

is dominated for both players, so we restrict attention to the four-cell payoff matrix

(u_1, u_2)	$x^2 = \text{item 1}$	$x^2 = \text{item 2}$
$x^1 = \text{item 1}$	(6, 6)	(2, 7)
$x^1 = \text{item 2}$	(7, 2)	(0, 0)

Equilibria and welfare-best CE. The pure equilibria are (item 1, item 2) and (item 2, item 1), both with $\phi = 9$. The cell (item 1, item 1) achieves $\phi = 12$ but is not a PNE (each player prefers item 2 against the other's item 1). The welfare-best CE is the uniform mixture over $\{(\text{item 1, item 1}), (\text{item 1, item 2}), (\text{item 2, item 1})\}$: a direct check confirms incentive compatibility (when player p is told “item 1,” the conditional posterior over x^{-p} makes item 1 weakly preferred), and its welfare is $\phi = (12 + 9 + 9)/3 = 10$.

BSW under each reference rule. Under rule (i) with reference (item 2, item 1), the IR floor $u_1 \geq 7$ excludes (item 1, item 1) and BSW returns the reference itself with $\phi(\bar{\mathbf{x}}_{\text{bsw}}) = 9 < 10$. Under rule (ii) the IR floor is (4.5, 4.5) and under rule (iii) it is (2, 2); both admit (item 1, item 1), so BSW returns it with $\phi(\bar{\mathbf{x}}_{\text{bsw}}) = 12 > 10$. Under our cascade neither pure equilibrium is individually best, so the instance lands in rule (ii) and BSW recovers the welfare-best joint action.

Discussion. The example demonstrates that no domination theorem holds in either direction between BSW and the best CE: the CE alternative is positioned relative to our framework, not dominated by it. The structural distinction is between *deterministic public recommendation with per-player IR floors* (our setting) and *stochastic mediator recommendation with conditional incentive compatibility* (the CE setting); each is appropriate for different institutional configurations.

3.D Analytical bounds on PIR for EBMC games

This appendix collects two analytical results on PIR for EBMC games that are referenced in Section 3.4.1 but are auxiliary to the empirical narrative of this chapter. Section 3.D.1 derives an instance-parameterized bound computable from the SW solution alone; the bound is loose on the Minnesota instance and is included to map the analytical ceiling around the structural narrative of Section 3.4.2. Section 3.D.2 exhibits a single-player degenerate construction with one inactive county ($\mathcal{B}_2 = 0$), in which PIR can be made arbitrarily large. The construction falls outside our framework convention $\mathcal{B}_i \geq 1$ for all players: the inactive county owns only a destination lake and contributes no strategic decision, so the game collapses to a single-agent problem in which the only active county must trade private coverage for cross-county externality. We include it as a boundary marker rather than as a result about the multi-player strategic setting on which this chapter centers.

3.D.1 Instance-parameterized bound

Proposition 3.8. *For the SW solution $\bar{\mathbf{x}}_{sw}^*$, let*

$$E_{\text{ext}}(\bar{\mathbf{x}}_{sw}^*) := \sum_{(j,k) \in \mathcal{A}: \bar{x}_k^{\text{sw}}=0, \bar{x}_j^{\text{sw}}=1, c(j) \neq c(k)} w_{jk}$$

denote the total weight of arcs covered under $\bar{\mathbf{x}}_{sw}^*$ solely by cross-county source selection, and let $\eta := E_{\text{ext}}(\bar{\mathbf{x}}_{sw}^*)/\phi(\bar{\mathbf{x}}_{sw}^*) \in [0, 1]$. Then every PNE $\hat{\mathbf{x}}$ of the EBMC game satisfies $\phi(\hat{\mathbf{x}}) \geq (1 - \eta) \phi(\bar{\mathbf{x}}_{sw}^*)$, and hence whenever $\eta < 1$,

$$\text{PIR} \leq \text{POS} \leq \text{POA} \leq \frac{1}{1 - \eta}$$

under the reference rules (i) and (iii) of Section 3.2.3.

Proof. Proof of Proposition 3.8. Fix any PNE $\hat{\mathbf{x}}$. For each county i , define a_i as the total weight of arcs in $\mathcal{A}_i^+$ covered by county i 's own selections under $\bar{\mathbf{x}}_{\text{sw}}^*$:

$$a_i := \sum_{\substack{(j,k) \in \mathcal{A}_i^+ : \\ j \in \mathcal{J}_i, \bar{x}_j^{\text{sw}}=1 \text{ or } \bar{x}_k^{\text{sw}}=1, \\ \text{or } j \notin \mathcal{J}_i, \bar{x}_k^{\text{sw}}=1}} w_{jk}.$$

Coverage is monotone in others' selections, so under the deviation profile $(\bar{x}_i^{\text{sw}}, \hat{\mathbf{x}}_{-i})$, all arcs counted in a_i remain covered; hence $u_i(\bar{x}_i^{\text{sw}}, \hat{\mathbf{x}}_{-i}) \geq a_i$. The equilibrium property gives $u_i(\hat{\mathbf{x}}) \geq u_i(\bar{x}_i^{\text{sw}}, \hat{\mathbf{x}}_{-i}) \geq a_i$ for every $i \in N$. Summing and using $\sum_i u_i = \phi$ (Corollary 1 of Lee et al. 98),

$$\phi(\hat{\mathbf{x}}) \geq \sum_{i \in N} a_i.$$

Each arc (j, k) covered under $\bar{\mathbf{x}}_{\text{sw}}^*$ lies in exactly one $\mathcal{A}_i^+$, namely $i = c(k)$, and is counted in $a_{c(k)}$ unless $j \notin \mathcal{J}_{c(k)}$ and $\bar{x}_k^{\text{sw}} = 0$, i.e., unless it is externality-covered. Hence $\sum_i a_i = \phi(\bar{\mathbf{x}}_{\text{sw}}^*) - E_{\text{ext}}(\bar{\mathbf{x}}_{\text{sw}}^*) = (1 - \eta) \phi(\bar{\mathbf{x}}_{\text{sw}}^*)$. The POA bound follows; POS and PIR bounds follow from $\text{PIR} \leq \text{POS} \leq \text{POA}$ (Proposition 3.3). $\square$

The argument is in the smoothness tradition of Roughgarden [127], but is non-trivial because EBMC games are not valid utility systems in the sense of Vetta [139]—the externality-covered arcs are precisely the source of failure of Vetta's marginal-contribution condition. To our knowledge, the parameterization by externality-covered fraction η is new in the EBMC setting. The value of Proposition 3.8 is qualitative direction and a priori certification, not tightness on the Minnesota instance: η ranges from 0.638 at the 125% budget to 0.742 at the 50% budget, giving an a priori bound of $1/(1 - \eta) \in [2.76, 3.88]$ on POA, while the realized POA on the same scenarios lies in $[1.05, 1.20]$; matching the bound and the realized POA scenario by scenario, the bound exceeds the realized POA by a factor of $3.88/1.20 \approx 3.23$ at the 50% budget down to $2.76/1.05 \approx 2.63$ at the 125% budget. The bound is loose by

design—it certifies that PIR remains controlled before any equilibrium is computed, but does not capture how favorable the realized equilibrium structure of the Minnesota network is.

Remark 3.9 (EBMC distance from valid-utility class). EBMC games are not valid utility systems in the sense of Vetta [139], whose framework would otherwise give $\text{POA} \leq 2$ free: the social function ϕ is monotone submodular coverage and $\sum_i u_i = \phi$ holds, but Vetta’s marginal-contribution condition $u_i(\mathbf{x}) \geq \phi(\mathbf{x}) - \phi(\mathbf{x} \setminus \mathbf{x}_i)$ fails. The shortfall at a hub lake $h \in \mathcal{J}_i$ equals the outgoing cross-county weight $W_h - W_h^{\text{priv}}$, which the owning county’s utility omits. The externality multiplier $\beta_j = W_j/W_j^{\text{priv}}$ thus measures the distance of EBMC from Vetta’s valid-utility class, and is a tidy structural reason why $\text{POS} > 1$ holds for EBMC at all.

3.D.2 Degenerate single-player construction with unbounded PIR

The construction below shows that if the framework convention $\mathcal{B}_i \geq 1$ is relaxed to allow an inactive county owning only a destination lake, then PIR in an EBMC instance can be made arbitrarily large. The resulting game is single-player in the strategic sense (only one county chooses a selection); we record it as a boundary marker.

Proposition 3.10. *There exists a sequence of EBMC instances, parameterized by $M > 1$ and each containing one active county ($\mathcal{B}_1 = 1$) and one inactive county ($\mathcal{B}_2 = 0$), in which $\text{PIR} = \text{POS} = \text{POA} = M$. Consequently $\text{PIR} \leq \text{POS}$ (Proposition 3.3) is tight, and no constant upper bound on PIR holds across the closure of the EBMC class that includes inactive counties.*

Proof. Proof of Proposition 3.10. Consider the EBMC instance shown in Figure 3.2. County 1 owns lakes h (infested), d (uninfested), and s' (infested) with budget $\mathcal{B}_1 = 1$. County 2

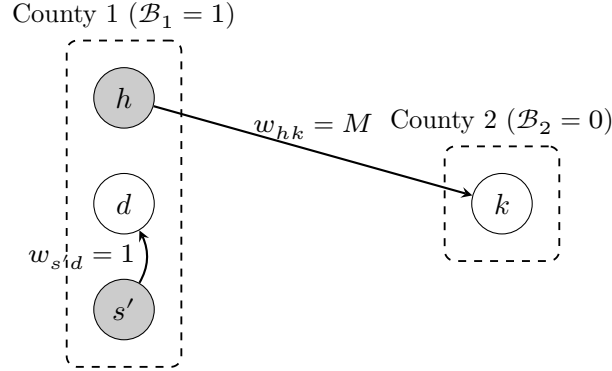


Figure 3.2: Degenerate EBMC instance used in the proof of Proposition 3.10. Shaded lakes are infested; dashed boxes mark counties.

owns only the destination lake k (uninfested) and is inactive ($\mathcal{B}_2 = 0$). The arcs are (h, k) with weight M (crossing to County 2) and (s', d) with weight 1 (internal to County 1), so $\mathcal{A}_1^+ = \{(s', d)\}$ and $\mathcal{A}_2^+ = \{(h, k)\}$. Since $\mathcal{B}_2 = 0$, county 2's strategy is fixed at $x_k = 0$, and the game reduces to county 1's single-player decision over $\{x_h, x_d, x_{s'}\}$ with $x_h + x_d + x_{s'} \leq 1$. County 1's utility is $u_1(\mathbf{x}) = w_{s'd} y_{s'd} = \max(x_{s'}, x_d)$.

Equilibrium welfare. County 1's best responses are $x_d = 1$ or $x_{s'} = 1$ (both cover (s', d) , giving $u_1 = 1$); selecting $x_h = 1$ yields $u_1 = 0$ and is dominated. In every best response, $x_h = 0$, so $y_{hk} = 0$ and $u_2(\hat{\mathbf{x}}_{\text{pne}}^\dagger) = 0$. Hence $\phi(\hat{\mathbf{x}}_{\text{pne}}^\dagger) = 1$.

SW. The social planner maximizes $\phi(\mathbf{x}) = u_1 + u_2 = \max(x_{s'}, x_d) + Mx_h$ subject to $x_h + x_d + x_{s'} \leq 1$. The maximum is achieved at $x_h = 1, x_d = x_{s'} = 0$, yielding $\phi(\bar{\mathbf{x}}_{\text{sw}}^*) = M$.

BSW. The IR constraint $u_1(\mathbf{x}) \geq u_1(\hat{\mathbf{x}}_{\text{pne}}^\dagger) = 1$ forces $x_d = 1$ or $x_{s'} = 1$, which (by budget) forces $x_h = 0$. Hence $z^{\text{BSW}} = \phi(\bar{\mathbf{x}}_{\text{bsw}}) = 1$.

Conclusion. The PNE coincides with WPNE and BPNE, so $\text{POS} = \text{POA} = \phi(\bar{\mathbf{x}}_{\text{sw}}^*) / \phi(\hat{\mathbf{x}}_{\text{pne}}^\dagger) = M$, and $\text{PIR} = \phi(\bar{\mathbf{x}}_{\text{sw}}^*) / z^{\text{BSW}} = M$. Letting $M \rightarrow \infty$ shows no constant upper bound on PIR holds within the closure of EBMC instances that includes inactive counties. $\square$

The mechanism of Proposition 3.10 relies on County 2 carrying no strategic decision: were $\mathcal{B}_2 \geq 1$, County 2 would select k in any best response and cover the high-weight arc (h, k) , collapsing PIR to one. The construction therefore illustrates the role of the strict-positivity convention $\mathcal{B}_i \geq 1$ adopted in Sections 3.2.1 and 3.3.2, rather than a property of multi-player EBMC games. We are not aware of a multi-player EBMC instance ($\mathcal{B}_i \geq 1$ for all $i \in N$) that admits unbounded PIR; the question is left open.

3.E Supplementary computational figures

County-level participation deficits under SW. Table 3.8 lists every county left below its reference utility by the social-optimum (SW) solution across the four budget scenarios; all active counties omitted have zero deficit. The deficits are small and concentrated in a few low-budget counties (chiefly Isanti and Waseca). The $u_i(\bar{\mathbf{x}}_{\text{bsw}})$ column shows that BSW restores each such county to at least its reference utility $u_i^\dagger$; this repair is precisely why the BSW solution departs from SW, and it is achieved at negligible aggregate welfare cost (realized PIR ≤ 1.0015 against the SW optimum).

Table 3.8: County-level participation deficits under the SW solution

Budget	County	$u_i^\dagger$	$u_i(\bar{\mathbf{x}}_{\text{sw}}^*)$	$u_i(\bar{\mathbf{x}}_{\text{bsw}})$	Deficit d_i	Shortfall
50%	Isanti	3,517	3,306	3,558	211	6.00%
75%	Waseca	2,836	2,687	3,043	149	5.25%
75%	Lake of the Woods	998	992	998	6	0.60%
75%	Isanti	4,223	4,218	4,223	5	0.12%
100%	Waseca	3,062	2,913	3,201	149	4.87%
125%	Waseca	3,123	3,006	3,244	117	3.75%
125%	Isanti	4,988	4,972	5,022	16	0.32%

$d_i = [u_i^\dagger - u_i(\bar{\mathbf{x}}_{\text{sw}}^*)]_+$; reference utilities $u_i^\dagger$ are from the socially best PNE. In every row $u_i(\bar{\mathbf{x}}_{\text{bsw}}) \geq u_i^\dagger$: BSW reoptimizes the plan to repair each deficit SW leaves, which is why the BSW solution differs from SW.

PIR under alternative reference rules. Table 3.9 reports the realized PIR for the Minnesota study under the three reference rules of Section 3.2.3. Because the socially best PNE is individually dominant over the collected equilibria, the average- and minimum-reference floors lie weakly below its per-player utilities, so all three rules are feasible.

County-level utility gains under the tailored coalition models. Figure 3.3 reports the distribution of per-county utility improvements over RPNE under each tailored model, one box per model over the 38 strategically active counties. Utility is covered risky-arc weight, that is, expected risky boat movements intercepted, and the vertical axis is the change relative to each county's RPNE utility, so the zero line is the equilibrium benchmark. Boxes span the interquartile range, the line inside each box is the median, whiskers extend to $1.5\times$ the interquartile range, and points beyond them are individual counties. The IR-constrained variants (BC_1–BC_3, BC_NVC) lie entirely above zero; NVC, which omits the IR constraints, does not.

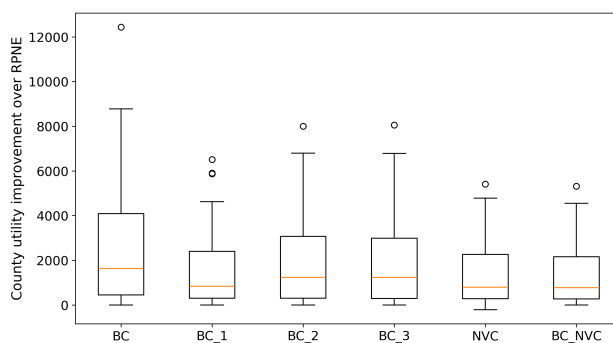


Figure 3.3: County-level utility improvements relative to RPNE under the tailored coalition models.

This E-companion section collects auxiliary figures from Section 3.5. The tables in the main text contain the underlying numerical content; the figures here provide visual summaries.

Table 3.9: Realized PIR under the three reference rules (Minnesota)

Budget	Socially best-PNE (i)	Average-PNE (ii)	Minimum-PNE (iii)
50%	1.0015	1.0015	1.0015
75%	1.0003	1.0003	1.0003
100%	1.0003	1.0003	1.0003
125%	1.0003	1.0003	1.0003

The best-PNE column reproduces Table 3.4. Average- and minimum-reference values are computed by re-solving BSW under those rules; all three rules are feasible in every scenario, and PIR remains within 0.15% of one under each.

Supplementary main-text tables. The following tables are referenced from Section 3.5 and relocated here to respect the main-text page limit.

Coalition maps for the tailored models. Figure 3.4 draws the resulting coalition structures on the Minnesota county map. Panels (a)–(d) show BC_1 (opt-out), BC_2 (coalition-size limits), BC_3 (DNR district-based), and the Network Volume-based structure shared by NVC and BC_NVC. Counties drawn in the same color belong to the same coalition; gray marks counties that do not participate in any coalition and plan on their own budget.

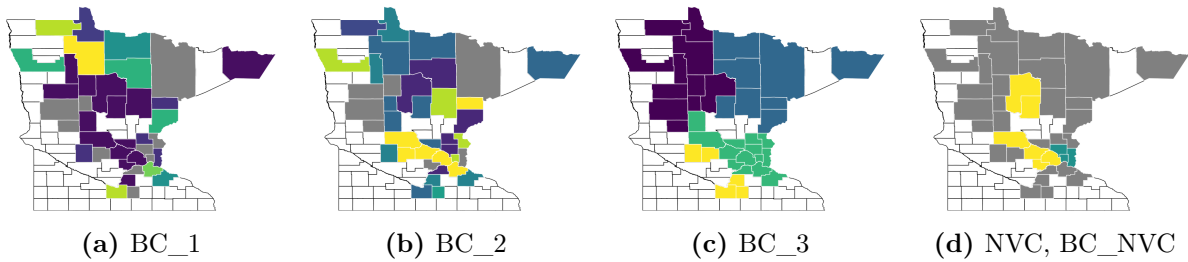


Figure 3.4: Coalition configurations across the tailored models.

Welfare and lake selection at the baseline. Figure 3.5 visualizes Table 3.2, showing social welfare and selected-infested-lake counts side by side across all models at the 100% budget. Bars give social welfare $\phi(\mathbf{x})$ on the left axis, in units of covered risky-arc weight, and the line gives the count of selected infested lakes on the right axis. Models are or-

Table 3.10: Tailored BC models (Minnesota, 100% budget)

Models	RPNE	BC	BC_1	BC_2	BC_3	NVC	BC_NVC
$\phi(\mathbf{x})$	665,588	773,960	729,879	745,759	745,329	722,212	721,001
Time (s)	7,202.0	4,640.7	2,705.7	6,460.6	204.3	123.6	242.7
MIPGAP	—	0.0%	0.1%	0.1%	0.1%	0.0%	0.0%
Sel. Inf. (Uninf.)	220 (201)	410 (11)	324 (97)	335 (86)	335 (86)	311 (110)	308 (113)
Sum of $ \Delta BG $	0	228	58	86	84	48	46

Table 3.11: Welfare and selected infested lakes under varying budget scenarios

Metric	Budget	Models						
		NG	RPNE	BSW	SW	BC	BGC	GC
$\phi(\mathbf{x})$	50%	439,410	462,518	544,101	544,917	603,218	603,218	603,996
	75%	539,334	587,691	654,179	654,389	725,648	725,648	726,619
	100%	596,535	665,588	708,176	708,397	773,960	773,960	774,638
	125%	642,534	705,678	736,626	736,839	783,645	783,645	783,947
Sel. Inf. (Uninf.)	50%	100 (101)	112 (90)	170 (31)	171 (30)	201 (0)	201 (0)	201 (0)
	75%	137 (175)	168 (144)	247 (65)	248 (64)	311 (1)	311 (1)	311 (1)
	100%	165 (256)	220 (201)	302 (119)	303 (118)	410 (11)	410 (11)	417 (4)
	125%	194 (340)	261 (273)	339 (195)	341 (193)	440 (95)	440 (95)	437 (98)

Counts are for selected lakes in the model solution.

Table 3.12: Descriptive statistics for the MN dataset

	Counties	Lakes	# Infested	Avg. $w(j)$ (Inf.)	# Uninfested	Avg. $w(j)$ (Uninf.)
Total	87	9,182	506	2,278.26	8,676	98.76
Active	38	7,515	466	2,246.18	7,049	99.25

$w(j) = \sum_{(j,k): j \in \mathcal{J}} w_{jk} + \sum_{(k,j): k \in \mathcal{J}} w_{kj}$ is the total incident weight on risky edges with at least one infested endpoint.

Table 3.13: Welfare range of the EBMC game (Minnesota, 100% budget)

		Social Welfare (ϕ)		Inefficiency			Time (s)		# PNEs	ind_better
	n	WPNE	BPNE	$\phi(\bar{\mathbf{x}}_{\text{sw}}^*)$	POA	POS	WPNE	BPNE		
MN	38	664,909	665,588	708,397	1.07	1.06	3,601.8	3,600.2	2	TRUE

dered by increasing coordination, from the non-game baseline (NG) through the reference equilibrium (RPNE) and the IR-constrained plan (BSW) to the pooled-budget models (BC, BGC, GC), with the IR-free statewide optimum (SW) shown for comparison. Both metrics

rise monotonically as the decision environment moves from non-cooperative (NG, RPNE) through Nash-informed refinement (BSW) to pooled-budget coordination (BC, BGC, GC).

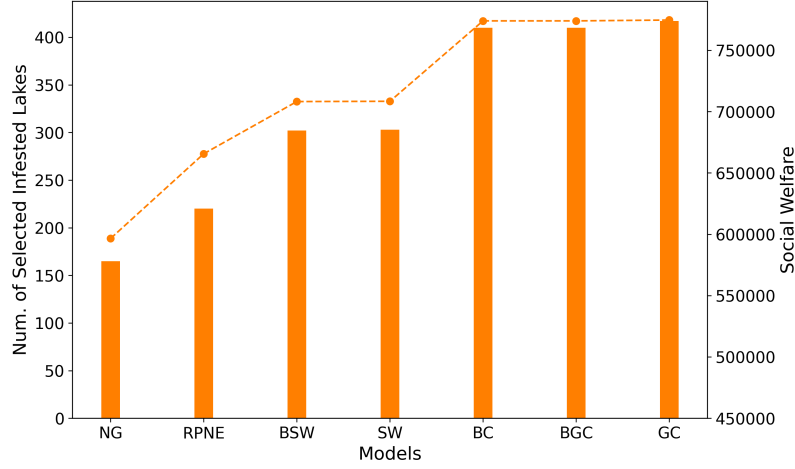


Figure 3.5: Social welfare and selected infested lakes across models (Minnesota, 100% budget).

Distribution of county-level utility improvements. Figure 3.6 reports the distribution of per-county utility improvements over RPNE under BSW and BC, drawn in the same way as Figure 3.3. Both distributions sit strictly above zero, confirming that the IR constraints are respected across all 38 counties; BC yields a higher median improvement and a longer upper tail than BSW.

Budget reallocation for high-ratio counties. Figure 3.7 compares allocated budgets across coalition models for counties with relatively excessive initial budgets (Ratio > 100%). The horizontal axis lists these counties and the vertical axis is the post-coordination budget minus the baseline budget $\mathcal{B}_i$, in inspection-station units, so negative values indicate capacity moved out of the county; one series is drawn per coalition model. The default BC model imposes sharp reductions; BC_1–BC_3 exhibit milder decreases consistent with the

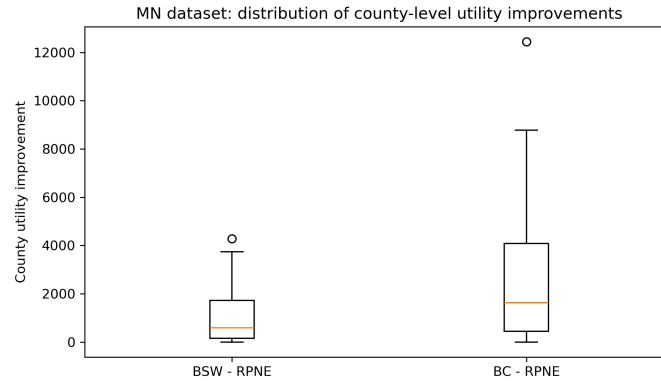


Figure 3.6: County-level utility improvements relative to RPNE under BSW and BC.

minimum-budget-retention rule; NVC and BC_NVC induce only limited budget movement at these high-ratio counties.

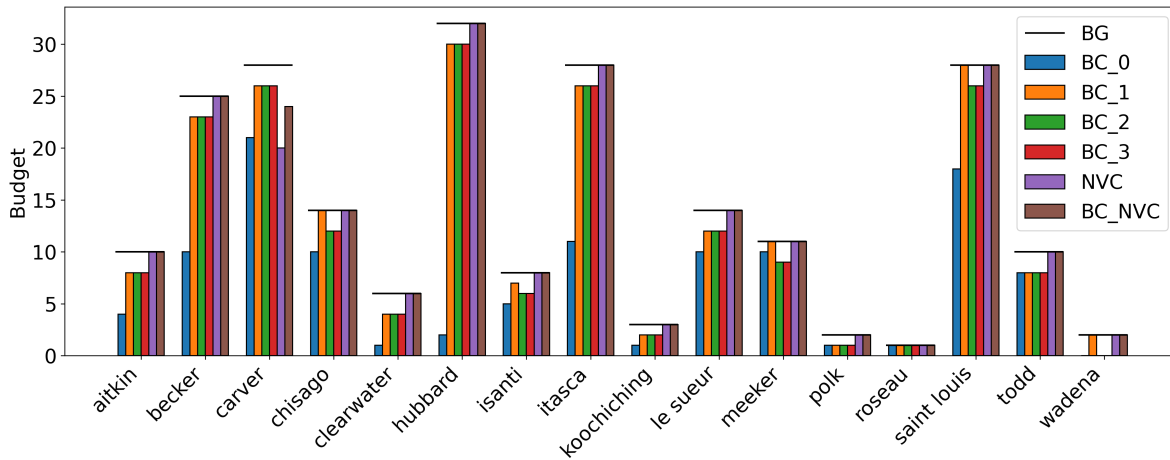


Figure 3.7: Budget reallocations across coalition models for counties with a high baseline budget per infested lake.

Welfare and lake-selection trends across budgets. Figure 3.8 visualizes Table 3.11, showing how social welfare and selected-infested-lake counts evolve across the four budget scenarios (50%, 75%, 100%, 125%) and seven models. The horizontal axis is the budget scenario and each series is one model (NG, RPNE, BSW, SW, BC, BGC, GC); the left panel reports social welfare $\phi(\mathbf{x})$ in units of covered risky-arc weight and the right panel the number

of selected infested lakes. Marginal gains diminish at higher funding levels, consistent with the highest-impact opportunities being captured earlier as budgets expand.

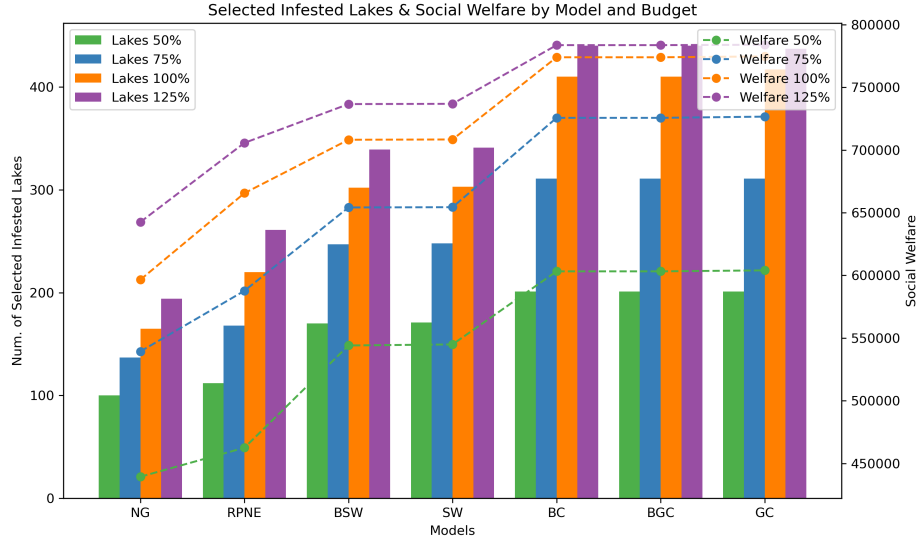


Figure 3.8: Social welfare and selected infested lakes across models under the four budget scenarios.

Selected county-level utilities and budget reallocation. Table 3.14 shows representative county-level utility values and budget reallocation outcomes for four counties at the 100% budget. Utility gains are broadly shared under both BSW and BC; “Ratio” > 100% counties (Aitkin, Becker) lose budget under coordination while lower-ratio counties retain or gain budget.

Table 3.14: Selected county-level utilities and budget reallocation (Minnesota, 100% budget)

County	NG	RPNE	BSW	SW	BC	BGC	GC	BG	# Inf.	Ratio	Re_BG
Aitkin	22,010	24,665	26,506	26,521	28,054	28,054	28,118	10	3	333%	4
Anoka	11,722	14,052	15,652	15,659	17,831	17,831	17,867	6	10	60%	9
Becker	26,861	29,892	31,664	31,681	32,993	32,993	33,053	25	10	250%	10
Beltrami	18,955	21,393	22,917	22,942	25,197	25,197	25,210	9	10	90%	10

“BG” is the original budget; “# Inf.” is the number of infested lakes in the county; “Ratio” = BG/# Inf. (treated as ∞ when # Inf. = 0); “Re_BG” is the redistributed budget under BC and BGC. Boldfaced “Re_BG” values indicate counties whose redistributed budget is lower than the original.

3.F EBMC random instances: detailed results

This E-companion section reports the detailed setup and per-instance results for the EBMC random study summarized in Section 3.6.

Instance generation. Each random EBMC instance is generated by a four-AIS-species variant of the Saidman-style boat-movement simulator. For a target number of counties $n \in \{2, 3, 5, 8, 10, 15, 20\}$, we assign each county a two-letter identifier (AA, AB, ...) and 30 lakes labeled AA1, AA2, ..., AA30. For each county and each of four AIS species (zebra mussel, spiny water flea, Eurasian watermilfoil, starry stonewort), exactly $\lfloor 30 \cdot p_s \rfloor$ lakes are flagged infested, with species-level prevalences $(p_{zm}, p_{ss}, p_{ew}, p_{sf}) = (0.5, 0.4, 0.3, 0.2)$. Candidate ordered lake pairs are enumerated, 80% of them are dropped uniformly at random, and the remaining pairs are kept as arcs only if $b_{ij} = \#\{s : f_i^s = 1, f_j^s = 0\} \geq 1$. Arc weights n_{ij} are drawn uniformly from $\{10, 11, \dots, 30\}$. Per-county budgets are drawn as $\lfloor U(0.3, 0.7) \cdot |\mathcal{J}_c| \rfloor$, where $|\mathcal{J}_c|$ denotes the county's infested-lake count. All draws use a fixed seed for reproducibility; the seven bundled instances are deterministic.

Solver configuration. We use BZR [98] with a 600-second time limit per direction. SW and BSW are each solved with a 600-second time budget and a 10^{-4} MIP gap target. All solves use 16 Gurobi threads. Because BSW returns a SW-feasible solution under additional IR constraints, $BSW \leq SW$ holds at optimality; SW values are computed independently of BSW.

Per-instance results. Table 3.15 reports, for each instance, the network size (number of lakes and risky arcs), total budget across counties, social welfare values for all four reference quantities $(SW, \hat{\mathbf{x}}_{\text{pne}}^*, \hat{\mathbf{x}}_{\text{pne}}^0, BSW)$, the number of distinct PNEs identified by BZR, and the

realized POS, POA, and PIR.

Table 3.15: EBMC random instances: full per-instance results

n	lakes	arcs	$\sum_c \mathcal{B}_c$	SW	BPNE	WPNE	BSW	#PNE	POS	PIR
2	60	424	10	5,556	5,156	4,871	5,533	2	1.078	1.004
3	90	1,035	31	20,499	18,052	18,052	20,456	1	1.136	1.002
5	150	2,840	54	56,298	51,029	50,570	56,298	2	1.103	1.000
8	240	7,242	103	159,810	142,513	142,513	159,754	1	1.121	1.000
10	300	11,284	109	223,020	189,798	189,477	223,018	3	1.175	1.000
15	450	25,697	197	555,879	492,174	492,174	553,615	1	1.129	1.004
20	600	44,192	216	876,652	738,425	738,425	835,089	1	1.187	1.050

POA = SW / WPNE, POS = SW / BPNE, PIR = SW / BSW. Counties are non-coalitional (the random instances do not use the active-counties partition).

Observations. The POS values cluster between 1.078 and 1.187, reflecting moderate equilibrium inefficiency at every scale. The realized PIR remains at or near 1 in every instance, ranging from exact 1.000 at $n \in \{5, 8, 10\}$ to 1.050 at $n = 20$. The latter is the largest deviation observed and corresponds to a tight budget regime (216 inspectors across 600 lakes); it is the only random-EBMC case in which the framework recovers visibly less than the full POS – 1 gap. Across the seven instances, the framework recovers an average of 12.4 percentage points of efficiency loss, ranging from 7.4 ($n = 2$) to 17.5 ($n = 10$) percentage points.

Reference utility construction. For two of the seven instances ($n \in \{5, 10\}$), BZR identifies multiple distinct PNEs in which no single PNE is individually best for all counties; the reference utility falls back to the per-county arithmetic mean (the construction in Section 3.2.3). The remaining five instances admit an individually-dominant best PNE. In all seven cases, the resulting BSW is feasible without invoking the min-PNE fallback.

3.G Knapsack Problem Game: detailed results

This E-companion section reports the detailed setup and per-instance results for the KPG study summarized in Section 3.6.

Game description. The KPG models n players, each solving an individual knapsack problem over $m = 100$ items. Player p selects $x_k^p \in \{0, 1\}$ for each item subject to a weight constraint $\sum_k w_k^p x_k^p \leq \mathcal{B}^p$. The payoff of player p is bilinear in the joint selection: $u_p(\mathbf{x}) = \sum_k \pi_k^p x_k^p + \sum_{q \neq p} \sum_k c_k^{pq} x_k^p x_k^q$, where π_k^p is the standalone profit of item k for player p and c_k^{pq} is the interaction coefficient. We follow the PNE-finding setting introduced by Dragotto and Scatamacchia [58] and extended to large-scale instances by Lee et al. [98].

The bundled instance families are: Type A draws c_k^{pq} equally and uniformly from $[1, 100]$ (potential game; positive equal interactions across items); Type B draws c_k^{pq} independently and uniformly from $[1, 100]$ (positive but heterogeneous interactions). The original Type C of Dragotto and Scatamacchia [58] draws $c_k^{pq} \sim \text{Unif}[-100, 100]$, but instances under this distribution often fail to admit a PNE even at $n = 2, 3$, and the issue worsens at larger n . We therefore adopt the modified Type C of Lee et al. [98], in which $c_k^{pq} \sim \text{Unif}[-20, 0]$: this preserves negative (adversarial) interactions while reducing their magnitude to a level that more frequently admits PNEs at scale. We focus on Types B and C: Type A reduces to a potential-game regime with guaranteed PNE existence and structurally bounded inefficiency, whereas Types B and C exhibit substantively different (positive vs. negative) interaction profiles relevant to testing the framework.

Instance set. We use the $n_{\text{items}} = 100$, ratio = 0.5 subset of the Dragotto and Scatamacchia [58] bundled instances at $n_{\text{players}} \in \{2, 3, 5, 8, 10, 15, 20\}$ for Type B and at $n_{\text{players}} \in \{2, 3, 5, 8, 10, 15\}$ for Type C. (Type C at $n_{\text{players}} = 20$ produced no PNE within the 600-

second budget; see the boundary discussion below.)

Solver configuration. The KPG master MIP includes a McCormick lift for each $y_k^{pq} = x_k^p x_k^q$, and BZR adds strategic dominance and payoff inequalities as user cuts for $n \leq 15$. For $n_{\text{players}} = 20$ the strategic inequality set is skipped to avoid combinatorial blowup. All solves use 16 Gurobi threads and a 600-second time budget.

Per-instance results. Table 3.16 reports per-instance social welfare values, the number of distinct PNEs, and the realized POS, POA, and PIR. POA and POS values quote the ratio of SW to WPNE and BPNE respectively; PIR quotes the ratio of SW to BSW.

Table 3.16: KPG (Types B and C) at $n_{\text{items}} = 100$, ratio = 0.5: full per-instance results

Type	n	SW	BPNE	WPNE	BSW	#PNE	POS	POA	PIR
B	2	14,390	14,246	13,675	14,390	478	1.010	1.052	1.000
B	3	29,393	28,942	26,939	29,393	464	1.016	1.091	1.000
B	5	78,841	78,004	74,250	78,841	285	1.011	1.062	1.000
B	8	189,885	188,581	180,014	189,885	156	1.007	1.055	1.000
B	10	289,156	287,886	278,853	289,156	63	1.004	1.037	1.000
B	15	644,218	642,233	618,308	644,218	35	1.003	1.042	1.000
B	20	1,126,092	1,123,062	1,092,597	1,126,092	23	1.003	1.031	1.000
C	2	7,809	7,757	7,718	7,809	10	1.007	1.012	1.000
C	3	10,407	10,295	10,245	10,407	14	1.011	1.016	1.000
C	5	14,617	14,318	14,026	14,617	41	1.021	1.042	1.000
C	8	17,561	15,758	15,517	17,561	21	1.114	1.132	1.000
C	10	20,275	16,725	16,464	20,273	21	1.212	1.232	1.000
C	15	23,039	16,115	15,824	23,039	11	1.430	1.456	1.000

Values rounded to three decimal places. SW and BSW computed under a 600 s limit. At $n = 10$ and $n = 20$ the SW solve is warm-started with the BSW incumbent so that $z^{\text{SW}} \geq z^{\text{BSW}}$ (both attain the same value), consistent with BSW being an IR-restricted SW; PIR = 1.000 throughout.

Observations. Type B exhibits mild inefficiency: POS shrinks with n_{players} (from 1.010 to 1.003), reflecting the modest interaction magnitudes built into the family. In Type C,

POS grows sharply with player count—from 1.007 at $n_{\text{players}} = 2$ to 1.430 at $n_{\text{players}} = 15$ —because the predominantly negative interactions amplify the equilibrium–SW gap as the number of contributing players grows. Despite the dramatic Type C inefficiency, Nash-informed planning recovers PIR to 1.000 in every Type C case. The framework’s recovery is therefore most valuable precisely in the family where the underlying inefficiency is most severe.

Type C boundary case ($n_{\text{players}} = 20$). For Type C at $n_{\text{players}} = 20$, BZR did not certify a PNE in either direction within the 600-second time budget; we therefore omit this instance from the headline table. The interpretation of such non-certification cases is discussed in Section 3.6. In every Type C instance where BZR did certify a PNE, the Nash-informed planning framework recovered SW to within numerical precision.

Reference utility construction. All KPG instances use the per-player arithmetic mean (the construction in Section 3.2.3) as the reference utility. These KPG instances typically admit multiple distinct PNEs without a single individually-dominant equilibrium, so the dominant-PNE scheme is rarely active. This contrasts with the Minnesota EBMC case study, where the best PNE is individually best across counties.

3.H Kidney Exchange Game: detailed results

This E-companion section reports the detailed setup and per-instance results for the KEG study summarized in Section 3.6.

Game description. The KEG models a kidney-exchange program shared by n_{players} transplant centers. Each donor–patient pair is owned by one player; the compatibility graph $G = (V, E)$ has $|V|$ pairs (nodes) and directed edges (i, j) when donor of i is compatible with patient of j . A feasible solution selects a packing of edge-disjoint directed cycles of length at most L_{max} ; each player’s utility equals the number of its own pairs covered by the packing. The KEG has been studied under different mechanisms: Carvalho et al. [34] establish mathematical foundations under a mediator-based formulation, while Carvalho et al. [36] develop the mediator-free IPG formulation that we adopt here. For the 2-player case with $L_{\text{max}} = 2$, Carvalho et al. [34, Theorem 3] prove that a social welfare equilibrium always exists, so $\text{POS} = 1$ for that case and the inefficiency question is trivial. For $L_{\text{max}} \geq 3$, the analytical status of POS is open even in the 2-player case: Carvalho et al. [36] disprove the potential-function structure conjectured for length-3 cycles in their supplementary material, so $\text{POS} = 1$ cannot be inferred from a potential-game argument. Because the $L_{\text{max}} = 2$ case is trivial for testing the Nash-informed planning framework—PIR collapses to 1 by $\text{POS} = 1$ —we extend the analysis empirically to $L_{\text{max}} = 3$, where POS can exceed 1 and the framework’s ability to recover SW is non-trivially exercised. With $L_{\text{max}} = 3$, the cycle catalogue partitions into internal cycles (one player), bilateral cycles (two players, bilinear interaction), and trilateral cycles (three players, trilinear interaction). The trilinear terms are McCormick-linearized in the master MIP.

Instance set and partition strategies. We use the Constantino et al. [48] kidney-exchange instances:

- *Small dataset:* $|V| \in \{60, 70, 80, 90, 100\}$ with 50 instances per size; we use instances $\{1, 2, 3\}$.
- *Large dataset:* $|V| = 200$ with 10 instances; we use instances $\{1, 2, 3\}$.

For each instance we apply three partition strategies that assign the $|V|$ pairs to $n_{\text{players}} = 3$ players: *uniform* (random equal split), *geographic* (synthetic regional clustering), and *blood_type* (group by patient blood type). This produces $5 \times 3 \times 3 + 1 \times 3 \times 3 = 54$ (size, instance, partition) combinations; one instance ($|V| = 200$, instance 01, *blood_type*) produced no PNE within the time budget, yielding 53 logged runs.

Solver configuration. We use BZR with a 600-second time limit per direction, 16 Gurobi threads, and a 10^{-4} MIP gap target. SW and BSW each receive a 600-second budget. The master MIP for $L_{\max} = 3$ uses standard McCormick lifts for bilinear cycle activations y_c^{pq} and an extended McCormick scheme for trilateral activations y_c^{pqr} .

Aggregated results by graph size. Table 3.17 aggregates results by graph size, showing the number of (instance, partition) runs, the count of trivial cases ($\text{POS} = 1$, i.e., $\hat{\mathbf{x}}_{\text{pne}}^*$ already coincides with SW), the count of non-trivial cases, and headline statistics. The fraction of non-trivial cases grows sharply with $|V|$: from $1/9$ at $|V| = 60$ to $4/8$ at $|V| = 200$. Mean POA exceeds 1.3 at every size, reflecting wide gaps between the best and worst PNEs even when the best PNE already attains SW.

Table 3.17: KEG results aggregated by graph size

$ V $	runs	triv.	non-triv.	POS mean	POS max	POA mean	POA max	PIR mean
60	9	8	1	1.010	1.091	1.643	3.000	1.000
70	9	9	0	1.000	1.000	1.344	1.750	1.000
80	9	9	0	1.000	1.000	1.314	1.546	1.000
90	9	9	0	1.000	1.000	1.246	1.600	1.000
100	9	6	3	1.107	1.708	1.322	1.864	1.000
200	8	4	4	1.064	1.195	1.104	1.195	1.000

“triv.” counts instances with $\text{POS} = 1$ (best PNE already coincides with SW); “non-triv.” counts $\text{POS} > 1$. The $|V| = 200$ row reports 8 runs because one (instance, partition) combination produced no PNE within the time budget.

Non-trivial cases. Table 3.18 enumerates the eight non-trivial cases. These are the instances where the Nash-informed planning framework actively raises welfare above the best discoverable PNE. PIR equals 1.000 in every case, indicating exact SW recovery; recoveries range from 2.6 percentage points ($|V| = 200$, instance 01, uniform) to 70.8 percentage points ($|V| = 100$, instance 02, uniform).

Table 3.18: KEG non-trivial instances ($\text{POS} > 1$)

$ V $	inst	partition	$ C $	$ C_{\text{tri}} $	SW	BPNE	WPNE	BSW	POS	POA
60	01	blood_type	93	16	12	11	9	12	1.091	1.333
100	02	uniform	2106	543	41	24	24	41	1.708	1.708
100	02	blood_type	2106	504	41	40	22	41	1.025	1.864
100	03	geographic	7781	1652	54	44	41	54	1.227	1.317
200	01	uniform	40556	8698	117	114	114	117	1.026	1.026
200	02	uniform	20351	4298	98	88	88	98	1.114	1.114
200	02	geographic	20351	4699	98	83	83	98	1.181	1.181
200	02	blood_type	20351	4074	98	82	82	98	1.195	1.195

$|C|$ denotes the total number of length- ≤ 3 cycles in the partitioned graph; $|C_{\text{tri}}|$ counts trilateral cycles (cycles whose three pairs span three different players). PIR equals 1.000 in every row.

Observations. Three patterns emerge. First, trivial cases dominate at small graph sizes: at $|V| \leq 90$, 35 of 36 runs have $\text{POS} = 1$, meaning the best PNE already attains SW and Nash-informed planning has no welfare to recover. As $|V|$ grows the non-trivial share rises to $1/3$ at $|V| = 100$ and $1/2$ at $|V| = 200$, suggesting that cross-player coordination opportunities emerge with scale. Second, when the framework is needed it delivers exact recovery: every non-trivial case has $\text{PIR} = 1.000$ regardless of the magnitude of POS . Third, POA is consistently larger than POS , often substantially: at $|V| = 100$, the worst PNE attains less than 54% of SW for one (instance, partition) combination ($\text{POA} = 1.864$). The wide POA range underscores the value of selecting among PNEs rather than treating any equilibrium as a planning anchor.

Boundary observations on PNE existence and scale. We additionally attempted KEG at $|V| = 300$ with 3 instances and 3 partitions; in every (instance, partition) attempted, BZR did not certify a PNE within the 600-second budget, and pilot runs at $|V| \in \{400, 500\}$ likewise failed to certify PNEs within that budget. The interpretation of such non-certification cases is discussed in Section 3.6. For the purpose of this chapter we therefore restrict KEG results to $|V| \leq 200$; in every (instance, partition) where BZR did certify a PNE, the framework recovers SW to within numerical precision.

Reference utility construction. Trivial cases ($\text{POS} = 1$) admit $\text{BSW} = \text{SW} = \text{BPNE}$ trivially. Among non-trivial cases, the dominant-PNE scheme is active when the same PNE is individually best for every player; otherwise the per-player arithmetic mean is used. All eight non-trivial KEG cases admitted a feasible BSW under the dominant scheme; no instance required the min-PNE fallback.

Chapter 4

Conditional Equilibrium Inequalities for Generalized Nash Equilibrium Problems with Pure-Integer Variables

Manuscript Note. This chapter is based in part on the following manuscript:

B. Hao, H. Lee, R. Hildebrand, and C. Michini, “Conditional Equilibrium Inequalities for Generalized Nash Equilibrium Problems with Pure-Integer Variables.” Working paper.

The manuscript has been adapted and expanded for inclusion in this dissertation; notation, exposition, and organization have been revised for consistency with the remaining chapters. Code and data are available at <https://github.com/HyunwooLee0429/GIPG>.

4.1 Introduction

4.1.1 Problem Setting

A *generalized integer programming game* (GIPG) is a non-cooperative game in which each of n players solves an integer program whose feasible set depends on the other players’ decisions through coupling constraints. GIPGs sit between two classes: removing the coupling—so

that each player’s feasible set is independent of opponents—recovers an *integer programming game* (IPG) [35, 96], while relaxing the integrality constraints in the players’ optimization problems gives a *generalized Nash equilibrium problem* (GNEP) [85].

GIPGs arise in settings involving shared discrete resources. In *bin packing games*, players split items across bins with proportional cost sharing and shared bin capacities [18, 84]. In *generalized knapsack games*, players select from a shared item set with per-item capacity limits [62]. In *network formation games*, players route integer-splittable flow through a shared network with edge capacities and proportional cost sharing [3, 45]. In each case, shared resource capacities make one player’s feasibility conditional on others’ consumption.

Computing PNE of GIPGs is fundamentally hard. Even for IPGs (without coupling constraints), deciding whether a PNE exists is Σ_2^P -complete [35]; GIPGs are at least as hard. GIPGs add a further difficulty: the *conditional feasibility* of deviations. In an IPG, any privately feasible deviation is always jointly feasible. In a GIPG, a deviation that is privately feasible may be jointly infeasible because opponents have consumed shared resources. An equilibrium-enforcing inequality must therefore be *conditional*: it enforces the no-regret condition only when the deviation is feasible against the current opponents’ strategies.

This chapter develops computational methods for GIPGs centered on *conditional equilibrium inequalities* (CEIs)—globally valid linear inequalities that encode the equilibrium condition with an explicit activation mechanism for conditional feasibility. CEIs are separated via best-response computations and embedded as lazy constraints in a branch-and-cut solver, yielding the *Generalized Zero-Regret* (GZR) algorithm. We apply this framework to four GIPG classes—integer splittable bin packing (ISBP), network formation games (NFG), generalized knapsack games (GKG), and generalized knapsack games with bilinear costs (GKGB)—and complement it with structural results (the Vertex Equilibrium Structure Theorem) and algorithms for approximate equilibria (best-response dynamics and bisection search).

4.1.2 Literature Review

Carvalho et al. [37] provide a comprehensive computational overview of IPGs, covering both PNE and mixed Nash equilibrium (MNE) algorithms. For computing PNEs, Sagratella [128] proposes a branching method that enumerates all solutions of Nash equilibrium problems with discrete strategy sets, and Schwarze and Stein [135] extend this approach to convex polyhedral strategy sets via a branch-and-prune algorithm. Dragotto and Scatamacchia [58] take a different approach with the Zero-Regret (ZR) algorithm, which embeds equilibrium inequalities as lazy constraints in a master MILP. A key feature of ZR is that it *optimizes* over the set of PNEs, enabling direct computation of socially optimal PNE and the price of stability (POS). Our GZR algorithm extends ZR to GIPGs by introducing conditional feasibility into the equilibrium inequalities.

For GNEPs with mixed-integer variables, Harks and Schwarz [85] establish existence results via a convexification technique using the Nikaido–Isoda function: they construct convexified instances and show that equilibria of the convexified game correspond to equilibria of the original. Duguet et al. [62] study GNEPs with mixed-integer variables—which fall within the GIPG class—and propose a branch-and-cut methodology using a high-point relaxation (HPR) of the Nikaido–Isoda function together with intersection cuts. HPR reformulates the GNEP as a bilevel optimization problem where the objective is the Nikaido–Isoda value; an equilibrium corresponds to a zero objective value. The intersection cuts are locally valid within the branching tree and require an explicit geometric construction rather than being expressed in closed form as linear inequalities. This makes HPR fundamentally a *feasibility* approach: it finds a PNE but does not optimize over the set of PNEs with a custom welfare objective. Duguet et al. [61] extend this line to approximate equilibria, introducing an (α, β) -Nash equilibrium concept and developing a binary-search procedure to find the tightest approximate equilibrium, enhanced by a single-tree implementation that reuses

cuts across iterations. Beyond this line, computational work on mixed-integer GNEPs is sparse. Sagratella [129] studies generalized potential games with mixed-integer variables and shows that a Gauss–Seidel best-response scheme approximates equilibria for linear coupling constraints with decoupled costs; Sagratella [130] extends this analysis to GNEPs with linear coupling constraints, and Fabiani and Grammatico [66] apply best-response schemes to multi-vehicle automated driving modeled as a generalized mixed-integer potential game. For games without coupling constraints, two further approaches relate to optimizing over the equilibrium set: Crönert and Minner [49] enumerate all equilibria of large finite games that cannot be efficiently represented in normal form, combining sampling with equilibrium-selection theory to identify a most likely equilibrium, and Carvalho et al. [38] compute (mixed) equilibria via cutting-plane outer approximations of a convexified game. In contrast to sampling-based enumeration, GZR optimizes a welfare objective over the implicitly described set of pure equilibria of a GIPG via globally valid equilibrium inequalities.

Rosen [125] studies concave games—where each player minimizes a concave cost over a polytope—and establishes existence and uniqueness of Nash equilibria. A standard fact of convex analysis is that the minimum of a strictly concave function over a polytope is attained at a vertex. For GIPGs with concave costs, this observation motivates the Vertex Equilibrium Structure Theorem (VEST) developed in this chapter, which characterizes the structure of PNE allocations. However, Rosen’s existence guarantee applies to games with fixed, player-specific strategy sets; for GNEPs with coupled constraints, a standard PNE may fail to exist even in the continuous relaxation.

Integer splittable bin packing games (ISBP) are introduced by Hao and Michini [84] as a weighted, splittable variant of the selfish bin packing games studied earlier by Bilò [18]; they establish structural properties of PNE, including conditions under which exact PNE fail to exist, and show that computing an exact PNE is NP-hard. Generalized knapsack games

(GKG) serve as a benchmark in Duguet et al. [62]. Motivated by the central role of the knapsack problem in combinatorial optimization, classical knapsack games—in which each player has an independent feasible set—have also received considerable attention in equilibrium computation; this line of work was initiated by Carvalho [33], with subsequent equilibrium-computation methods developed in Carvalho et al. [36, 38], Dragotto and Scatamacchia [58]. For network formation games (NFG), we study an integer-splittable variant of the weighted, capacitated network design model of Chassein et al. [42], which falls within the GIPG class; already in their unsplittable model, PNE may fail to exist and it is strongly NP-hard to decide whether a PNE exists. The classical network formation game with Shapley cost sharing [3] considers unsplittable single-unit routing where PNE always exists by a potential argument. The weighed variant of this model is studied by Chen and Roughgarden [45] and Dragotto and Scatamacchia [58] compute exact POS for this model via ZR. Feldman and Ron [68] study the capacitated variant. Congestion games were initiated by Rosenthal [126] with PNE existence via exact potentials; Fotakis et al. [70] study selfish unsplittable flows, and Del Pia et al. [53] establish PNE existence for totally unimodular congestion games. These classical results apply to uncapacitated or unsplittable settings, whereas the ISBP and NFG models studied in this chapter involve integer-splittable strategies with shared capacity constraints, making them GIPGs where PNE existence is not guaranteed.

4.1.3 Contributions

- (i) We introduce CEIs, globally valid linear inequalities for GIPGs that enforce the no-regret condition conditionally on deviation feasibility, and prove that they are exact on integer points. The Generalized Zero-Regret (GZR) algorithm embeds CEIs as lazy constraints in a MILP, extending ZR [58] from IPGs to GIPGs. Because CEIs are globally valid and expressed in closed form, GZR can optimize over the set of PNEs—

unlike the approach of Duguet et al. [62], which relies on locally valid intersection cuts and targets finding any PNE. We also adapt round random-restart best-response dynamics (RRR-BRD) [98] to GIPGs (Section 4.4) and combine it with GZR and a bisection search procedure that reuses CEIs across iterations, analogously to Duguet et al. [61]. Together, these algorithms cover three tasks: RRR-BRD quickly finds a PNE heuristically, GZR computes the socially optimal PNE, and bisection search finds the tightest α -approximate PNE when exact PNE may not exist.

- (ii) For GIPGs with strictly concave costs and integral conditional polytopes, we prove that at any PNE, each player’s strategy lies at a vertex of their conditional strategy polytope; for separable concave costs, a weak form of the theorem restricts feasible directions at a non-vertex PNE strategy to coordinates where the cost is locally affine. On regular ISBP instances—where solo-resource costs are affine, so the vertex property is established instead by a cost-preserving transformation of equilibria (Theorem 4.22)—a VEST-inspired, optimality-preserving cut requires each player to use at most one unsaturated resource; computational experiments show that this cut provides a $1.55\times$ aggregate speedup. On flow polytopes (as in NFG), the resulting bound is weaker but still valid.
- (iii) We demonstrate GZR and the accompanying algorithms on a broad class of GIPGs by developing CEI encodings, activation conditions, and game-specific valid inequalities for ISBP, NFG, GKG, and GKGB. For the cost-sharing games (ISBP and NFG), we show how one-hot encodings linearize nonlinear proportional cost-sharing terms and derive the VEST and weak-VEST cuts for these game classes. For general GIPGs, we find a PNE for 100% of GKG benchmark instances within 10 minutes and prove social optimality for all but one, compared to termination rates of 30–50% in 1 hour for the method of Duguet et al. [62]. We further introduce GKGB, a challenging extension

that integrates the reciprocally bilinear cost structure of Dragotto and Scatamacchia [58] with the shared capacity constraints of Duguet et al. [62], and evaluate GZR on the resulting McCormick-linearized formulations.

4.1.4 Paper Organization

Section 4.2 develops the CEI framework, including feasible slices, activation variables, validity and closure properties, and the GZR algorithm. Section 4.3 presents the VEST theorem and VEST cuts. Section 4.4 describes the complementary solution algorithms: RRR-BRD and bisection search. Section 4.5 details the four game applications: ISBP, NFG, GKG, and GKGB. Section 4.6 reports computational results. Section 4.7 concludes.

4.2 Conditional Equilibrium Inequalities

We begin by introducing notation for GIPG and then develop the CEI framework. Consider a game $\mathcal{G}$ with players indexed by $N = \{1, \dots, n\}$. Each player $i \in N$ chooses an integer strategy $x^i \in \mathbb{Z}^{k_i}$; let x^{-i} denote the strategies of all other players, and let $K := \sum_{i \in N} k_i$ so that a joint strategy profile $x = (x^i)_{i \in N}$ lies in $\mathbb{Z}^K$. The feasible set of player i depends on opponents' strategies:

$$\mathcal{X}_i(x^{-i}) := \{x^i \in \mathbb{Z}^{k_i} : g^i(x^i, x^{-i}) \leq 0\}$$

for a constraint function $g^i : \mathbb{R}^K \rightarrow \mathbb{R}^{m_i}$. The joint feasible region is

$$\mathcal{X} := \{x \in \mathbb{Z}^K : g^i(x^i, x^{-i}) \leq 0 \ \forall i \in N\}.$$

The (*opponent-independent*) *strategy set* of player i is

$$\mathcal{X}_i := \text{proj}_{x^i}(\mathcal{X}) = \{x^i \in \mathbb{Z}^{k_i} : (x^i, x^{-i}) \in \mathcal{X} \text{ for some } x^{-i}\},$$

the set of strategies feasible for player i against at least one opponent profile. Throughout this chapter we make the standing assumption that the coupling constraints are *shared*: whenever $x \in \mathcal{X}$ and $\hat{x}^i \in \mathcal{X}_i(x^{-i})$, the deviated profile $(\hat{x}^i, x^{-i})$ again lies in $\mathcal{X}$. All four game classes studied in this chapter have this structure (it is the same property that keeps best-response dynamics jointly feasible; see Section 4.4). Under this assumption, $\mathcal{X}_i(x^{-i}) \subseteq \mathcal{X}_i$ for every opponents' profile x^{-i} arising from a feasible $x \in \mathcal{X}$, and every cost expression $\pi^i(\hat{x}^i, x^{-i})$ with $\hat{x}^i \in \mathcal{X}_i(x^{-i})$ appearing below is well defined. When $\mathcal{X}_i(x^{-i})$ depends on x^{-i} , the game is a GIPG; when $\mathcal{X}_i(x^{-i}) = \mathcal{X}_i$ is independent of opponents, the game reduces to an IPG.

Each player i has a nonnegative cost function $\pi^i : \mathcal{X} \rightarrow \mathbb{R}_{\geq 0}$ and solves

$$\min_{x^i} \{\pi^i(x^i, x^{-i}) \mid x^i \in \mathcal{X}_i(x^{-i}) \subseteq \mathbb{Z}^{k_i}\}. \quad (4.1)$$

Definition 4.1 (α -approximate PNE). Fix $\alpha \geq 1$. A strategy profile $\hat{x} \in \mathcal{X}$ is an α -approximate pure Nash equilibrium (α -PNE) if for every player $i \in N$,

$$\pi^i(\hat{x}^i, \hat{x}^{-i}) \leq \alpha \cdot \pi^i(x^i, \hat{x}^{-i}) \quad \forall x^i \in \mathcal{X}_i(\hat{x}^{-i}).$$

When $\alpha = 1$, this reduces to an exact PNE.

Each player in Definition 4.1 optimizes over its own conditional set $\mathcal{X}_i(\hat{x}^{-i})$, so the concept is the pure-integer analogue of a *generalized* Nash equilibrium rather than of a *variational* equilibrium, which additionally requires the multipliers on the shared constraints to be common across players. The distinction is immaterial when the coupling constraints are inactive,

but not in general: every variational equilibrium is a generalized Nash equilibrium, and the converse fails. Every equilibrium computed in this chapter—exact or α -approximate—is a generalized pure Nash equilibrium in the sense of Definition 4.1.

We next introduce notation for feasible deviations.

Definition 4.2 (α -approximate best response). Fix $\alpha \geq 1$ and $\hat{x}^{-i}$. A strategy $\hat{x}^i \in \mathcal{X}_i(\hat{x}^{-i})$ is an α -approximate best response (α -BR) against $\hat{x}^{-i}$ if

$$\pi^i(\hat{x}^i, \hat{x}^{-i}) \leq \alpha \cdot \pi^i(x^i, \hat{x}^{-i}) \quad \forall x^i \in \mathcal{X}_i(\hat{x}^{-i}).$$

Let $\mathcal{BR}_i^\alpha(\hat{x}^{-i})$ denote the set of all α -BRs of player i against $\hat{x}^{-i}$.

A fundamental difficulty in generalized equilibrium settings is that feasibility of a candidate deviation depends on opponents' decisions. At an α -PNE profile x , for each $i \in N$, player i 's strategy x^i is an α -BR against x^{-i} :

$$\pi^i(x^i, x^{-i}) \leq \alpha \cdot \pi^i(\hat{x}^i, x^{-i}) \quad \forall \hat{x}^i \in \mathcal{X}_i(x^{-i}). \quad (4.2)$$

To realize (4.2) as a single mixed-integer linear inequality, we first define an indicator that captures whether a candidate strategy is feasible under the current opponents' profile.

Definition 4.3 (Activation variable). For player $i \in N$ and under the opponents' profile x^{-i} , we define the *activation variable*

$$\delta_i(\hat{x}^i, x^{-i}) := \begin{cases} 0 & \text{if } \hat{x}^i \in \mathcal{X}_i(x^{-i}), \\ 1 & \text{otherwise.} \end{cases} \quad (4.3)$$

to indicate whether a given candidate strategy $\hat{x}^i$ of player i is feasible.

By construction, $\delta_i(\hat{x}^i, x^{-i}) = 0$ when the candidate strategy $\hat{x}^i$ is feasible for player i (i.e., $\hat{x}^i \in \mathcal{X}_i(x^{-i})$), and $\delta_i(\hat{x}^i, x^{-i}) = 1$ otherwise. The precise linear encoding of δ_i in terms of the game's constraints is game-dependent (see Section 4.5).

Using this activation variable, for a given $\hat{x}^i$, we define the *conditional equilibrium inequality* (CEI):

$$\pi^i(x^i, x^{-i}) \leq \alpha \cdot \pi^i(\hat{x}^i, x^{-i}) + M \cdot \delta_i(\hat{x}^i, x^{-i}). \quad (4.4)$$

When $\delta_i = 0$ (the deviation is feasible), inequality (4.4) enforces the equilibrium bound $\pi^i(x^i, x^{-i}) \leq \alpha \pi^i(\hat{x}^i, x^{-i})$. When $\delta_i = 1$ (the deviation is infeasible), the right-hand side is increased by M , rendering the inequality inactive. Here $M > 0$ is a constant chosen large enough that the inequality is trivially satisfied whenever $\delta_i = 1$; since costs are nonnegative, any $M \geq \max_{x \in \mathcal{X}} \pi^i(x^i, x^{-i})$ suffices (the maximum exists because $\mathcal{X}$ is finite in all settings considered in this chapter).

Example 4.4 (Conditional equilibrium inequality). Consider a two-player *minimization* game modeling competition for a shared resource of capacity 6. The players' costs and strategy sets are

$$\begin{aligned} \pi^1(x^1, x^2) &= (6 - x^1) \cdot x^2, & x^1 &\in \{0, 1, 2, 3, 4, 5\}, \\ \pi^2(x^2, x^1) &= (6 - x^2) \cdot x^1, & x^2 &\in \{1, 2, 3, 4\}, \end{aligned}$$

subject to the coupling constraint $x^1 + x^2 \leq 6$. Both costs are nonnegative over $\mathcal{X}$, since $0 \leq x^1 \leq 5 < 6$ and $1 \leq x^2 \leq 4 < 6$ for every $(x^1, x^2) \in \mathcal{X}$. The joint feasible region is

$$\mathcal{X} = \{(x^1, x^2) \in \{0, \dots, 5\} \times \{1, 2, 3, 4\} : x^1 + x^2 \leq 6\}.$$

Figure 4.1 depicts $\mathcal{X}$ and the best-response structure used below.

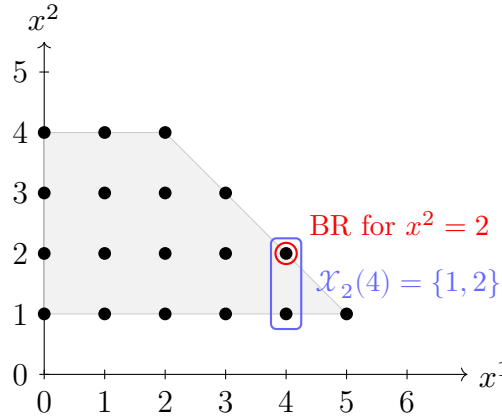


Figure 4.1: The joint feasible region $\mathcal{X}$ of Example 4.4: feasible integer profiles (dots) inside the coupling constraint $x^1 + x^2 \leq 6$ (shaded). Against $x^2 = 2$, player 1's best response is $\hat{x}^1 = 4$ (red circle). The deviation $\hat{x}^1 = 4$ is feasible only against $x^2 \in \mathcal{X}_2(4) = \{1, 2\}$ (blue box), so the CEI generated at $\hat{x}^1 = 4$ enforces the equilibrium bound precisely on these opponents' strategies and is deactivated ($\delta_1 = 1$) otherwise.

Fix $x^2 = 2$. Player 1's feasible strategies are $\{0, 1, 2, 3, 4\}$ with costs $\pi^1(0, 2) = 12$, $\pi^1(1, 2) = 10$, $\pi^1(2, 2) = 8$, $\pi^1(3, 2) = 6$, $\pi^1(4, 2) = 4$, so the best response (minimizer) is $\hat{x}^1 = 4$. The strategies of player 2 against which $\hat{x}^1 = 4$ is feasible are

$$\mathcal{X}_2(4) = \{x^2 \in \{1, 2, 3, 4\} : 4 + x^2 \leq 6\} = \{1, 2\},$$

a proper subset of $\{1, 2, 3, 4\}$. For $\alpha = 1$, the CEI takes the form

$$\pi^1(x^1, x^2) \leq \pi^1(4, x^2) + M \delta_1(4, x^2), \quad \text{i.e.,} \quad (6 - x^1)x^2 \leq 2x^2 + M \delta_1(4, x^2).$$

When $x^2 \in \mathcal{X}_2(4) = \{1, 2\}$, the activation variable $\delta_1(4, x^2) = 0$ and the cut enforces $(6 - x^1)x^2 \leq 2x^2$, equivalently $x^1 \geq 4$ (since $x^2 > 0$). When $x^2 \notin \mathcal{X}_2(4)$, we have $\delta_1 = 1$ and the inequality is relaxed by M . Thus, the CEI enforces $x^1 \geq 4$ *precisely* on those opponents' strategies that make the deviation $\hat{x}^1 = 4$ feasible.

4.2.1 Lifted Formulation

We assume that each player's objective and constraints admit a mixed-integer linear representation after introducing auxiliary variables. Accordingly, we collect all auxiliary variables into a vector w , a subvector of which may be required to take integer values: we write $\mathcal{W} := \mathbb{Z}^p \times \mathbb{R}^q$ for the auxiliary domain (in the one-hot formulations of Section 4.5, for instance, the encoding variables are binary). We assume the existence of a rational polyhedron

$$\mathcal{H} \subseteq \mathbb{R}^D \quad \text{in variables } (x, w),$$

such that for any $x \in \mathcal{X}$ there exists $w \in \mathcal{W}$ with $(x, w) \in \mathcal{H}$, and for any $(x, w) \in \mathcal{H}$ with $x \in \mathbb{Z}^K$ and $w \in \mathcal{W}$, the projection x lies in $\mathcal{X}$. Together with the shared-constraint assumption, the activation variable (4.3) then admits an equivalent lifted characterization: for any candidate strategy $\hat{x}^i$ and opponents' profile x^{-i} ,

$$\delta_i(\hat{x}^i, x^{-i}) = 0 \quad \Longleftrightarrow \quad \exists \hat{w} \in \mathcal{W} : (\hat{x}^i, x^{-i}, \hat{w}) \in \mathcal{H}. \quad (4.5)$$

Moreover, for each player $i \in N$, we assume there exists an affine function $\tilde{\pi}^i : \mathbb{R}^D \rightarrow \mathbb{R}$ such that

$$\tilde{\pi}^i(x, w) = \pi^i(x^i, x^{-i}) \quad \text{for all } (x, w) \in \mathcal{H} \text{ with } x \in \mathbb{Z}^K, w \in \mathcal{W}.$$

We call $\tilde{\pi}^i$ the *lifted cost function* of player i : an affine function that agrees with π^i on the integer points of $\mathcal{H}$.

The *lifted equilibrium set* is

$$\mathcal{E}^\alpha(\mathcal{G}) := \{(x, w) \in \mathcal{H} \cap (\mathbb{Z}^K \times \mathcal{W}) : x \text{ is an } \alpha\text{-PNE}\}.$$

Example 4.4 (Conditional equilibrium inequality, continued). We continue Example 4.4. Player 1's cost $\pi^1(x^1, x^2) = (6 - x^1) \cdot x^2$ is bilinear and hence not affine in (x^1, x^2) directly. To construct an affine lifted cost $\tilde{\pi}^1$, we introduce auxiliary variables using a *unary (one-hot) encoding* of x^2 .

For each $k \in \{1, 2, 3, 4\}$, introduce a binary variable $\lambda_k \in \{0, 1\}$ representing the event $\{x^2 = k\}$. The encoding is defined by the constraints

$$\sum_{k=1}^4 \lambda_k = 1, \quad x^2 = \sum_{k=1}^4 k \lambda_k.$$

Since $\pi^1 = (6 - x^1) \cdot x^2 = \sum_{k=1}^4 k(6 - x^1)\lambda_k$, we need to linearize the product $x^1\lambda_k$. For each k , introduce an auxiliary variable $w_k \geq 0$ representing $x^1\lambda_k$, enforced by

$$w_k \leq 5\lambda_k, \quad x^1 - 5(1 - \lambda_k) \leq w_k \leq x^1.$$

When $\lambda_k = 1$ these constraints force $w_k = x^1$; when $\lambda_k = 0$ they force $w_k = 0$. The lifted cost function is then

$$\tilde{\pi}^1(x^1, x^2, \lambda, w) := \sum_{k=1}^4 k(6\lambda_k - w_k),$$

which is affine in (x^1, x^2, λ, w) . One can verify that for any integer point $(x^1, x^2) \in \mathcal{X}$, exactly one λ_k equals 1 (with $k = x^2$) and $w_k = x^1$, so $\tilde{\pi}^1 = (6 - x^1) \cdot x^2 = \pi^1(x^1, x^2)$ as required.

The polyhedron $\mathcal{H}$ is defined in variables (x^1, x^2, λ, w) by the coupling constraint $x^1 + x^2 \leq 6$, the bounds $x^1 \in [0, 5]$, $x^2 \in [1, 4]$, $\lambda_k \in [0, 1]$, $w_k \geq 0$, and the encoding and linearization constraints above. A symmetric construction yields $\tilde{\pi}^2$ for player 2 by encoding x^1 instead.

4.2.2 CEI Validity and Closure

We now collect the main properties of the CEI system. We require one structural assumption on the lifted formulation.

Assumption 4.5 (CEI computability). *For every player $i \in N$ and every candidate deviation $\hat{x}^i \in \mathcal{X}_i$ there exists an affine function $\tilde{\pi}_{\hat{x}^i}^i : \mathbb{R}^D \rightarrow \mathbb{R}$, the lifted deviation cost, such that for every $(x, w) \in \mathcal{H}$ with $x \in \mathbb{Z}^K$:*

$$(i) \quad \tilde{\pi}_{\hat{x}^i}^i(x, w) = \pi^i(\hat{x}^i, x^{-i}) \text{ whenever } \hat{x}^i \in \mathcal{X}_i(x^{-i});$$

$$(ii) \quad \tilde{\pi}_{\hat{x}^i}^i(x, w) \geq 0.$$

For brevity, we write $\tilde{\pi}^i(\hat{x}^i, x^{-i}, w) := \tilde{\pi}_{\hat{x}^i}^i(x, w)$.

In the simplest case, $\tilde{\pi}_{\hat{x}^i}^i$ is obtained by literally substituting $\hat{x}^i$ into $\tilde{\pi}^i$; this happens when $\tilde{\pi}^i$ is affine in the player's own variables, as in GKG. In general, however, $\tilde{\pi}^i$ may depend on x^i only through the auxiliary variables w (as in the one-hot encodings for ISBP and NFG), and $\tilde{\pi}_{\hat{x}^i}^i$ is then a separate affine function whose coefficients are computed from $\hat{x}^i$ at separation time, using the joint information about x^{-i} carried by w ; see (4.11), (4.37), and (4.42). All four game classes in this chapter satisfy the assumption by construction; see Section 4.5.

We emphasize that the assumption concerns only the *cut*: the separation check in Algorithm 4.8 evaluates the deviation cost by re-lifting, i.e., at a point $(\hat{x}^i, x^{-i}, \hat{w}) \in \mathcal{H}$, which requires no assumption since $\tilde{\pi}^i$ agrees with π^i on the integer points of $\mathcal{H}$. Assumption 4.5 is what allows the same quantity to be written as a linear function of the *master* variables, so that the resulting CEI is globally valid throughout the branch-and-cut tree.

For any player $i \in N$ and candidate strategy $\hat{x}^i \in \mathcal{X}_i$, the *lifted CEI* is

$$\tilde{\pi}^i(x, w) \leq \alpha \cdot \tilde{\pi}^i(\hat{x}^i, x^{-i}, w) + M \cdot \delta_i(\hat{x}^i, x^{-i}), \quad (4.6)$$

where M is the same constant as in (4.4). Since $\tilde{\pi}^i$ and $\tilde{\pi}_{\hat{x}^i}^i$ are affine and $\hat{x}^i$ is a fixed parameter, (4.6) is a linear inequality in the master variables (x, w) . Let $\mathcal{E}_{\text{cei}}^\alpha$ be the set of points $(x, w) \in \mathcal{H}$ with $x \in \mathbb{Z}^K$ satisfying all lifted CEIs (4.6) over all i and $\hat{x}^i \in \mathcal{X}_i$, and let $P_{\text{cei}}^\alpha := \text{conv}(\mathcal{E}_{\text{cei}}^\alpha)$ denote the *CEI closure*.

Proposition 4.6 (CEI exactness and closure). *Fix $\alpha \geq 1$ and suppose each $\mathcal{X}_i$ is finite.*

- (i) (Exactness.) $\mathcal{E}_{\text{cei}}^\alpha = \mathcal{E}^\alpha(\mathcal{G})$: *the integer points of $\mathcal{H}$ satisfying every lifted CEI (4.6) are exactly the lifted α -PNE profiles.*
- (ii) (Closure.) $P_{\text{cei}}^\alpha = \text{conv}(\mathcal{E}^\alpha(\mathcal{G}))$ *is a rational polytope admitting a finite inequality description. Every integer point of P_{cei}^α is an α -PNE, and in particular every vertex of P_{cei}^α lies in $\mathcal{E}^\alpha(\mathcal{G})$; consequently, optimizing any linear objective F over P_{cei}^α attains its optimum at an α -PNE.*

Proof. (i). For $\mathcal{E}^\alpha(\mathcal{G}) \subseteq \mathcal{E}_{\text{cei}}^\alpha$, let $(x, w) \in \mathcal{E}^\alpha(\mathcal{G})$, so $x \in \mathbb{Z}^K$ is an α -PNE and $(x, w) \in \mathcal{H}$. Fix i and $\hat{x}^i \in \mathcal{X}_i$. If $\delta_i(\hat{x}^i, x^{-i}) = 0$ then $\hat{x}^i \in \mathcal{X}_i(x^{-i})$, and the α -PNE condition with Assumption 4.5 gives

$$\tilde{\pi}^i(x, w) = \pi^i(x^i, x^{-i}) \leq \alpha \pi^i(\hat{x}^i, x^{-i}) = \alpha \tilde{\pi}^i(\hat{x}^i, x^{-i}, w),$$

so (4.6) holds. If $\delta_i(\hat{x}^i, x^{-i}) = 1$, it holds since $M \geq \max_{x \in \mathcal{X}} \pi^i(x^i, x^{-i})$ and $\alpha \tilde{\pi}^i(\hat{x}^i, x^{-i}, w) \geq 0$ by Assumption 4.5(ii). For the reverse inclusion, suppose $(x, w) \in \mathcal{E}_{\text{cei}}^\alpha$ but x is not an α -PNE. Then some player i has $\hat{x}^i \in \mathcal{X}_i(x^{-i})$ with $\pi^i(x^i, x^{-i}) > \alpha \pi^i(\hat{x}^i, x^{-i})$. Since

$\hat{x}^i \in \mathcal{X}_i(x^{-i})$ we have $\delta_i(\hat{x}^i, x^{-i}) = 0$, so by Assumption 4.5 the CEI (4.6) generated by $\hat{x}^i$ reads $\tilde{\pi}^i(x, w) = \pi^i(x^i, x^{-i}) > \alpha \tilde{\pi}^i(\hat{x}^i, x^{-i}, w)$ and is violated, contradicting $(x, w) \in \mathcal{E}_{\text{cei}}^\alpha$. Hence $\mathcal{E}_{\text{cei}}^\alpha = \mathcal{E}^\alpha(\mathcal{G})$.

(ii). Finiteness of each $\mathcal{X}_i$ makes $\mathcal{X}$ finite; since the lifted auxiliaries are determined by x at integer points (as holds in every formulation of Section 4.5), $\mathcal{E}^\alpha(\mathcal{G})$ is a finite set of rational points. (Determination is convenient but not essential: if each fiber $\{w : (x, w) \in \mathcal{H}\}$ is merely a polytope, then $\mathcal{E}^\alpha(\mathcal{G})$ is a finite union of polytopes rather than a finite point set, and its convex hull is still a rational polytope whose vertices are vertices of individual fibers, hence lifted α -PNE profiles; the conclusion of part (ii) is unchanged.) By (i), $P_{\text{cei}}^\alpha = \text{conv}(\mathcal{E}_{\text{cei}}^\alpha) = \text{conv}(\mathcal{E}^\alpha(\mathcal{G}))$ is a rational polytope by the Minkowski–Weyl theorem and admits a finite inequality description. Each lifted CEI is valid for $\mathcal{E}^\alpha(\mathcal{G})$ and, being linear, for its convex hull P_{cei}^α . Thus any integer point $(x, w) \in P_{\text{cei}}^\alpha$ lies in $\mathcal{H}$, is integer, and satisfies every CEI, hence $(x, w) \in \mathcal{E}_{\text{cei}}^\alpha = \mathcal{E}^\alpha(\mathcal{G})$ by (i); in particular every vertex is an α -PNE. A linear F attains its optimum over a polytope at a vertex, which is therefore an α -PNE. $\square$

Remark 4.7 (Relation to Dragotto and Scatamacchia; perfect equilibrium formulation). Proposition 4.6 extends Theorem 1 of Dragotto and Scatamacchia [58] from IPGs to GIPGs: their equilibrium inequalities become the conditional CEIs, with the activation term $M\delta_i$ accounting for deviations rendered infeasible by the coupling constraints. In their terminology P_{cei}^α is the perfect equilibrium formulation, and part (ii) states that optimizing a linear welfare over it returns an α -PNE.

The boundary structure carries over as well. For a profile x , set $m_i(x^{-i}) := \min_{x'^i \in \mathcal{X}_i(x^{-i})} \pi^i(x'^i, x^{-i})$ and $\alpha^*(x) := \max_i \pi^i(x^i, x^{-i})/m_i(x^{-i})$ (with $\alpha^*(x) = \infty$ if some $m_i = 0 < \pi^i$, and $0/0 := 1$ when $\pi^i = m_i = 0$). Then x is an α -PNE iff $\alpha \geq \alpha^*(x)$. At $\alpha = \alpha^*(x)$ the best-response CEI of every worst-off player is active: for such a player the lifted CEI generated by a best response $\hat{x}^i$ reads $\tilde{\pi}^i(x, w) = \pi^i(x^i, x^{-i}) = \alpha^*(x) \pi^i(\hat{x}^i, x^{-i})$; for $\alpha > \alpha^*(x)$ all of x 's best-

response CEIs are slack (provided $m_i > 0$ for every player; a player with $m_i = \pi^i = 0$ has a CEI that is tight at every level). In particular, at an exact PNE ($\alpha = 1$, where x^i is itself a best response, so $\alpha^*(x) = 1$) every best-response CEI is tight, and every PNE lies on the boundary of P_{cei}^1 unless that CEI is an implicit equality of the formulation. This $\alpha^*(x)$ is the per-profile ratio that RRR-BRD reports and that α -bisection searches for. None of this is used by GZR (Proposition 4.9), whose correctness rests only on exactness (i) and on optimizing to a vertex (ii).

4.2.3 Generalized Zero-Regret Algorithm

Consider a global objective function F over strategy profiles. Our goal is to find an α -PNE that is optimal with respect to F , or to certify that none exists. For instance, minimizing the social cost $F(x, w) = \sum_i \tilde{\pi}^i(x, w)$ over all α -PNEs yields a solution achieving the Price of Stability of the game.

We propose the *Generalized Zero-Regret* (GZR) algorithm, which finds an optimal α -PNE by solving a sequence of MILPs over $\mathcal{H}$, progressively augmented with CEIs and possibly additional game-specific valid inequalities. GZR extends the Zero-Regret algorithm of Dragotto and Scatamacchia [58] in two directions: it handles α -approximate PNEs for any $\alpha \geq 1$, and it applies to GIPGs with coupling constraints, beyond the IPG setting of the original algorithm.

Algorithm 4.8 implements GZR within a lazy-constraint MILP framework. During branch-and-cut, whenever the solver finds an incumbent mixed-integer solution $(\bar{x}, \bar{w})$, we run lifted CEI separation: for each player i we solve a best-response MILP over $\mathcal{H}$ to obtain $(\hat{x}^i, \hat{w})$, and if $\tilde{\pi}^i(\bar{x}^i, \bar{x}^{-i}, \bar{w}) > \alpha \tilde{\pi}^i(\hat{x}^i, \bar{x}^{-i}, \hat{w})$, we add the violated lifted CEI as a lazy constraint. If no violated lifted CEI is found, the incumbent $\bar{x}$ is an α -PNE; the solver returns the best

Algorithm 4.8 Generalized Zero-Regret (GZR)

Require: Game $\mathcal{G}$, lifted formulation $\mathcal{H}$, objective $F(x, w)$, $\alpha \geq 1$, Big- $M > 0$, cut pool Φ_0

Ensure: $(x^*, \text{FOUND}, \Phi)$ or $(\text{null}, \text{INFEASIBLE}, \Phi)$

```

1:  $\Phi \leftarrow \Phi_0$ 
2: while true do
3:   Solve  $\mathcal{Q}(\Phi) := \min\{F(x, w) : (x, w) \in \mathcal{H} \cap (\mathbb{Z}^K \times \mathcal{W}), \Phi\}$ 
4:   if  $\mathcal{Q}(\Phi)$  is infeasible return  $(\text{null}, \text{INFEASIBLE}, \Phi)$ 
5:   Let  $(\bar{x}, \bar{w})$  be optimal;  $\varphi \leftarrow \emptyset$ 
6:   for  $i \in N$  do
7:      $(\hat{x}^i, \hat{w}) \leftarrow \arg \min\{\tilde{\pi}^i(x^i, \bar{x}^{-i}, w) : (x^i, \bar{x}^{-i}, w) \in \mathcal{H}, x^i \in \mathbb{Z}^{k_i}, w \in \mathcal{W}\}$ 
8:     if  $\tilde{\pi}^i(\bar{x}, \bar{w}) > \alpha \tilde{\pi}^i(\hat{x}^i, \bar{x}^{-i}, \hat{w})$  then add CEI for  $(\hat{x}^i, \hat{w})$  to  $\varphi$ 
9:
10:  if  $\varphi = \emptyset$  return  $(\bar{x}, \text{FOUND}, \Phi)$  else  $\Phi \leftarrow \Phi \cup \varphi$ 
11:
```

such incumbent with respect to F .

Proposition 4.9 (GZR correctness). *Suppose each $\mathcal{X}_i$ is finite and ?? 4.5 holds, and let Algorithm 4.8 run to termination (that is, without an imposed time limit). If Algorithm 4.8 returns x^* , then x^* is an α -PNE. If it returns “no α -PNE,” then no α -PNE exists in $\mathcal{X}$.*

Proof. If x^* is returned, then no lifted CEI is violated at $(x^*, \bar{w})$: for each player i , the best response $(\hat{x}^i, \hat{w})$ against $x^{*, -i}$ satisfies $\tilde{\pi}^i(x^{*, i}, x^{*, -i}, \bar{w}) \leq \alpha \tilde{\pi}^i(\hat{x}^i, x^{*, -i}, \hat{w})$, so x^* is an α -PNE. If the MILP becomes infeasible, then by Proposition 4.6(i), every α -PNE satisfies all lifted CEIs; since the MILP with the accumulated CEIs — a subset of all lifted CEIs — is already infeasible, no α -PNE exists. $\square$

4.3 VEST Inequalities

The GZR algorithm can be strengthened with additional valid inequalities when the game's cost functions and feasible sets have special structure. For a fixed opponents' profile x^{-i} , let

$$P^i(x^{-i}) := \{x^i \in \mathbb{R}^{k_i} : A^i x^i = b^i(x^{-i}), 0 \leq x_j^i \leq u_{ij}(x^{-i}) \forall j\}$$

denote the *conditional polytope* of player i , i.e., a polytope whose integer points are precisely the strategies of player i : $\mathcal{X}_i(x^{-i}) = P^i(x^{-i}) \cap \mathbb{Z}^{k_i}$. We write $P^i = P^i(x^{-i})$ when the dependence on x^{-i} is clear. The *Vertex Equilibrium Structure Theorem* (VEST) shows that under some structural assumptions, every PNE strategy must be a vertex of the player's feasible polytope, yielding *VEST cuts* that can be added to the GZR master problem. We develop VEST as a single face-constancy theorem (Theorem 4.14) with two corollaries: strict concavity forces vertex strategies (Corollary 4.16), and separability localizes the conclusion to coordinates where the cost is not locally affine (Corollary 4.17).

The intuition behind VEST is economic rather than merely geometric. Under proportional cost sharing, a player's cost on a resource is concave in its own usage of that resource, so the average cost of a unit falls as the player concentrates load: splitting a demand across two partially used resources is weakly dominated by shifting all of it onto one of them. Equilibrium therefore pushes each player toward an extreme point of its conditional polytope, and the VEST cuts below impose exactly that structural consequence on the GZR master problem.

We first record a face-constancy property of PNE strategies that holds under mere concavity of the costs; strict concavity and separability then yield the two corollaries used in our applications.

Assumption 4.10 (Concave Costs). *For each player $i \in N$ and each opponents' profile x^{-i} , the cost function $\pi^i(\cdot, x^{-i})$ is concave on $P^i(x^{-i})$.*

Assumption 4.11 (Integer Vertices). *For each player $i \in N$ and each opponents' profile x^{-i} , every vertex of $P^i(x^{-i})$ is an integer point.*

Remark 4.12. Assumption 4.11 holds whenever the constraint matrix A^i is totally unimodular and $b^i(x^{-i})$, $u_{ij}(x^{-i})$ are integer-valued for every x^{-i} . This is the case for the two cost-sharing games considered in this chapter: in ISBP $A^i = \mathbf{1}^\top$, $w_i \in \mathbb{Z}$, and the bin capacities are integer; in NFG the constraint matrices are the node-edge incidence matrices, and weights and capacities are integral.

We call a nonzero $d \in \mathbb{R}^{k_i}$ a *feasible direction* at x^i within a face F of P^i if $x^i + \theta d \in F$ for some $\theta > 0$. Explicitly, when F is the minimal face of P^i containing x^i , d is a feasible direction within F if and only if $A^i d = 0$, $d_j = 0$ for all $j \notin \text{free}(x^i)$, and d_j is unrestricted for $j \in \text{free}(x^i)$, where the *free variables* of x^i are:

$$\text{free}(x^i) = \{j : 0 < x_j^i < u_{ij}(x^{-i})\}.$$

A strategy x^i is a vertex of P^i if and only if there is no feasible direction within the minimal face of P^i containing x^i .

Lemma 4.13 (Vertices of a bounded simplex). *Suppose $A^i = \mathbf{1}^\top$, so that $P^i(x^{-i})$ is a bounded simplex (as in ISBP). Then $x^i \in P^i(x^{-i})$ is a vertex of $P^i(x^{-i})$ if and only if $|\text{free}(x^i)| \leq 1$.*

Proof. If $j, l \in \text{free}(x^i)$ with $j \neq l$, then $d = e_j - e_l$ satisfies $\mathbf{1}^\top d = 0$ and $d_r = 0$ for $r \notin \text{free}(x^i)$, so d is a feasible direction within the minimal face containing x^i and x^i is not a

vertex. Conversely, if $|\text{free}(x^i)| \leq 1$, any feasible direction d within the minimal face satisfies $d_r = 0$ for $r \notin \text{free}(x^i)$ and $\mathbf{1}^\top d = 0$; the single (or empty) remaining coordinate is then forced to zero, so $d = 0$ and x^i is a vertex. $\square$

Theorem 4.14 (Weak VEST). *Under Assumptions 4.10 and 4.11, if x is a PNE and x^i is not a vertex of $P^i(x^{-i})$ for some player i , then $\pi^i(\cdot, x^{-i})$ is constant on the minimal face F of $P^i(x^{-i})$ containing x^i .*

Proof. Since x^i is not a vertex, there exists a nonzero feasible direction within the minimal face F of P^i containing x^i , so $\dim(F) \geq 1$. Since x is a PNE, x^i minimizes $\pi^i(\cdot, x^{-i})$ over the integer points of P^i . We first upgrade this to minimality over all of P^i . Let $y \in P^i$ and write $y = \sum_k \lambda_k v_k$ as a convex combination of vertices v_k of P^i . By Assumption 4.10,

$$\pi^i(y, x^{-i}) \geq \sum_k \lambda_k \pi^i(v_k, x^{-i}) \geq \min_k \pi^i(v_k, x^{-i}) = \pi^i(v^*, x^{-i}),$$

where v^* attains the minimum over the (finitely many) vertices. By Assumption 4.11, v^* is an integer point of $P^i(x^{-i})$ and hence a feasible deviation for player i , so the PNE condition gives $\pi^i(x^i, x^{-i}) \leq \pi^i(v^*, x^{-i}) \leq \pi^i(y, x^{-i})$. Since $y \in P^i$ was arbitrary, x^i minimizes $\pi^i(\cdot, x^{-i})$ over P^i ; let $f^* = \pi^i(x^i, x^{-i})$ denote the minimum value. We claim $\pi^i(\cdot, x^{-i})$ is constant on F . Let $y \in F$. Since $x^i \in \text{ri}(F)$, there exist $z \in F$ and $\lambda \in (0, 1)$ such that $x^i = \lambda y + (1 - \lambda)z$ [124, Theorem 6.4]. By concavity of $\pi^i(\cdot, x^{-i})$:

$$f^* = \pi^i(x^i, x^{-i}) \geq \lambda \pi^i(y, x^{-i}) + (1 - \lambda) \pi^i(z, x^{-i}) \geq \lambda f^* + (1 - \lambda) f^* = f^*,$$

where the second inequality uses $\pi^i(y, x^{-i}) \geq f^*$ and $\pi^i(z, x^{-i}) \geq f^*$ (since $y, z \in P^i$ and x^i is a minimizer). Equality must hold throughout, forcing $\pi^i(y, x^{-i}) = f^*$. Since $y \in F$ was arbitrary, $\pi^i(\cdot, x^{-i})$ is constant on F . $\square$

Strengthening Assumption 4.10 to strict concavity pins every PNE strategy to a vertex.

Assumption 4.15 (Strictly Concave Costs). *For each player $i \in N$ and each opponents' profile x^{-i} , the cost function $\pi^i(\cdot, x^{-i})$ is strictly concave on $P^i(x^{-i})$.*

Corollary 4.16 (Strong VEST). *Under Assumptions 4.15 and 4.11, if x is a PNE then x^i is a vertex of $P^i(x^{-i})$ for every $i \in N$.*

Proof. Suppose x^i is not a vertex of $P^i(x^{-i})$. By Theorem 4.14, $\pi^i(\cdot, x^{-i})$ is constant on the minimal face F of $P^i(x^{-i})$ containing x^i , and $\dim(F) \geq 1$. A strictly concave function cannot be constant on a segment, contradicting Assumption 4.15. Hence $\dim(F) = 0$, i.e., x^i is a vertex. $\square$

When $\pi^i(x^i, x^{-i}) = \sum_j f_j^i(x_j^i; x^{-i})$ is separable, Theorem 4.14 can be sharpened. For $x^i \in P^i$ let $S = \text{free}(x^i)$, and partition S by the local behavior of each f_j^i at x_j^i :

$$S_0^i(x^i) = \{ j \in S : f_j^i(\cdot; x^{-i}) \text{ is affine on a neighborhood of } x_j^i \text{ in } [0, u_{ij}(x^{-i})] \}, \quad S_+^i(x^i) = S \setminus S_0^i(x^i).$$

Since every $j \in S$ satisfies $0 < x_j^i < u_{ij}(x^{-i})$, the point x_j^i is interior to $[0, u_{ij}(x^{-i})]$ and these neighborhoods are well defined. If f_j^i is twice differentiable at x_j^i , then $S_0^i(x^i)$ and $S_+^i(x^i)$ are the coordinates with $f_j^{i''}(x_j^i) = 0$ and $f_j^{i''}(x_j^i) < 0$, respectively.

Corollary 4.17 (Weak VEST for separable functions). *Suppose $\pi^i(x^i, x^{-i}) = \sum_j f_j^i(x_j^i; x^{-i})$ for all i , where each $f_j^i(\cdot; x^{-i})$ is concave. Under Assumption 4.11, if x is a*

PNE and x^i is not a vertex of $P^i(x^{-i})$, then every feasible direction d at x^i within the minimal face F of $P^i(x^{-i})$ containing x^i satisfies $d_j = 0$ for all $j \in S_+^i(x^i)$: the only coordinates that can vary within F are those in $S_0^i(x^i)$.

Proof. Suppose x^i is not a vertex of P^i , and let d be a feasible direction at x^i within the minimal face F . Since $x^i \in \text{ri}(F)$, there is $\varepsilon > 0$ with $x^i + td \in F$ for all $t \in (-\varepsilon, \varepsilon)$. By Theorem 4.14, $\pi^i(\cdot; x^{-i})$ is constant on F , so

$$g(t) := \pi^i(x^i + td; x^{-i}) = \sum_{j \in S} f_j^i(x_j^i + td_j; x^{-i})$$

is constant on $(-\varepsilon, \varepsilon)$, using separability and $d_j = 0$ for $j \notin S$.

For $j \in S$ put $h_j(t) := f_j^i(x_j^i + td_j; x^{-i})$. Each h_j is concave on $(-\varepsilon, \varepsilon)$ and $\sum_{j \in S} h_j \equiv g$ is constant. A constant sum of concave functions has affine summands [124, §7]: for any $a, b \in (-\varepsilon, \varepsilon)$ and $\lambda \in (0, 1)$,

$$\sum_j h_j(\lambda a + (1 - \lambda)b) \geq \lambda \sum_j h_j(a) + (1 - \lambda) \sum_j h_j(b),$$

and both sides equal the same constant, so every term inequality is tight; hence each h_j is affine on $[a, b]$, and as a, b were arbitrary, on $(-\varepsilon, \varepsilon)$.

Fix $j \in S_+^i(x^i)$ and suppose $d_j \neq 0$. The map $t \mapsto x_j^i + td_j$ is an affine bijection of $(-\varepsilon, \varepsilon)$ onto a neighborhood of x_j^i , so affinity of h_j is equivalent to affinity of $f_j^i(\cdot; x^{-i})$ on that neighborhood, contradicting $j \in S_+^i(x^i)$. Hence $d_j = 0$ for every $j \in S_+^i(x^i)$. $\square$

In cost-sharing games f_j^i is strictly concave whenever resource j is shared, so Corollary 4.17 forces every feasible direction within the minimal face to have zero component along each shared free resource; we exploit this in Sections 4.5.1.3 and 4.5.2.4 for ISBP and NFG.

We close this section with a quantitative consequence of Corollary 4.16 that measures how restrictive the vertex condition is for a given game class.

Corollary 4.18 (Free-coordinate bound). *Under Assumptions 4.15 and 4.11, suppose in addition that the constraint matrix A^i of $P^i(x^{-i})$ has rank p_i . Then at any PNE x , every player satisfies $|\text{free}(x^i)| \leq p_i$.*

Proof. By Corollary 4.16, x^i is a vertex of $P^i(x^{-i}) \subseteq \mathbb{R}^{k_i}$. A vertex requires k_i linearly independent active constraints; the equalities $A^i x^i = b^i(x^{-i})$ supply p_i of them, and each non-free coordinate supplies one active bound, so $p_i + (k_i - |\text{free}(x^i)|) \geq k_i$, i.e., $|\text{free}(x^i)| \leq p_i$. $\square$

The strength of this bound depends on the geometry of $P^i(x^{-i})$: for a bounded simplex ($A^i = \mathbf{1}^\top$, $p_i = 1$, as in ISBP) each player uses at most one unsaturated resource, while for a flow polytope (A^i a node–edge incidence matrix, $p_i = |V| - 1$, as in NFG) up to $|V| - 1$ edges may remain unsaturated, a considerably weaker restriction.

4.4 Complementary Solution Algorithms

4.4.1 Random-Restart Best-Response Dynamics for GNEPs

Lee et al. [98] develop *round random-restart BRD* (RRR-BRD) for IPGs, establishing Las Vegas-style convergence guarantees via a Markov chain on the best-response state graph. We adapt their framework to GIPGs.

Joint feasibility under BR updates. For RRR-BRD to be well-defined, the state space must be closed under best-response updates: if the current profile x is jointly feasible and player i updates to a best response $\hat{x}^i \in \mathcal{X}_i(x^{-i})$, the resulting profile $(\hat{x}^i, x^{-i})$ must remain in $\mathcal{X}$. In an IPG this is immediate since $\mathcal{X} = \prod_{i \in N} \mathcal{X}_i$ and each $\mathcal{X}_i$ is independent. In a GIPG the situation is more subtle because the constraint sets $\mathcal{X}_i(x^{-i})$ are coupled. However,

Algorithm 4.19 RRR-BRD for GIPGs

Require: GIPG instance with $\mathcal{X}$ closed under BR updates, max rounds R , max restarts L

Ensure: PNE x or NONE

```

1: for  $l = 1, \dots, L$  do
2:    $x \leftarrow \text{RANDOMFEASIBLE}(\mathcal{X})$ 
3:   for  $r = 1, \dots, R$  do
4:      $x^{\text{prev}} \leftarrow x$ ; draw random permutation  $\sigma$  of  $N$ 
5:     for  $i$  in order  $\sigma$  do
6:        $\hat{x}^i \leftarrow \arg \min \{ \pi^i(x^i, x^{-i}) : x^i \in \mathcal{X}_i(x^{-i}) \}$ ; if  $\pi^i(\hat{x}^i, x^{-i}) < \pi^i(x^i, x^{-i})$  then
7:          $x^i \leftarrow \hat{x}^i$ 
8:     if  $x = x^{\text{prev}}$  return  $x$   $\triangleright$  PNE certified
9:
10:
11: return NONE

```

when the coupling takes the form of *shared constraints*—i.e., the same capacity or resource constraints appear in every player’s feasible set—a BR update by player i that satisfies the shared constraints from i ’s perspective automatically preserves them for all other players. Concretely, if the shared constraint is $\sum_{k \in N} h_k(x^k) \leq b$ and $\hat{x}^i \in \mathcal{X}_i(x^{-i})$ satisfies $h_i(\hat{x}^i) + \sum_{k \neq i} h_k(x^k) \leq b$, then the constraint remains satisfied for every player $j \neq i$ whose strategy x^j is unchanged. All four GIPGs in this chapter (ISBP, NFG, GKG, GKGB) have this shared-constraint structure, so $\mathcal{X}$ is closed under BR updates.

This property does not hold for all GNEPs. Harks and Schwarz [85] observe that best-response algorithms can fail in congestion-based GNEPs when the algorithm operates on profiles outside the joint feasible set $\mathcal{X}$ —for instance, when initialized from individually feasible but jointly infeasible profiles, opponents’ strategies may consume shared capacity so that a player’s best-response problem has no feasible solution. RRR-BRD avoids this issue by initializing from $\mathcal{X}$ (Line 1) and maintaining joint feasibility throughout.

Algorithm 4.19 presents the procedure: each round visits players in random order, applying improving moves; if no player moves, the profile is a certified PNE; after R rounds without convergence, the algorithm restarts from a fresh jointly feasible profile.

Convergence guarantee. Lee et al. [98, Theorem 1] show that if a PNE exists and the restart law μ satisfies $q_\mu[R] > 0$ —where $q_\mu[R]$ is the probability that a single capped attempt finds a PNE—then RRR-BRD converges with probability 1 and uses at most $nR/q_\mu[R]$ BR solves in expectation. Their proof requires only that the state space is finite and closed under BR updates, so it extends directly to shared-constraint GIPGs. A sufficient condition for $q_\mu[R] > 0$ is that μ has *full support* on $\mathcal{X}$; in IPGs this is easy because $\mathcal{X} = \prod_{i \in N} \mathcal{X}_i$ decomposes across players, but in GIPGs the coupling constraints make full-support sampling over $\mathcal{X}$ computationally expensive in general. In our implementation, we use game-specific heuristics to generate feasible restart profiles, so the theoretical guarantee may not apply; nevertheless, RRR-BRD performs well in practice across all four games.

4.4.2 α -Bisection Search

When exact PNE may not exist, we search for $\alpha^* = \inf\{\alpha \geq 1 : \text{an } \alpha\text{-PNE exists}\}$. The search is driven by the following monotonicity properties of α -best-response sets, which follow directly from Assumption 4.5(ii). Let $1 \leq \alpha_1 \leq \alpha_2$. Since deviation costs are nonnegative, $\pi^i(x) \leq \alpha_1 \pi^i(\hat{x}^i, x^{-i})$ implies $\pi^i(x) \leq \alpha_2 \pi^i(\hat{x}^i, x^{-i})$; hence $\mathcal{BR}_i^{\alpha_1}(x^{-i}) \subseteq \mathcal{BR}_i^{\alpha_2}(x^{-i})$ for every player $i \in N$ and every opponents' profile x^{-i} , and the α -PNE sets are nested, $\mathcal{E}^{\alpha_1}(\mathcal{G}) \subseteq \mathcal{E}^{\alpha_2}(\mathcal{G})$. In particular, if an α_1 -PNE exists, then an α_2 -PNE exists. It follows that every CEI generated at level α_2 remains valid at every level $\alpha_1 \in [1, \alpha_2]$, since an inequality valid for $\mathcal{E}^{\alpha_2}(\mathcal{G})$ is valid for every subset. Note the direction: validity propagates to *smaller* α , matching the **FOUND** branch of the bisection (which decreases α). When the bisection *increases* α (the **INFEASIBLE** branch), pooled cuts are not valid as written; instead, each pooled cut is stored as its generating deviation $\hat{x}^i$ and re-instantiated at the new level—the coefficients are unchanged and only the multiplier α is rescaled, so re-instantiation is free.

Algorithm 4.20 α -Bisection Search (ABS)

Require: Tolerance ε , bounds $1 \leq \alpha_{\min} \leq \alpha_{\max}$ with known $\alpha_{\max}$ -PNE $x_{\max}$

Ensure: $(\alpha_L, \alpha_U, x^*)$ with $\alpha_U - \alpha_L \leq \varepsilon$ and x^* an α_U -PNE

- 1: $\alpha_L \leftarrow \alpha_{\min}; \alpha_U \leftarrow \alpha_{\max}; x^* \leftarrow x_{\max}; \Phi \leftarrow \emptyset$
 - 2: **while** $\alpha_U - \alpha_L > \varepsilon$ **do**
 - 3: $\alpha \leftarrow (\alpha_L + \alpha_U)/2; (x, status, \Phi) \leftarrow \text{GZR}(\alpha, \Phi)$ $\triangleright$ Algorithm 4.8; Φ stores
deviations, re-instantiated as level- α CEIs
 - 4: **if** $status = \text{FOUND}$ **then** $\alpha_U \leftarrow \alpha, x^* \leftarrow x$ **else** $\alpha_L \leftarrow \alpha$
 - 5:
 - 6: **return** $(\alpha_L, \alpha_U, x^*)$
-

Algorithm 4.20 returns an α_U -PNE x^* with $\alpha_U - \alpha^* \leq \varepsilon$ in $O(\log((\alpha_{\max} - \alpha_{\min})/\varepsilon))$ GZR calls, provided $\alpha_{\min} \leq \alpha^*$; we always take $\alpha_{\min} = 1$, for which this holds by definition of α^* . As noted above, CEIs generated at any level remain valid at all smaller levels, so the pool Φ carries over directly on the FOUND branch; on the INFEASIBLE branch, the pooled deviations are re-instantiated at the new (larger) α at no cost. A good initial upper bound $\alpha_{\max}$ is therefore critical for efficiency.

Initializing $\alpha_{\max}$ via RRR-BRD. We obtain $\alpha_{\max}$ by running RRR-BRD (Algorithm 4.19). Each end-of-round profile $\bar{x}^{(r)}$ is a candidate approximate equilibrium whose quality is measured by $\alpha(\bar{x}^{(r)}) := \max_{i \in N} \pi^i(\bar{x}^{(r)})/\pi^i(\hat{x}^i, \bar{x}^{(r), -i})$, where $\hat{x}^i$ is the exact best response of player i . We set $\alpha_{\max} := \min_r \alpha(\bar{x}^{(r)})$. If any round converges to a PNE (certified by no-change), we have $\alpha_{\max} = 1$ and ABS is unnecessary. In our experiments, $R = 20$ rounds suffice to produce effective initial bounds.

4.5 Applications

4.5.1 Integer Splittable Bin Packing (ISBP)

Integer-splittable bin packing (ISBP) games, studied by [84], naturally model decentralized allocation problems in which self-interested agents share capacitated resources and benefit from cost sharing, such as logistics, inventory allocation, and distributed resource management. They are variants of the classical bin packing games introduced by [18]. In an ISBP game, each player $i \in [n]$ controls an item of integer size w_i and decides how to split this item into integer units across m bins. Each bin $j \in [m]$ has integer capacity $u_j \in \mathbb{Z}^+$ and constant cost $c_j > 0$, shared proportionally among the players using the bin. The strategy of player i is an integer vector $x^i = (x_j^i)_{j \in [m]}$, where x_j^i denotes the number of units of item i assigned to bin j . Let $x_j = \sum_{i \in [n]} x_j^i$ denote the total load on bin j , and $x_j^{-i} = \sum_{k \neq i} x_j^k$ the load due to the opponents of player i . The allocation must assign all units of item i and must not violate the bin capacities. Throughout, we use the convention that the cost contribution of an empty bin is zero, i.e., $c_j x_j^i / (x_j^{-i} + x_j^i) := 0$ whenever $x_j^i = 0$. Given the opponents' loads x^{-i} , player i solves:

$$\min_{x^i} \left\{ \sum_{j \in [m]} \frac{c_j}{x_j^{-i} + x_j^i} x_j^i : \sum_{j \in [m]} x_j^i = w_i, x_j^{-i} + x_j^i \leq u_j \ \forall j \in [m], x_j^i \in \mathbb{Z}_{\geq 0} \ \forall j \in [m] \right\}. \quad (4.7)$$

An ISBP game instance is called *uniform* if all bins have the same capacity, and *non-uniform* if bins may have different capacities. Moreover, a uniform instance is called *regular* if all bins have the same cost and all items have the same size. Hao and Michini [84] showed that computing a socially optimal state is NP-hard for non-uniform ISBP game instances. This implies that solving the best response problem of a single player is NP-hard in the non-uniform instances.

4.5.1.1 One-Hot Encoding and Lifted Formulation

The cost π^i is nonlinear in x due to the ratio x_j^i/x_j . To linearize it, we introduce a one-hot encoding over all possible individual contributions and total bin loads. For each player $i \in [n]$, bin $j \in [m]$, individual contribution $s \in \{0, \dots, w_i\}$, and total load $t \in \{0, \dots, u_j\}$ with $s \leq \min\{t, w_i\}$, we introduce a binary variable $y_{i,s}^{j,t} \in \{0, 1\}$ that equals one if and only if $x_j^i = s$ and $x_j = t$. We also introduce binary variables $z_t^j \in \{0, 1\}$, where $z_t^j = 1$ indicates that the total load of bin j is exactly t . The joint feasible set $\mathcal{X}$ is encoded by the following constraints, which define the polyhedron $\mathcal{H}$ of the lifted formulation (Section 4.2.1):

$$\sum_{t=0}^{u_j} \sum_{s=0}^{\min\{t, w_i\}} y_{i,s}^{j,t} = 1 \quad \forall i \in [n], j \in [m], \quad (4.8a)$$

$$\sum_{j \in [m]} \sum_{t=0}^{u_j} \sum_{s=0}^{\min\{t, w_i\}} s y_{i,s}^{j,t} = w_i \quad \forall i \in [n], \quad (4.8b)$$

$$\sum_{s=0}^{\min\{t, w_i\}} y_{i,s}^{j,t} \leq z_t^j \quad \forall i \in [n], j \in [m], t = 0, \dots, u_j, \quad (4.8c)$$

$$\sum_{t=0}^{u_j} z_t^j = 1 \quad \forall j \in [m], \quad (4.8d)$$

$$\sum_{i \in [n]} \sum_{s=0}^{\min\{t, w_i\}} s y_{i,s}^{j,t} = t z_t^j \quad \forall j \in [m], t = 0, \dots, u_j, \quad (4.8e)$$

$$y_{i,s}^{j,t} \in \{0, 1\} \quad \forall i \in [n], j \in [m], t = 0, \dots, u_j, s = 0, \dots, \min\{t, w_i\}, \quad (4.8f)$$

$$z_t^j \in \{0, 1\} \quad \forall j \in [m], t = 0, \dots, u_j. \quad (4.8g)$$

Constraint (4.8a) is the one-hot selection constraint: for every player–bin pair (i, j) , exactly one pair (s, t) must be selected. Constraint (4.8b) enforces each player’s demand requirement. Constraint (4.8c) ensures that each player selects a state consistent with the total load declared by z_t^j , and Constraint (4.8d) requires each bin to declare exactly one total-load level, forcing all players to agree on the same total load for bin j . Constraint (4.8e) is the

coupling constraint: if $z_t^j = 1$, the sum of all players' contributions to bin j must equal t .

For each player i , the lifted cost $\tilde{\pi}^i$ is:

$$\tilde{\pi}^i = \sum_{j \in [m]} \sum_{t=1}^{u_j} \sum_{s=1}^{\min\{t, w_i\}} c_j \frac{s}{t} y_{i,s}^{j,t}, \quad (4.9)$$

which is affine in the y variables and equals the original nonlinear cost π^i at every integer point of $\mathcal{H}$.

The one-hot formulation introduces $O(n \cdot m \cdot u \cdot w)$ binary variables and a similar number of constraints, where $u = \max_j u_j$ and $w = \max_i w_i$. While this may appear large, the instances in our computational study have moderate values of u and w , and the formulation remains tractable at these sizes: for the largest regular instances ($n = 10$, $m = 13$, $w = 15$, $u \leq 23$), the lifting has roughly 3.5×10^4 binary variables per game. The lifting thus confines the approach to instances with small integer demands and capacities; by contrast, the method of Duguet et al. [62] operates directly on the mixed-integer formulation without such a lifting, at the price of locally valid cuts. The key advantage is that it linearizes the proportional cost-sharing term $c_j s/t$ exactly. Alternative formulations based directly on x_j^i would require linearizing the bilinear term $x_j^i / \sum_k x_j^k$, which introduces products of integer variables and is significantly harder to handle in a branch-and-cut framework.

Best response subproblem. In GZR, the separation oracle requires solving the best response subproblem for player i given fixed x^{-i} . Since x_j^{-i} is known, the total load on bin j is $x_j^{-i} + s$ for each choice s , and no z variables are needed. Let $\bar{u}_{ij} = \min\{w_i, u_j - x_j^{-i}\}$ denote the maximum units player i can assign to bin j , reflecting both the residual capacity and the player's own demand. Introducing $y_{i,s}^j \in \{0, 1\}$ for $s \in \{0, \dots, \bar{u}_{ij}\}$, the best response

subproblem reduces to:

$$\min \sum_{j \in [m]} \sum_{s=1}^{\bar{u}_{ij}} c_j \frac{s}{x_j^{-i} + s} y_{i,s}^j \quad (4.10a)$$

$$\text{s.t. } \sum_{s=0}^{\bar{u}_{ij}} y_{i,s}^j = 1 \quad \forall j \in [m], \quad (4.10b)$$

$$\sum_{j \in [m]} \sum_{s=0}^{\bar{u}_{ij}} s y_{i,s}^j = w_i, \quad (4.10c)$$

$$y_{i,s}^j \in \{0, 1\} \quad \forall j \in [m], s = 0, \dots, \bar{u}_{ij}. \quad (4.10d)$$

This is a significantly smaller MIP than the joint formulation, with $O(m \cdot w)$ binary variables per player and no z variables. Once the best response MIP is solved, the optimal deviation $\hat{x}^i$ used in the CEI is recovered as:

$$\hat{x}_j^i = \sum_{s=0}^{\bar{u}_{ij}} s y_{i,s}^j \quad \forall j \in [m],$$

since exactly one $y_{i,s}^j$ equals 1 for each bin j by (4.10b).

4.5.1.2 CEI Encoding and Activation

To compute a PNE, we embed the lifted formulation $\mathcal{H}$ into the GZR algorithm. The CEIs enforce the no-regret condition: whenever the current allocation admits a feasible improving deviation for some player, GZR adds a CEI that cuts off this non-equilibrium allocation.

Fix a player $i \in [n]$ and a candidate deviation $\hat{x}^i = (\hat{x}_j^i)_{j \in [m]} \in \mathbb{Z}^m$, a feasible allocation for player i satisfying $\sum_j \hat{x}_j^i = w_i$ and $0 \leq \hat{x}_j^i \leq u_j$ for all j . Let $\hat{B}_i = \{j \in [m] : \hat{x}_j^i > 0\}$ denote the set of bins used by player i in the deviation. When the current state of bin j is (s, t) (i.e., $y_{i,s}^{j,t} = 1$), the opponents' load on bin j is $t - s$, so the new total load after the

deviation is $(t - s) + \hat{x}_j^i$. The deviation is feasible on bin j if and only if $(t - s) + \hat{x}_j^i \leq u_j$, equivalently $s \geq t - u_j + \hat{x}_j^i$. For each $j \in \hat{B}_i$, define the index sets of feasible and infeasible states:

$$\begin{aligned}\mathcal{F}_j(\hat{x}_j^i) &= \{(s, t) : 0 \leq t \leq u_j, \max\{0, t - u_j + \hat{x}_j^i\} \leq s \leq \min\{t, w_i\}\}, \\ \mathcal{J}_j(\hat{x}_j^i) &= \{(s, t) : u_j - \hat{x}_j^i + 1 \leq t \leq u_j, 0 \leq s \leq \min\{t - u_j + \hat{x}_j^i - 1, w_i\}\}.\end{aligned}$$

The cost of player i after deviating to $\hat{x}^i$, given current state (s, t) on bin $j \in \hat{B}_i$, is $c_j \hat{x}_j^i / ((t - s) + \hat{x}_j^i)$, a known constant since $\hat{x}^i$ is fixed at separation time. The lifted CEI is:

$$\sum_{j \in [m]} \sum_{t=1}^{u_j} \sum_{s=1}^{\min\{t, w_i\}} c_j \frac{s}{t} y_{i,s}^{j,t} \leq \alpha \sum_{j \in \hat{B}_i} \sum_{(s,t) \in \mathcal{F}_j(\hat{x}_j^i)} c_j \frac{\hat{x}_j^i}{(t-s) + \hat{x}_j^i} y_{i,s}^{j,t} + M \sum_{j \in \hat{B}_i} \sum_{(s,t) \in \mathcal{J}_j(\hat{x}_j^i)} y_{i,s}^{j,t}. \quad (4.11)$$

The left-hand side is player i 's current cost (4.9). The first term on the right is α times the deviation cost, summing over feasible states in $\mathcal{F}_j(\hat{x}_j^i)$ for each $j \in \hat{B}_i$, with known coefficients $c_j \hat{x}_j^i / ((t - s) + \hat{x}_j^i)$. The second term is the activation term: for infeasible states in $\mathcal{J}_j(\hat{x}_j^i)$, the big- M penalty relaxes the inequality, rendering it inactive. We set $M = \sum_{j \in [m]} c_j$ as the sum of the cost over all bins, which is always larger than any possible player's cost in the game if $n \geq 2$. When the deviation is feasible on all bins, $\mathcal{J}_j(\hat{x}_j^i) = \emptyset$ for all j and the CEI enforces the α -approximate no-regret condition directly. Therefore, when GZR finds a violated CEI, the current allocation is not an α -PNE; when no violated CEI exists for any player, the allocation is an α -PNE.

4.5.1.3 VEST in ISBP

In an ISBP game, the cost of player i on bin j is:

$$f_j^i(x_j^i) = \frac{c_j x_j^i}{x_j^{-i} + x_j^i},$$

with the convention $f_j^i(0) = 0$, so that f_j^i is well defined on $[0, u_j - x_j^{-i}]$ also when $x_j^{-i} = 0$. When bin j is shared ($x_j^{-i} > 0$), f_j^i is strictly concave in x_j^i . However, when bin j is solo ($x_j^{-i} = 0$), f_j^i equals c_j for all $x_j^i > 0$ and 0 for $x_j^i = 0$, so it is affine and hence not strictly concave on $(0, u_j]$. Therefore, Assumption 4.15 does not hold in general for ISBP games, and the Strong VEST theorem does not directly apply.

We instead apply Corollary 4.17. The single-resource cost $f_j^i(s) = c_j s / (x_j^{-i} + s)$ is strictly concave in s on $(0, u_j]$ whenever $x_j^{-i} > 0$ (a shared bin), and is affine—in fact constant, equal to c_j —for $s > 0$ whenever $x_j^{-i} = 0$ (a solo bin). Hence, from the definitions of S_+^i and S_0^i , for every free bin $j \in \text{free}(x^i)$,

$$j \in S_+^i(x^i) \iff x_j^{-i} > 0 \text{ (shared free bin)}, \quad j \in S_0^i(x^i) \iff x_j^{-i} = 0 \text{ (solo free bin)}.$$

By Corollary 4.17, every feasible direction d within the minimal face F of $P^i(x^{-i})$ containing x^i must satisfy $d_j = 0$ for all shared free bins $j \in S_+^i(x^i)$: the only bins along which a feasible direction can move are solo free bins.

We now connect this to the vertex structure of $P^i(x^{-i})$. In an ISBP game, the conditional polytope is a bounded simplex with constraint matrix $A^i = \mathbf{1}^\top$, so feasible directions within the minimal face F satisfy $\sum_{j \in \text{free}(x^i)} d_j = 0$. Combined with $d_j = 0$ for all $j \in S_+^i(x^i)$, this gives $\sum_{j \in S_0^i(x^i)} d_j = 0$. The following corollary shows that no free bin can be shared. The following corollary generalizes Lemma 2 in [84] from uniform to non-uniform instances.

Corollary 4.21. *Let x be a PNE of a non-uniform ISBP instance. If player $i \in [n]$ uses two unsaturated bins j and k in x , then neither j nor k is shared with any other player.*

Proof. Suppose for contradiction that player i uses two unsaturated bins and at least one of them, say j , is shared, so $x_j^{-i} > 0$ and $j \in S_+^i(x^i)$ (recall $S_+^i(x^i) = \{j \in \text{free}(x^i) : f_j^i(\cdot; x^{-i}) \text{ is strictly concave at } x_j^i\}$, which in ISBP equals the set of shared free bins). Since $|\text{free}(x^i)| \geq 2$, x^i is not a vertex of $P^i(x^{-i})$ by Lemma 4.13. Let F be the minimal face of $P^i(x^{-i})$ containing x^i . By Corollary 4.17, there exists a nonzero feasible direction d within F , and every such direction satisfies $d_l = 0$ for all $l \in S_+^i(x^i)$.

Since $A^i = \mathbf{1}^\top$, every feasible direction within F also satisfies $\sum_{l \in \text{free}(x^i)} d_l = 0$. Since $d_l = 0$ for all $l \in S_+^i(x^i)$, this reduces to $\sum_{l \in S_0^i(x^i)} d_l = 0$ (recall $S_0^i(x^i) = \{j \in \text{free}(x^i) : f_j^i(\cdot; x^{-i}) \text{ is locally affine at } x_j^i\}$, the set of solo free bins). For d to be nonzero with $d_l = 0$ on $S_+^i(x^i)$, at least one component in $S_0^i(x^i)$ must be nonzero, which requires $|S_0^i(x^i)| \geq 2$.

Now take any $l \in S_0^i(x^i)$ and consider the direction d' defined by $d'_j = 1$, $d'_l = -1$, and $d'_r = 0$ for all other r . Since $j, l \in \text{free}(x^i)$ and $\sum_r d'_r = 0$, d' is a feasible direction within F . But $d'_j = 1 \neq 0$ with $j \in S_+^i(x^i)$, contradicting Corollary 4.17. Therefore $S_+^i(x^i) = \emptyset$: no free bin can be shared. $\square$

Weak VEST cuts. Corollary 4.21 yields valid inequalities that can be added to the GZR master problem. For each player i and bin j , define:

$$g_{i,j} := \sum_{\substack{0 < t < u_j \\ 1 \leq s \leq \min\{t-1, w_i\}}} y_{i,s}^{j,t}, \quad h_{i,j} := \sum_{1 \leq t \leq \min\{u_j-1, w_i\}} y_{i,t}^{j,t}, \quad f_{i,j} := g_{i,j} + h_{i,j},$$

indicating, respectively, whether bin j is *shared-unsaturated*, *solo-unsaturated*, or *unsaturated* for player i . Let $G_i := \sum_{j=1}^m g_{i,j}$ count the number of shared-unsaturated bins used by player

i. The weak VEST cuts are:

$$G_i \leq 1 \quad \forall i \in [n], \quad (4.12)$$

$$g_{i,j_1} + f_{i,j_2} \leq 1 \quad \forall i \in [n], j_1 \neq j_2. \quad (4.13)$$

Constraint (4.12) ensures that each player has at most one shared-unsaturated bin. Constraint (4.13) ensures that if player i uses a shared-unsaturated bin, it cannot also use any other unsaturated bin. Together, these enforce the necessary condition of Corollary 4.21.

Strong VEST (Corollary 4.16) does not directly apply to ISBP since cost functions are not strictly concave on solo bins. However, in uniform instances we can show that every PNE can be transformed into a PNE in which every player plays a vertex strategy and every player's cost is unchanged.

Theorem 4.22. *In any uniform ISBP instance, for every PNE x , there exists a PNE $\tilde{x}$ such that $\tilde{x}^i$ is a vertex of $P^i(\tilde{x}^{-i})$ for every $i \in [n]$, and*

$$\pi^i(\tilde{x}) = \pi^i(x) \quad \forall i \in [n]. \quad (4.14)$$

Proof. Let the common bin capacity be u . For a PNE y , let

$$p(y) := |\{i \in [n] : y^i \text{ is not a vertex of } P^i(y^{-i})\}|. \quad (4.15)$$

Fix the given PNE x . Since the strategy space is finite, among all PNEs y satisfying

$$\pi^h(y) = \pi^h(x) \quad \forall h \in [n], \quad (4.16)$$

choose one, still denoted by x , that minimizes $p(x)$. We show that $p(x) = 0$.

Suppose, for contradiction, that $p(x) \geq 1$. Choose a player i such that x^i is not a vertex of $P^i(x^{-i})$. Since $P^i(x^{-i})$ is a bounded simplex, Lemma 4.13 implies that player i has at least two free coordinates, i.e., player i uses at least two unsaturated bins. By Corollary 5.1, every unsaturated bin used by player i is solo. Define

$$S := \{j \in [m] : x_j^i > 0, x_j < u\}, \quad (4.17)$$

the set of unsaturated bins used by player i . Since every bin in S is used only by player i . Let

$$r := \sum_{j \in S} (u - x_j^i) \quad (4.18)$$

be the total residual capacity in the bins of S .

We first claim that $r < u$. If $r \geq u$, let $j^* \in S$ be a bin of maximum cost among the bins in S . Since

$$\sum_{j \in S \setminus \{j^*\}} (u - x_j^i) = r - (u - x_{j^*}^i) \geq x_{j^*}^i,$$

player i could move all units from bin j^* into the other bins of S . This deviation is feasible, opens no new bin, and removes the positive cost c_{j^*} paid on the solo bin j^* . This contradicts that x^i is a best response to x^{-i} . Hence $r < u$.

Choose $j^* \in S$ with maximum cost c_{j^*} , and define a new strategy $\tilde{x}^i$ for player i by

$$\tilde{x}_j^i = \begin{cases} x_j^i, & j \notin S, \\ u, & j \in S \setminus \{j^*\}, \\ u - r, & j = j^*. \end{cases} \quad (4.19)$$

Let $\tilde{x} := (\tilde{x}^i, x^{-i})$. Since $r < u$, we have $\tilde{x}_{j^*}^i = u - r > 0$, and

$$\sum_{j \in S} \tilde{x}_j^i = (|S| - 1)u + (u - r) \quad (4.20)$$

$$= |S|u - r \quad (4.21)$$

$$= \sum_{j \in S} x_j^i. \quad (4.22)$$

Thus $\tilde{x}^i$ assigns the same total demand as x^i and is feasible against x^{-i} . Moreover, every bin in $S \setminus \{j^*\}$ is saturated under $\tilde{x}^i$, and j^* is the only possibly unsaturated bin in S . All bins outside S are unchanged. Therefore $\tilde{x}^i$ has at most one free coordinate, so it is a vertex of $P^i(x^{-i}) = P^i(\tilde{x}^{-i})$ by Lemma 4.13.

Because all bins in S are solo both before and after the modification, player i pays c_j on each bin $j \in S$ in both x and $\tilde{x}$. All other bins are unchanged, so

$$\pi^i(\tilde{x}) = \pi^i(x). \quad (4.23)$$

Since x^i is a best response to x^{-i} and $\tilde{x}^i$ is feasible with the same cost, $\tilde{x}^i$ is also a best response to x^{-i} .

Now consider any player $h \neq i$. Since every bin in S is solo for player i under x , player h

uses no bin in S . Outside S , the allocation is unchanged. Hence

$$\pi^h(\tilde{x}) = \pi^h(x). \quad (4.24)$$

We show that player h has no profitable deviation from $\tilde{x}$. Suppose otherwise, and let $\hat{x}^h$ be a profitable deviation for player h from $\tilde{x}$.

If $\hat{x}^h$ assigns no units to bins in S , then the same deviation is feasible from x and has the same cost, because all bins outside S are unchanged. This contradicts that x is a PNE.

It remains to consider the case in which $\hat{x}^h$ assigns $p \geq 1$ units to bins in S . Under $\tilde{x}$, all bins in $S \setminus \{j^*\}$ are saturated, so these p units must all be assigned to j^* . Thus feasibility of $\hat{x}^h$ implies $p \leq r$, and the cost of these p units on bins in S is

$$c_{j^*} \frac{p}{u - r + p}. \quad (4.25)$$

We construct a deviation $\bar{x}^h$ from x as follows. Outside S , $\bar{x}^h$ agrees with $\hat{x}^h$. Inside S , place the same total number p of units greedily into the bins of S starting from the bins with largest current load x_j^i . This is feasible because the total residual capacity in S under x is $r \geq p$.

Let

$$L := u - r + p. \quad (4.26)$$

After placing p units in S , the remaining total residual capacity in S is $r - p$. By the greedy construction, every bin in S that receives a positive amount has final load at least L : otherwise the last bin receiving a unit would have final load strictly smaller than L , hence

its remaining capacity alone would be strictly larger than $r - p$, contradicting that the total remaining residual capacity in all of S is exactly $r - p$.

Therefore, for every bin $j \in S$ used by $\bar{x}^h$, its final load is at least $L = u - r + p$. Since j^* has maximum cost in S , we have $c_j \leq c_{j^*}$ for all $j \in S$. Hence the cost of $\bar{x}^h$ on bins in S is at most

$$\sum_{j \in S} c_j \frac{\bar{x}_j^h}{x_j^i + \bar{x}_j^h} \leq \sum_{j \in S} c_{j^*} \frac{\bar{x}_j^h}{u - r + p} \quad (4.27)$$

$$= c_{j^*} \frac{p}{u - r + p}, \quad (4.28)$$

which is exactly the cost of the p units placed in S by $\hat{x}^h$ from $\tilde{x}$. Outside S , the two deviations have identical allocations and face identical loads. Therefore

$$\pi^h(\bar{x}^h, x^{-h}) \leq \pi^h(\hat{x}^h, \tilde{x}^{-h}) \quad (4.29)$$

$$< \pi^h(\tilde{x}) \quad (4.30)$$

$$= \pi^h(x), \quad (4.31)$$

contradicting that x is a PNE. Thus no player $h \neq i$ has a profitable deviation from $\tilde{x}$.

We have shown that $\tilde{x}$ is a PNE and that

$$\pi^h(\tilde{x}) = \pi^h(x) \quad \forall h \in [n]. \quad (4.32)$$

Moreover, player i now uses a vertex strategy, while every other player's strategy is unchanged. The vertex/non-vertex status of the other players is also unchanged: bins in S are

not used by them, and all bins outside S are unchanged. Therefore

$$p(\tilde{x}) < p(x), \quad (4.33)$$

contradicting the choice of x as a same-player-cost PNE minimizing p . Hence $p(x) = 0$. Thus every player uses a vertex strategy, and this PNE has the same cost as the originally given PNE for every player. $\square$

Strong VEST cut. Theorem 4.22 shows that in uniform instances we can always find a PNE where every player uses at most one unsaturated bin, i.e., a vertex of $P^i(x^{-i})$. We can therefore add the following cut to the GZR master problem for uniform instances:

$$\sum_{j \in [m]} \sum_{t=0}^{u_j-1} \sum_{s=1}^{\min\{t, w_i\}} y_{i,s}^{j,t} \leq 1 \quad \forall i \in [n], \quad (4.34)$$

which enforces that each player uses at most one unsaturated bin. Unlike the weak VEST cuts (4.12)–(4.13), inequality (4.34) is optimality-preserving rather than valid for every PNE: it may cut off PNE profiles, but by Theorem 4.22 some PNE with the same individual (hence social) costs satisfies it. It therefore preserves the optimal value of any objective that depends only on the players' costs, such as the social cost, and should not be imposed under arbitrary objectives over allocations.

4.5.2 Network Formation Games (NFG)

4.5.2.1 Game Definition

NFG is an integer-splittable, edge-capacitated variant of the classical network design game with Shapley cost sharing. Anshelevich et al. [3] study the unsplittable, uncapacitated ver-

sion: each player selects a single o_i - d_i path, the game admits an exact potential function, and PNE always exists with $\text{POS} \leq H(n)$. For this classical model, Dragotto and Scatamacchia [58] compute the exact instance-level POS via their ZR algorithm for IPGs. Our NFG departs from this setting in two ways: (i) players route *integer-splittable* flow (distributing demand D_i across multiple paths in integer units), and (ii) shared edge capacities $\sum_i x_e^i \leq u_e$ couple all players' feasibility. The potential argument no longer applies, PNE need not exist, and the POS bound of Anshelevich et al. [3] does not carry over. Since our NFG is a GIPG (not an IPG), the ZR framework of Dragotto and Scatamacchia [58] is also inapplicable. Our GZR framework provides the first method to compute the socially optimal PNE (and thus the exact POS) of this model when one exists; when no PNE exists, it certifies non-existence and computes the tightest α -approximate PNE. The NFG is also a considerable generalization of ISBP: the latter is the special case where the network consists of m parallel edges between a single source-sink pair ($|V| = 2$), reducing the flow polytope to a bounded simplex and Corollary 4.24 to Corollary 4.21.

Definition 4.23 (Capacitated Network Formation Game). A capacitated NFG consists of a directed network $G = (V, E)$ with edge capacities $u_e \in \mathbb{Z}_+$ and edge costs $c_e > 0$, and players $[n]$ each with origin-destination pair (o_i, d_i) and integer demand $D_i \in \mathbb{Z}_+$. Player i routes D_i units of integer-splittable flow from o_i to d_i ; let $x_e^i \in \mathbb{Z}_{\geq 0}$ denote the flow of player i on edge e , and $x_e = \sum_k x_e^k$ the total flow. Under Shapley cost sharing, player i 's cost is:

$$\pi^i(x) = \sum_{e \in E} c_e \cdot \frac{x_e^i}{x_e},$$

with the convention $c_e x_e^i / x_e = 0$ whenever $x_e^i = 0$. Given the opponents' flows x^{-i} , player

i solves:

$$\min_{x^i} \left\{ \sum_{e \in E} \frac{c_e}{x_e^{-i} + x_e^i} x_e^i : x^i \text{ is an } o_i\text{-}d_i \text{ flow of value } D_i, x_e^{-i} + x_e^i \leq u_e \ \forall e \in E, x_e^i \in \mathbb{Z}_{\geq 0} \ \forall e \in E \right\}.$$

4.5.2.2 One-Hot Encoding and Lifted Formulation

The cost structure of the NFG is identical to that of ISBP, with edges playing the role of bins and total edge flow playing the role of total bin load. We therefore use the same one-hot encoding: for each player $i \in [n]$, edge $e \in E$, individual flow $s \in \{0, \dots, D_i\}$, and total flow $t \in \{0, \dots, u_e\}$ with $s \leq \min\{t, D_i\}$, introduce a binary variable $y_{i,s}^{e,t} \in \{0, 1\}$ equal to one if and only if $x_e^i = s$ and $x_e = t$. Similarly, introduce $z_t^e \in \{0, 1\}$ indicating that the total flow on edge e is exactly t . The constraints (4.8a) and (4.8c)–(4.8g) carry over with bins replaced by edges, w_i replaced by D_i , and u_j replaced by u_e ; the per-player demand constraint (4.8b) does *not* carry over (a flow of value D_i on a path of length ℓ uses ℓD_i edge-units) and is replaced by the flow-conservation constraints below, which enforce each player's demand at the source.

The only additional constraints relative to ISBP are the *flow conservation* constraints, which ensure that player i 's flow is a valid $o_i\text{-}d_i$ flow of value D_i :

$$\sum_{e \in \delta^+(v)} \sum_{t,s} s y_{i,s}^{e,t} - \sum_{e \in \delta^-(v)} \sum_{t,s} s y_{i,s}^{e,t} = \begin{cases} D_i & v = o_i, \\ -D_i & v = d_i, \\ 0 & \text{otherwise,} \end{cases} \quad \forall i \in [n], v \in V, \quad (4.35)$$

where $\delta^+(v)$ and $\delta^-(v)$ denote the sets of edges leaving and entering node v . Together with the constraints inherited from ISBP, these define the polyhedron $\mathcal{H}$ for the NFG. The lifted

cost $\tilde{\pi}^i$ is:

$$\tilde{\pi}^i = \sum_{e \in E} \sum_{t=1}^{u_e} \sum_{s=1}^{\min\{t, D_i\}} c_e \frac{s}{t} y_{i,s}^{e,t},$$

which is affine in the y variables and equals π^i at every integer point of $\mathcal{H}$.

Best response subproblem. As in ISBP, when x^{-i} is fixed the total flow on edge e is $x_e^{-i} + s$, and no z variables are needed. Let $\bar{u}_{ie} = \min\{D_i, u_e - x_e^{-i}\}$ denote the maximum flow player i can route on edge e . Introducing $y_{i,s}^e \in \{0, 1\}$ for $s \in \{0, \dots, \bar{u}_{ie}\}$, the best response subproblem is a min-cost integer flow problem:

$$\min \sum_{e \in E} \sum_{s=1}^{\bar{u}_{ie}} c_e \frac{s}{x_e^{-i} + s} y_{i,s}^e \quad (4.36a)$$

$$\text{s.t. } \sum_{s=0}^{\bar{u}_{ie}} y_{i,s}^e = 1 \quad \forall e \in E, \quad (4.36b)$$

$$\sum_{e \in \delta^+(v)} \sum_{s=0}^{\bar{u}_{ie}} s y_{i,s}^e - \sum_{e \in \delta^-(v)} \sum_{s=0}^{\bar{u}_{ie}} s y_{i,s}^e = \begin{cases} D_i & v = o_i \\ -D_i & v = d_i \\ 0 & \text{otherwise} \end{cases} \quad \forall v \in V, \quad (4.36c)$$

$$y_{i,s}^e \in \{0, 1\} \quad \forall e \in E, s = 0, \dots, \bar{u}_{ie}. \quad (4.36d)$$

The optimal deviation $\hat{x}^i$ is recovered as $\hat{x}_e^i = \sum_s s y_{i,s}^e$ for each $e \in E$.

4.5.2.3 CEI Encoding and Activation

The CEI for the NFG has the same structure as for ISBP (Section 4.5.1.2), with edges replacing bins. Fix a player $i \in [n]$ and a candidate deviation $\hat{x}^i = (\hat{x}_e^i)_{e \in E}$, a feasible o_i - d_i flow of value D_i satisfying $\hat{x}_e^i \leq u_e$ for all e . Let $\hat{E}_i = \{e \in E : \hat{x}_e^i > 0\}$ be the set of edges

used by the deviation. For each $e \in \hat{E}_i$, define:

$$\begin{aligned}\mathcal{F}_e(\hat{x}_e^i) &= \{(s, t) : 0 \leq t \leq u_e, \max\{0, t - u_e + \hat{x}_e^i\} \leq s \leq \min\{t, D_i\}\}, \\ \mathcal{J}_e(\hat{x}_e^i) &= \{(s, t) : u_e - \hat{x}_e^i + 1 \leq t \leq u_e, 0 \leq s \leq \min\{t - u_e + \hat{x}_e^i - 1, D_i\}\}.\end{aligned}$$

The lifted CEI is:

$$\sum_{e \in E} \sum_{t=1}^{u_e} \sum_{s=1}^{\min\{t, D_i\}} c_e \frac{s}{t} y_{i,s}^{e,t} \leq \alpha \sum_{e \in \hat{E}_i} \sum_{(s,t) \in \mathcal{F}_e(\hat{x}_e^i)} c_e \frac{\hat{x}_e^i}{(t-s) + \hat{x}_e^i} y_{i,s}^{e,t} + M \sum_{e \in \hat{E}_i} \sum_{(s,t) \in \mathcal{J}_e(\hat{x}_e^i)} y_{i,s}^{e,t}. \quad (4.37)$$

The interpretation is identical to (4.11): the left-hand side is player i 's current cost, the first term on the right is α times the deviation cost over feasible states, and the activation term relaxes the inequality when the deviation is infeasible on some edge.

4.5.2.4 VEST in NFG

The cost structure of the NFG is identical to that of ISBP at the edge level: $f_e^i(x_e^i) = c_e x_e^i / (x_e^{-i} + x_e^i)$ is strictly concave when $x_e^{-i} > 0$ (shared edge) and affine when $x_e^{-i} = 0$ (solo edge). By the same argument as in Section 4.5.1.3, $S_+^i(x^i)$ consists of shared free edges and $S_0^i(x^i)$ consists of solo free edges.

The key difference from ISBP is the structure of the conditional polytope $P^i(x^{-i})$, which is now a *flow polytope* rather than a bounded simplex. The constraint matrix A^i is the node-edge incidence matrix of G , whose kernel restricted to the free edges $\text{free}(x^i)$ is the *cycle space* of the subgraph induced by $\text{free}(x^i)$. Feasible directions within the minimal face F correspond to circulations on the free subgraph, i.e., flows along undirected cycles of free edges.

By Corollary 4.17, every feasible direction d within F must have $d_e = 0$ for all $e \in S_+^i(x^i)$.

Since feasible directions correspond to circulations, this means:

Corollary 4.24. *Let x be a PNE of an NFG. If x^i is not a vertex of $P^i(x^{-i})$, then every undirected cycle in the subgraph of free edges $\text{free}(x^i)$ consists entirely of solo edges (edges with $x_e^{-i} = 0$). In particular, the shared free edges form a forest within the free subgraph.*

Proof. By Corollary 4.17, every feasible direction d within the minimal face F of $P^i(x^{-i})$ containing x^i satisfies $d_e = 0$ for all $e \in S_+^i(x^i)$. A feasible direction within F corresponds to a circulation on the free subgraph, i.e., a flow along undirected cycles of free edges. If any such cycle passed through a shared free edge $e \in S_+^i(x^i)$, the corresponding circulation would have $d_e \neq 0$, a contradiction. Hence no undirected cycle in the free subgraph can pass through a shared free edge, meaning the shared free edges contain no cycle and form a forest. $\square$

Weak VEST cuts. Corollary 4.24 yields valid inequalities that can be added to the GZR master problem. For each player $i \in [n]$ and each undirected cycle C in G , define:

$$g_{i,e} := \sum_{t=1}^{u_e-1} \sum_{s=1}^{\min\{t-1, D_i\}} y_{i,s}^{e,t},$$

which equals one if and only if edge e is unsaturated, shared, and used by player i (i.e., $0 < s < t < u_e$). Define the aggregate counts for cycle C :

$$G_i^C := \sum_{e \in C} g_{i,e}, \quad H_i^C := \sum_{e \in C} \sum_{t=1}^{\min\{u_e-1, D_i\}} y_{i,t}^{e,t},$$

where G_i^C counts the unsaturated shared edges of C used by player i , and H_i^C counts the unsaturated solo edges of C used by player i (those with $s = t < u_e$). The sum $G_i^C + H_i^C$ equals $|C|$ if and only if player i uses all edges of C and none are saturated. By Corollary 4.24,

in that case all edges must be solo, i.e., $G_i^C = 0$. This is enforced by the following cut, for each player $i \in [n]$, cycle C , and edge $e \in C$:

$$g_{i,e} + (G_i^C + H_i^C - |C|) \leq 0. \quad (4.38)$$

When $G_i^C + H_i^C = |C|$ (player i uses all edges of C and none are saturated), this forces $g_{i,e} = 0$ for every $e \in C$, eliminating all shared free edges from the cycle. When $G_i^C + H_i^C < |C|$, the constraint is automatically satisfied since $g_{i,e} \geq 0$ and the right-hand side is non-negative.

4.5.3 Generalized Knapsack Games (GKG and GKGB)

4.5.3.1 Game Definitions

Definition 4.25 (Generalized Knapsack Game). A GKG consists of n players and m items. Player i has a private 0-1 knapsack with profits $p_{ij} \geq 0$, weights w_{ij} , and capacity b_i : $\mathcal{X}_{\text{private}}^i = \{x^i \in \{0,1\}^m : \sum_j w_{ij}x_j^i \leq b_i\}$. Items have shared capacities: $\sum_i x_j^i \leq c_j$ for all j , where $c_j \in \{1, \dots, n\}$. Player i minimizes $\pi_i(x) = P_i - \sum_j p_{ij}x_j^i$, where $P_i := \sum_j p_{ij}$ ensures $\pi_i \geq 0$.

GKGB extends GKG by incorporating *reciprocally bilinear interaction terms* from Dragotto and Scatamacchia [58]. In addition to the private knapsack profits, each player's cost depends on opponents' item selections through pairwise interactions.

Definition 4.26 (Generalized Knapsack Game with Bilinear Coefficients). A GKGB is a GKG (Definition 4.25) in which each player i 's cost includes bilinear interaction terms:

$$\pi_i(x) = P_i - \sum_j p_{ij}x_j^i - \sum_{k \neq i} \sum_j C_{kj}^i x_j^i x_j^k,$$

where $C_{kj}^i \in \mathbb{R}$ captures the externality that player k 's selection of item j imposes on player i . Positive coefficients represent synergistic effects; negative coefficients represent adversarial externalities. In GKGB the offset is enlarged relative to GKG,

$$P_i := \sum_j p_{ij} + \sum_{k \neq i} \sum_j \max\{C_{kj}^i, 0\},$$

which ensures $\pi_i \geq 0$ also in the presence of positive interaction coefficients.

The interpretation of the players' incentives is unchanged: the offset is a constant per player, so exact best responses and exact PNE are unaffected, while nonnegativity of costs is required by Assumption 4.5(ii) and by the α -approximate machinery (Definition 4.1 and the monotonicity properties of Section 4.4.2).

Since both GKG and GKGB use binary decision variables and the cost is linear (or bilinear) in the strategy profile, no auxiliary encoding is required: the variables $x_{ij} \in \{0, 1\}$ directly represent the strategy. This contrasts with ISBP and NFG, where the nonlinear proportional cost-sharing term necessitates a one-hot encoding (Sections 4.5.1.1 and 4.5.2).

4.5.3.2 CEI Encoding

Binary variables $x_{ij} \in \{0, 1\}$ for each player–item pair. Feasibility indicators $z_{ij}^{\text{full}} \in \{0, 1\}$ track whether item j is at capacity from player i 's perspective:

$$\sum_{k \neq i} x_{kj} \leq (c_j - 1) + (n - 1)z_{ij}^{\text{full}} \tag{4.39}$$

$$\sum_{k \neq i} x_{kj} \geq c_j z_{ij}^{\text{full}}. \tag{4.40}$$

GKG. For player i and deviation $\hat{x}^i$, the CEI is:

$$P_i - \sum_j p_{ij} x_{ij} \leq \alpha \left(P_i - \sum_j p_{ij} \hat{x}_j^i \right) + M \sum_{j: \hat{x}_j^i = 1} z_{ij}^{\text{full}}. \quad (4.41)$$

When the deviation is feasible ($z_{ij}^{\text{full}} = 0$ for all j with $\hat{x}_j^i = 1$), this enforces the α -equilibrium condition. When infeasible, the big- M term deactivates the constraint.

GKGB. For a fixed incumbent profile $\hat{x}$, the bilinear interaction terms in the best-response deviation collapse to terms linear in x_j^k , so the CEI remains linear in the master decision variables. The feasibility indicators z_{ij}^{full} carry over unchanged. For player i and deviation $\hat{x}^i$, the CEI is:

$$P_i - \sum_j p_{ij} x_{ij} - \sum_{k \neq i} \sum_j C_{kj}^i x_{ij} x_{kj} \leq \alpha \left(P_i - \sum_j p_{ij} \hat{x}_j^i - \sum_{k \neq i} \sum_j C_{kj}^i \hat{x}_j^i x_{kj} \right) + M \sum_{j: \hat{x}_j^i = 1} z_{ij}^{\text{full}}. \quad (4.42)$$

In the master MILP, each bilinear term $x_j^i \cdot x_j^k$ is linearized via McCormick relaxation with auxiliary variable y_j^{ik} and constraints $y_j^{ik} \leq x_j^i$, $y_j^{ik} \leq x_j^k$, $y_j^{ik} \geq x_j^i + x_j^k - 1$, $y_j^{ik} \geq 0$. For GKGB we take $M_i = P_i + \sum_{k \neq i} \sum_j \max\{-C_{kj}^i, 0\}$, an upper bound on any realized cost π_i .

4.6 Computational Study

4.6.1 Experimental Design

We evaluate the GZR algorithm on four game classes: ISBP, NFG, GKG, and GKGB. For each class, we conduct up to two experiments.

Experiment 1: Best PNE and Price of Stability. We first run *round random-restart*

best-response dynamics (RRR-BRD) [98], a heuristic that performs up to $R = 20$ rounds of sequential best-response updates in uniformly random player orders across multiple random restarts. If any round produces a profile with no improving deviation for any player, it is certified as an exact PNE and the heuristic terminates early. Otherwise, the approximation ratio α is evaluated for each end-of-round profile from the final restart, and the profile with the smallest α is selected. This best candidate serves as a warm-start for GZR (Algorithm 4.8), which optimizes social welfare over the set of α -PNE and computes the exact price of stability (POS).

Experiment 2: α -Bisection Search (ABS). For instances where Experiment 1 finds no exact PNE—i.e., GZR returns `INFEASIBLE` or hits the time limit without confirming a PNE—we apply the ABS algorithm (Algorithm 4.20) to find the tightest α -approximate PNE via bisection. ABS uses the BRD-computed α as the initial upper bound $\alpha_{\max}$, and a per-iteration GZR time limit of 10 minutes (600s). ABS is invoked only for ISBP and GKGB, since every GKG and NFG instance admits an exact PNE. Two distinct tolerances are involved, and we keep them separate. The *stopping tolerance* governs when the bisection halts and was set to 10^{-3} for ISBP and 10^{-2} for GKGB, the looser value reflecting the higher cost of each bisection step in the McCormick-linearized GKGB formulation. The *verification threshold* ε is the width below which we declare the certified interval $[\alpha^{\text{LB}}, \alpha^{\text{UB}}]$ tight enough to report as verified; we apply a single value $\varepsilon = 10^{-2}$ to both games, so that the Verified column has the same meaning throughout. This is a reporting criterion applied to intervals that are certified independently of it: α^{UB} is an α at which GZR exhibited an α -PNE and α^{LB} is an α at which GZR proved none exists, so both endpoints remain valid regardless of the threshold used to summarize them. By the monotonicity properties of Section 4.4.2, all CEIs generated across iterations carry over, providing computational savings. The VEST cut and the player-anchoring constraint are established for exact equilibria and are therefore

imposed only in the initial test at $\alpha = 1$; every bisection call at $\alpha > 1$ omits them, since with either constraint active an **INFEASIBLE** outcome would not certify that no α -PNE exists. The bin-load ordering constraint is valid at every α and is kept throughout.

Implementation. Our implementation, instance generators, and scripts reproducing all tables are publicly available at <https://github.com/HyunwooLee0429/GIPG>. All experiments are implemented in Python using Gurobi 13.0 with lazy constraint callbacks for CEI separation. Both the jointly constrained master problem and the best-response subproblems are solved with 16 threads. The GZR time limit is 10 minutes (600 s) per instance. Experiments are run on a single machine with an Intel Core Ultra 7 265F processor (2.40 GHz) and 32 GB RAM.

Solver settings and the meaning of reported statuses. The master problem is solved with `LazyConstraints = 1`; the social-optimum reference model, which requires no separation, is solved with `LazyConstraints = 0`. All remaining tolerances are left at their Gurobi defaults, in particular `MIPGap = 10-4`, `IntFeasTol = 10-5`, and `FeasibilityTol = 10-6`, for both the master and the best-response subproblems. Accordingly, a status of **OPTIMAL** certifies optimality within the default relative gap rather than in exact arithmetic. This distinction is immaterial at the scales considered here: across all instances reported as **OPTIMAL**, the largest terminal gap is 1.4×10^{-7} for regular ISBP, 1.7×10^{-5} for non-regular ISBP, 10^{-4} for GKG, and 10^{-4} for GKGB. The best-response subproblems are small—a bounded knapsack or a min-cost flow over a single player’s strategy set—and are solved to optimality in milliseconds; the 10 s best-response time limit is never reached, and no best-response solve returns an infeasible status. A violation is declared when the incumbent’s regret exceeds a tolerance of 10^{-9} , uniformly across the four games. Finally, within ABS the cut pool stores the *generating deviations* rather than the inequality rows, and each pooled CEI is re-instantiated at the current value of α ; retention is therefore valid in both directions

of the bisection.

Performance metrics. For each instance group, we report the following metrics: *Count*: number of instances; *BRD*: number for which RRR-BRD found an exact PNE ($\alpha_{\text{brd}} = 1$); *GZR*: number for which GZR found a PNE; α_{brd} : average BRD approximation ratio; T_{BRD} and T_{GZR} : average wall-clock times (seconds) for BRD and GZR; *POS*: average price of stability, computed as the cost of the best PNE divided by the cost of the social optimum (SO); $\#_{\text{CEI}}$: average number of CEIs added during GZR; *OPT*, *INF*, *TL*: number of instances where GZR terminated with status **OPTIMAL**, **INFEASIBLE** (no PNE exists), or **TIME_LIMIT**, respectively. POS is averaged only over instances for which GZR returned a PNE (cells are left blank when none was found); α_{brd} is averaged over all instances, using the best end-of-round profile of each.

4.6.2 ISBP: Integer Splittable Bin Packing Games

Instance generation. We generate two families of ISBP instances following Hao and Michini [84]. *Regular instances* have identical item weights w and bin capacities u . We enumerate all feasible tuples (n, m, w, u) with $n \in \{4, \dots, 10\}$ players, $m \in \{n-2, \dots, n+3\}$ bins, $w \in \{4, \dots, 15\}$, and u derived from divisibility constraints with $mu - nw \geq 0$, yielding 841 instances. These conditions do not exclude $w > u$, and 215 of the 841 tuples have $w > u$. The player-anchoring constraint of Appendix 4.A requires $w \leq u$ and is therefore imposed only on the remaining 626 instances; bin-load ordering and the VEST cut carry no such restriction and are applied throughout.

Non-regular instances introduce 20% heterogeneity ($\delta = 0.2$) around the mean: item weights w_i and bin capacities u_j are drawn uniformly from $[\lceil \bar{w}(1 - \delta) \rceil, \lfloor \bar{w}(1 + \delta) \rfloor]$ and $[\lceil \bar{u}(1 - \delta) \rceil, \lfloor \bar{u}(1 + \delta) \rfloor]$, respectively. Of the 841 base tuples, 834 yield feasible non-regular instances.

Experiment 1: Best PNE and POS. For regular instances, RRR-BRD finds an exact PNE in 818 of 841 cases (97.3%), and GZR finds a PNE for 819 (including one instance where GZR finds a PNE that BRD missed). GZR confirms social optimality of the best PNE for 817 instances. Among the 22 instances where no exact PNE was found, GZR certifies infeasibility (INFEASIBLE) for 15 and hits the time limit without confirming a PNE for 7; the remaining 2 time-limit instances returned a feasible PNE without an optimality certificate, which is why GZR (819) exceeds OPT (817). These 22 instances proceed to Experiment 2 (ABS). For non-regular instances, BRD finds a PNE in 833 of 834 cases, and GZR confirms a PNE for all 834 instances, with an average POS of 1.000. The average GZR time is 8.4 s for regular and 6.1 s for non-regular instances. Comprehensive results by (n, m) are given in Tables 4.6 and 4.7 in the appendix.

Experiment 2: ABS for non-PNE instances. All 22 instances entering Experiment 2 are regular ISBP instances; no non-regular instance enters ABS. The concentration of non-PNE instances in the regular family is expected: the symmetric structure of regular instances—identical item weights and bin capacities—creates situations where multiple players can endlessly cycle through equivalent deviations, preventing convergence to an exact PNE. Notably, the instance $(n, m, w, u) = (5, 4, 5, 8)$ is precisely the counterexample of Hao and Michini [84] showing non-existence of PNE in regular ISBP. With the extended weight range $w \in \{4, \dots, 15\}$, additional non-PNE instances emerge at larger n and m —particularly $(n, m) \in \{(7, 5), (7, 6), (8, 6), (9, 8), (10, 8)\}$ —where the increased item weights make the packing problems more tightly constrained.

Table 4.1 presents the ABS results. For each instance, we report the game parameters (n, m, w, u) , the Experiment 1 GZR status ($\alpha = 1$), the BRD-computed initial upper bound α_{init} , the verified upper bound α^{UB} , the verified lower bound α^{LB} , verification status, the number of bisection iterations, and the total ABS wall-clock time. Here “Iters” counts

bisection steps only, while T_{ABS} is end-to-end wall-clock time and additionally includes the RRR-BRD warm start, the initial $\alpha = 1$ feasibility check, and the final lower-bound GZR call; T_{ABS} may therefore exceed $\text{Iters} \times 600\text{s}$. Here, α^{UB} is the smallest α for which GZR finds an α -PNE, and α^{LB} is the largest α for which GZR certifies infeasibility (no α -PNE exists). An instance is *verified* when both α^{UB} and α^{LB} are obtained and the gap satisfies $\alpha^{\text{UB}} - \alpha^{\text{LB}} \leq \varepsilon$ with the common verification threshold $\varepsilon = 10^{-2}$ of Section 4.6.1, providing a rigorous certificate that the tightest α -PNE lies in this interval. Unverified instances have a valid upper bound but fail the gap criterion, in all but one case because the lower-bound GZR call timed out without producing an infeasibility certificate. This asymmetry is expected: proving infeasibility (certifying that no α -PNE exists) is generally harder than finding an α -PNE.

Of the 22 instances, 7 are fully verified. Of the remaining 15, seven lack a verified lower bound entirely—the GZR calls at the lower-bound side timed out, particularly at larger weight values ($w \geq 11$) where the GZR subproblems become computationally more demanding—and eight close to widths between 0.013 and 0.037, outside the verification threshold. The asymmetry is pronounced here because the VEST cut and the player-anchoring constraint are valid only for exact equilibria and are therefore dropped once α exceeds one, leaving the lower-bound calls without the pruning available at $\alpha = 1$. The tightest approximation ratios range from $\alpha^{\text{UB}} = 1.009$ (very close to exact PNE) to $\alpha^{\text{UB}} = 1.081$ (moderate deviation), with most values below 1.05.

ABS typically converges in 5–7 bisection iterations. Wall-clock times range from 4.9s to 4,222s, and four instances exceed one hour: two at $(n, m) = (9, 8)$, including the longest at 4,222s, and one each at $(8, 6)$ and $(10, 8)$. These results demonstrate that even when no exact PNE exists, high-quality approximate equilibria are attainable.

Table 4.1: ISBP Experiment 2: ABS results for non-PNE instances. The Verified column is evaluated at the common threshold $\varepsilon = 10^{-2}$ of Section 4.6.1.

Type	n	m	w	u	$\alpha=1$	α_{init}	α^{UB}	α^{LB}	Verified	Iters	T_{ABS}
Reg	5	4	5	8	INF	1.081	1.081	1.080	TRUE	7	7.8
Reg	5	4	7	11	INF	1.055	1.055	1.054	TRUE	6	30.6
Reg	5	4	8	13	INF	1.106	1.015	1.014	TRUE	7	31.4
Reg	5	4	9	14	INF	1.041	1.034	1.034	TRUE	6	45.7
Reg	5	4	10	16	INF	1.067	1.009	1.009	TRUE	7	91.0
Reg	5	4	11	17	INF	1.033	1.023	1.022	TRUE	6	246.9
Reg	5	4	13	20	INF	1.028	1.016	1.016	TRUE	5	605.3
Reg	7	5	11	19	INF	1.018	1.013	1.000	FALSE	5	2089.8
Reg	7	6	8	11	INF	1.073	1.037	1.000	FALSE	7	3629.1
Reg	7	6	11	15	INF	1.025	1.013	1.000	FALSE	5	2470.8
Reg	7	6	13	18	TL	1.047	1.020	-	FALSE	6	3877.5
Reg	7	6	14	19	INF	1.039	1.019	1.000	FALSE	6	2953.6
Reg	8	6	7	11	INF	1.055	1.032	1.000	FALSE	6	1372.3
Reg	8	6	9	14	INF	1.041	1.021	1.000	FALSE	6	3111.9
Reg	8	6	11	17	INF	1.033	1.029	1.000	FALSE	6	3277.5
Reg	8	6	12	19	TL	1.065	1.056	-	FALSE	7	3296.0
Reg	8	6	13	20	TL	1.028	1.021	-	FALSE	5	3539.9
Reg	8	6	14	22	INF	1.055	1.021	1.000	FALSE	6	3047.1
Reg	8	6	15	23	TL	1.024	1.024	-	FALSE	5	3620.5
Reg	9	8	11	14	TL	1.041	1.040	-	FALSE	6	3719.9
Reg	9	8	15	19	TL	1.044	1.044	-	FALSE	6	4233.0
Reg	10	8	12	17	TL	1.026	1.026	-	FALSE	5	3628.3

4.6.3 NFG: Network Formation Games

Instance generation. We generate 128 NFG instances on layered directed networks. Each network has $L \in \{3, 5, 8, 10\}$ layers with $W \in \{3, 5, 8, 10\}$ nodes per layer. Inter-layer edges are drawn independently with probability $p = 0.8$, with at least one edge per node guaranteed to ensure full connectivity from the source layer to the sink layer. Edge costs are drawn as $c_e \sim U\{1, 20\}$. Each of $n \in \{2, 4, 6, 8\}$ players routes integer-splittable flow from a common source in the first layer to a common sink in the last layer, with individual demands $d_i \sim U\{2, 5\}$. Edge capacities are set relative to the total demand $D = \sum_i d_i$ in two regimes: *tight*, where $\kappa_e \sim U[\max(d_{\max}, \lceil 0.7D \rceil), \lceil 0.9D \rceil]$, forcing competition for edge space and

strong GNEP coupling; and *loose*, where $\kappa_e \sim U[\max(d_{\max}, \lceil 0.9D \rceil), \lceil 1.1D \rceil]$, providing mild coupling. If the resulting instance is infeasible (not all players can simultaneously route their demand), capacities are increased until feasibility is achieved. The full design is: 2 capacity levels $\times$ 4 player counts $\times$ 4 layer counts $\times$ 4 widths = 128 instances.

Experiment 1: Best PNE and POS. RRR-BRD finds an exact PNE ($\alpha_{\text{brd}} = 1$) for all 128 instances. GZR confirms the socially optimal PNE for 121 instances, while the remaining 7 hit the time limit but still return a feasible PNE. Under loose capacity, GZR proves optimality for 63 of 64 instances with an average POS of 1.005. Under tight capacity, GZR proves optimality for 58 of 64, with a slightly higher average POS of 1.014 and more CEIs (3.0 on average), reflecting the increased difficulty of tighter coupling constraints. Since all instances possess a PNE, Experiment 2 (ABS) is not needed.

Solve time and difficulty increase with the number of players: for $n = 2$, GZR terminates in about 1 second on average; for $n = 8$, average times reach 102 s (loose) and 172 s (tight), with several instances hitting the time limit. The POS remains close to 1 across all configurations¹, suggesting that the socially optimal PNE is typically very efficient for these network structures. Comprehensive results by capacity, player count, and layer count are given in Table 4.11 in the appendix.

4.6.4 GKG: Generalized Knapsack Games

Instance generation. We follow the Pisinger-like generation scheme of Duguet et al. [62]. For each player i and item j , weights are drawn independently as $w_{ij} \sim U\{1, R\}$ with $R = 1,000$. Profits depend on the *correlation type*: (i) *uncorrelated*: $p_{ij} \sim U\{1, R\}$,

¹For NFG the social-optimum reference is solved with a 1,200 s limit rather than the 600 s used elsewhere. Two instances still reach it; their reference is a feasible incumbent, so the POS reported for them is a lower bound on the true value.

independent of weights; (ii) *weakly correlated*: $p_{ij} \sim U\{\max(1, w_{ij} - R/10), w_{ij} + R/10\}$; (iii) *strongly correlated*: $p_{ij} = w_{ij} + R/10$ (deterministic). Each player's knapsack capacity is set proportionally to their total weight: $b_i = \max(1, \lfloor cf \cdot \sum_j w_{ij} \rfloor)$, where $cf \in \{0.2, 0.5\}$ is the *capacity factor*. A smaller cf yields tighter knapsacks, forcing players to be more selective and intensifying competition for shared items. Shared item capacities are drawn as $c_j \sim U\{1, n\}$.

We evaluate GKG in two settings. The first is the benchmark of Duguet et al. [62]: $n \in \{2, 3, 4\}$ players, $m \in \{5, 10, 15, 20, 30, 40, 50\}$ items, 10 replications per (n, m, cf, corr) combination, yielding $3 \times 7 \times 2 \times 3 \times 10 = 1,260$ instances. The second is a scalability study with $n \in \{2, 3, 5, 8, 10, 15, 20, 25, 30, 35, 40, 45, 50\}$ players, $m \in \{30, 50\}$ items, the same three correlation types, and 2 replications, yielding $13 \times 2 \times 3 \times 2 = 156$ instances; the capacity factor is not crossed here but assigned one value per replication, so that each (n, m, corr) cell contains one instance at $cf = 0.2$ and one at $cf = 0.5$.

Comparison with Duguet et al. [62]. Table 4.2 compares our GZR results with the method of Duguet et al. [62] on instances generated following their scheme. For each (n, m) pair, we report the number of instances (out of 60) for which GZR confirms the socially optimal PNE within a 10-minute time limit. GZR finds a PNE for all 1,260 instances (100%) and proves social optimality for 1,259 of them, the single exception hitting the time limit after having found a PNE within 0.03 s, whereas Duguet et al. [62] report termination rates of 30–50% under a one-hour time limit. The 30–50% rates are as reported in their paper, obtained with their implementation and hardware: their implementation is not publicly available, and the method—a custom branch-and-cut tree with an explicit geometric construction of intersection cuts—is nontrivial to reimplement faithfully, so we did not re-run it and the comparison is indicative rather than machine-to-machine. The advantage is driven by the global validity of CEIs (versus locally valid intersection cuts) and the ability

to leverage Gurobi’s solver for the master MILP.

Table 4.2: GKG benchmark: instances solved by CEI-GZR within 10 minutes (out of 60 per cell).

n	Items m							Solved (%)
	5	10	15	20	30	40	50	
2	60	60	60	60	60	60	60	100
3	60	60	60	60	60	60	60	100
4	60	60	60	60	60	60	60	100

Scalability study. To assess scalability beyond the small- n regime of Duguet et al. [62], we test on instances with up to 50 players. RRR-BRD finds an exact PNE ($\alpha_{\text{brd}} = 1$) for all 156 instances, confirming that GKG instances tend to possess PNE. GZR confirms the optimal PNE with POS = 1.000 across all correlation types. As expected, computational difficulty increases with correlation strength: average GZR time grows from 3.3s (uncorrelated) to 303.0s (strongly correlated), and the number of time-limit instances increases from 0 to 24. Comprehensive results by (n, m, corr) are given in Tables 4.8 and 4.9 in the appendix.

4.6.5 GKGB: Generalized Knapsack Games with Reciprocally Bilinear Coefficients

Instance generation. We extend the GKG benchmark to a generalized knapsack game by incorporating *reciprocally bilinear coefficients* from Dragotto and Scatamacchia [58], which we refer to as GKGB (GKG with bilinear coefficients). In addition to the standard knapsack profit terms, each player i ’s objective includes bilinear interaction terms $\sum_{k \neq i} \sum_j C_{kj}^i x_j^i x_j^k$ that capture the externality one player’s item selections impose on another’s payoff. We adopt Dragotto’s Type B and Type C datasets, which define interactions between $n \in \{2, 3\}$ players over $m \in \{25, 50, 75, 100\}$ items with budget ratios in $\{2, 5, 8\}$. Type B instances

have *positive* interaction coefficients $C_{kj}^i \sim U[1, 100]$, so that opponents' overlapping selections are mutually beneficial (synergistic effects). Type C instances have *mixed* coefficients $C_{kj}^i \sim U[-100, 100]$, introducing adversarial externalities. In addition, we introduce *Type BC* instances with $C_{kj}^i \sim U[-50, 100]$, a skewed-positive distribution that interpolates between the purely cooperative Type B and the fully adversarial Type C regimes. To convert each IPG into a GNEP, we add shared per-item capacity constraints $\sum_i x_j^i \leq c_j$ with $c_j \sim U\{1, n\}$ (seed 42), yielding 72 instances (24 per type).

In the master MILP, each bilinear term $x_j^i \cdot x_j^k$ is linearized via McCormick relaxation with auxiliary variable y_j^{ik} and constraints $y \leq x^i$, $y \leq x^k$, $y \geq x^i + x^k - 1$. The CEI formulation exploits the fact that, for a fixed incumbent profile $\hat{x}$, the bilinear terms in the best-response deviation collapse to terms linear in the opponents' variables x_j^k , so the resulting cut remains linear in the master decision variables.

Experiment 1: Best PNE and POS. Table 4.3 summarizes the results by interaction type. The three interaction types exhibit a clear difficulty ordering: $C > BC > B$. For Type B, BRD finds an exact PNE for all 24 instances and GZR confirms a PNE for all 24, proving optimality (best-welfare PNE) for 22; the remaining 2 hit the time limit but still return a feasible PNE. The average POS of 1.012—close to socially optimal—aligns with the intuition that positive interactions create cooperative incentives: players benefit from each other's selections, making equilibria easier to find and more efficient.

Type BC instances are considerably harder. BRD finds an exact PNE in 15 of 24 cases (62.5%), and GZR confirms a PNE for 16 (66.7%). The average GZR time rises to 386.4s, with an average of 4,111 CEIs, and 15 of 24 instances hit the time limit. The average POS of 1.057 reflects moderately less efficient equilibria compared to Type B. Even a partial introduction of negative interaction terms (roughly one-third of the coefficients are negative under $U[-50, 100]$) is sufficient to substantially increase computational difficulty.

Type C instances are the hardest. BRD finds an exact PNE in only 8 of 24 cases (33.3%), and GZR confirms a PNE for just 11 (45.8%). The average GZR time is 386.6 s (versus 68.0 s for Type B), the average number of CEIs is 4,016 (versus 295), and 15 of 24 instances hit the time limit. The average POS of 1.072 indicates the least efficient equilibria among the three types. These findings are consistent with Dragotto and Scatamacchia [58], who also observe that mixed-sign interactions greatly increase computational difficulty and can prevent PNE existence. Adversarial effects cause best-response dynamics to cycle without converging, enlarging the effective search space. The monotonic degradation from Type B through Type BC to Type C confirms that computational hardness scales with the proportion of adversarial externalities in the interaction coefficients.

Experiment 2: ABS for non-PNE instances. Table 4.4 presents ABS results for the 21 instances (8 Type BC and 13 Type C) where Experiment 1 failed to confirm a PNE. No Type B instances require ABS, reinforcing that adversarial externalities are the primary driver of equilibrium non-existence. The number of non-PNE instances increases monotonically from B (0) through BC (8) to C (13).

In ABS, BRD is re-run with more generous parameters (5 random restarts, 20 rounds per restart) compared to Experiment 1 (3 restarts, 15 rounds) in order to obtain a tighter initial upper bound α_{init} . For several instances this enhanced BRD already finds an exact PNE ($\alpha_{\text{init}} = 1$), confirming that these instances do admit a PNE—Experiment 1’s BRD simply had insufficient restarts to discover it. This occurs for 4 of the 8 Type BC and 1 of the 13 Type C ABS instances.

The verified upper bounds are modest, ranging from $\alpha^{\text{UB}} = 1.000$ to $\alpha^{\text{UB}} = 1.034$, indicating that even when exact PNE are not confirmed, very close approximations are available. Most instances are resolved in 0–2 bisection iterations, though the larger instances ($n = 3, m \geq 50$) can require up to 2,505 s. Of the 21 instances, 6 are fully verified. The remaining 15 lack

a verified lower bound because the lower-bound GZR calls timed out without producing an infeasibility certificate.

Table 4.3: GKGB summary.

Type	Count	BRD	GZR	α_{brd}	T_{BRD}	T_{GZR}	POS	$\#_{\text{CEI}}$	OPT	INF	TL
Type B	24	24	24	1.000	0.0	68.0	1.012	294.5	22	0	2
Type BC	24	15	16	1.002	0.2	386.4	1.057	4111.1	9	0	15
Type C	24	8	11	1.007	0.6	386.6	1.072	4015.7	8	1	15

Table 4.4: GKGB Experiment 2: ABS results for non-PNE instances.

Type	n	m	BG	$\alpha=1$	α_{init}	α^{UB}	α^{LB}	Verified	Iters	T_{ABS}
Type BC	2	75	2	TL	1.018	1.009	-	FALSE	1	1.0
Type BC	2	75	8	TL	1.000	1.000	1.000	TRUE	0	0.2
Type BC	2	100	5	TL	1.003	1.003	-	FALSE	0	1.0
Type BC	3	50	2	TL	1.022	1.011	-	FALSE	2	2504.6
Type BC	3	75	2	TL	1.000	1.000	1.000	TRUE	0	0.7
Type BC	3	100	2	TL	1.009	1.009	-	FALSE	0	1.2
Type BC	3	100	5	TL	1.000	1.000	1.000	TRUE	0	0.5
Type BC	3	100	8	TL	1.000	1.000	1.000	TRUE	0	0.7
Type C	2	50	2	INF	1.006	1.006	1.000	TRUE	0	0.9
Type C	2	75	2	TL	1.005	1.005	-	FALSE	0	0.9
Type C	2	75	5	TL	1.008	1.008	-	FALSE	0	0.9
Type C	2	100	5	TL	1.021	1.010	-	FALSE	2	1216.4
Type C	3	50	2	TL	1.058	1.022	-	FALSE	3	1285.1
Type C	3	50	5	TL	1.013	1.006	-	FALSE	1	466.7
Type C	3	50	8	TL	1.024	1.012	-	FALSE	2	1513.1
Type C	3	75	2	TL	1.024	1.012	-	FALSE	2	1532.2
Type C	3	75	5	TL	1.000	1.000	1.000	TRUE	0	0.3
Type C	3	75	8	TL	1.004	1.004	-	FALSE	0	2.5
Type C	3	100	2	TL	1.001	1.001	-	FALSE	0	1.5
Type C	3	100	5	TL	1.034	1.034	-	FALSE	2	2405.4
Type C	3	100	8	TL	1.006	1.006	-	FALSE	0	1.5

4.6.6 Impact of VEST Cuts

We now evaluate the computational impact of the VEST cuts introduced in Section 4.3.

Recall that the VEST cut exploits the vertex structure of each player's feasible region to

reduce the search space within GZR. For regular ISBP, where each player’s feasible region is a bounded simplex, the VEST cut provides a tight structural bound: at most one coordinate per player can take a non-extreme value at any PNE. For non-regular ISBP, where the uniformity hypothesis of Theorem 4.22 fails, the weaker bin-based cuts (4.12)–(4.13) apply; for NFG, whose conditional polytopes are flow polytopes, the weaker cuts are based on fundamental cycle bases. Both are valid but less restrictive. We compare GZR with and without these cuts on all three game classes. To eliminate execution-order bias, each variant is run in a separate kernel with an independent warm-start from RRR-BRD.

Table 4.5 reports the results. For each game class, we compare two configurations: “Cut on” (VEST or weak VEST enabled) and “Cut off” (no VEST cut), reporting the average GZR time, the number of instances solved to optimality, and two speedup metrics. The *aggregate speedup* is the ratio of average times, $\bar{T}_{\text{off}}/\bar{T}_{\text{on}}$, which reflects the overall computational savings and is dominated by harder instances. The *per-instance speedup* is the mean of individual ratios $T_{\text{off}}^{(k)}/T_{\text{on}}^{(k)}$ over instances solved to optimality by both configurations, giving equal weight to every instance. In both cases, values above 1 indicate that the cut is faster.

Table 4.5: Impact of VEST cuts: CEI-GZR with and without VEST, averaged over matched instances.

Game	Cut	Count	T_{GZR}	# _{CEI}	OPT	INF	TL	Speedup	
								Agg.	Per-inst.
ISBP Regular	VEST on	841	8.4	89.7	817	15	9	1.552	1.016
	VEST off	841	13.0	85.0	816	9	16		
ISBP Non-regular	wVEST on	834	6.1	19.7	831	0	3	0.888	1.002
	wVEST off	834	5.4	23.1	831	0	3		
NFG	wVEST on	128	66.1	1.8	121	0	7	0.923	0.865
	wVEST off	128	61.1	1.9	121	0	7		

VEST cut is effective for regular ISBP. For regular ISBP, enabling the VEST cut reduces the average GZR time from 13.0s to 8.4s, yielding an aggregate speedup of $1.55\times$.

The cut also improves robustness: with VEST enabled, only 9 instances hit the time limit versus 16 without, and GZR certifies infeasibility for 6 additional instances (15 versus 9). The per-instance speedup of 1.016 is more modest because the majority of instances are easy (solved in under 1 s) and show little timing difference; the aggregate speedup is driven by the harder instances where the cut provides meaningful pruning. This demonstrates that the tight structural bound of the VEST cut—limiting each player to at most one non-extreme coordinate—provides substantial acceleration for the bounded-simplex structure of regular ISBP, particularly on the most challenging instances.

Weak VEST cuts have negligible impact on non-regular ISBP and NFG. For non-regular ISBP and NFG, the weak VEST cuts—the bin-based constraints (4.12)–(4.13) for non-regular ISBP and the fundamental-cycle cuts for NFG—have no significant computational effect. On non-regular ISBP, both configurations solve 831 of 834 instances, with the same 3 instances hitting the time limit. The aggregate speedup of 0.888 indicates a slight disadvantage, while the per-instance speedup of 1.002 is essentially neutral—the two metrics differ because the weak cut adds modest overhead on the harder instances that dominate the average time. On NFG, both speedup metrics favor the no-cut configuration (0.923 aggregate, 0.865 per-instance), confirming that the weak cut provides no benefit. Both configurations solve the same 121 of 128 instances. This outcome is expected: the weak VEST bound provides valid but loose inequalities for general polytopes, and the overhead of generating and managing the additional constraints offsets their modest pruning power. These results confirm that the effectiveness of VEST cuts depends critically on the tightness of the underlying structural bound, with substantial benefits when the bound is sharp (regular ISBP) and negligible impact when it is loose (non-regular ISBP and NFG).

4.7 Conclusion

We developed a cutting-plane framework for computing exact, socially optimal, and approximate pure Nash equilibria of generalized integer programming games. The framework rests on conditional equilibrium inequalities: globally valid, closed-form linear inequalities that enforce the no-regret condition while an explicit activation mechanism accounts for deviations rendered infeasible by the coupling constraints. Embedding CEIs as lazy constraints yields the Generalized Zero-Regret algorithm, which optimizes over the set of (approximate) PNE, and the α -Bisection Search, which finds the tightest α -approximate PNE when exact equilibria do not exist. For games with strictly concave costs and integral conditional polytopes, the Vertex Equilibrium Structure Theorem shows that every PNE strategy is a vertex of the player’s conditional polytope, yielding structural cuts whose strength depends on the geometry of that polytope.

The computational study supports two main conclusions. First, global validity matters: on instances generated following the scheme of Duguet et al. [62], CEI-GZR solves all instances within the time limit, whereas the locally valid intersection-cut approach is reported to terminate on 30–50% of instances within one hour, and GZR further scales to 50 players and to the bilinear-cost extension GKGB. Second, the effectiveness of structural cuts tracks the tightness of the underlying bound: on regular ISBP, where the VEST cut limits each player to a single unsaturated bin, we observe a $1.55\times$ aggregate speedup, while the weaker cycle-based cuts for non-regular ISBP and NFG are valid but provide no measurable acceleration.

Several directions remain open, including structural cuts for convex-cost GIPGs, equilibrium-aware branching rules, sharper big- M -free encodings of the activation mechanism, the complexity of computing the optimal approximation factor α^* , and extensions to mixed-integer strategy spaces.

Appendix

4.A Symmetry-Breaking Techniques for ISBP

We employ three symmetry-breaking techniques to reduce the search space of the GZR algorithm for ISBP instances with identical bins or identical players. The three techniques are compatible when imposed simultaneously (together with the vertex cut (4.34) on regular instances): starting from any PNE, the transformation of Theorem 4.22 yields a cost-vector-preserving vertex PNE, and the relabelings of Theorems 4.27 and 4.29 permute identical bins and players, which preserves each player's number of unsaturated bins; hence a PNE satisfying all constraints at once exists whenever a PNE exists.

Technique (i): Bin-load ordering. Let $\mathcal{D}$ partition bins into groups of identical bins ($u_j = u_k, c_j = c_k$). Within each group $D = \{j_1, \dots, j_{|D|}\}$, impose nonincreasing total loads:

$$\sum_t t z_t^{j_\ell} \geq \sum_t t z_t^{j_{\ell+1}} \quad \forall \ell = 1, \dots, |D| - 1. \quad (4.43)$$

Theorem 4.27 (Validity of bin-load ordering). *For any PNE x of an ISBP instance, there exists a PNE y satisfying (4.43) for every identical-bin class $D \in \mathcal{D}$ such that $\pi^i(y^i, y^{-i}) = \pi^i(x^i, x^{-i})$ for each player i , obtained by relabeling bins within each identical-bin class.*

Proof. Let x be a PNE. For each identical-bin class $D \in \mathcal{D}$, reorder the bins in D in nonincreasing order of their total loads in x . Let $p : [m] \rightarrow [m]$ be the resulting permutation, fixing every bin outside the identical-bin classes, where $p(b)$ denotes the new position of bin b , so that within each class p^{-1} enumerates the bins in nonincreasing order of their loads in

x ; define

$$y_j^i := x_{p^{-1}(j)}^i \quad \forall i \in [n], j \in [m].$$

Because p only permutes bins with identical capacities and costs, we have $u_{p^{-1}(j)} = u_j$ and $c_{p^{-1}(j)} = c_j$ for every j . Moreover,

$$y_j = \sum_{i \in [n]} y_j^i = \sum_{i \in [n]} x_{p^{-1}(j)}^i = x_{p^{-1}(j)}.$$

Thus the demand constraints and the capacity constraints are preserved, so y is feasible. By construction, y satisfies the bin-load ordering constraints (4.43).

It remains to show that y is still a PNE. First, for each player i , using the usual convention that the cost contribution of an empty bin is zero,

$$\begin{aligned} \pi^i(y^i, y^{-i}) &= \sum_{j \in [m]} c_j \frac{y_j^i}{y_j} \\ &= \sum_{j \in [m]} c_j \frac{x_{p^{-1}(j)}^i}{x_{p^{-1}(j)}} \\ &= \sum_{j \in [m]} c_{p(j)} \frac{x_j^i}{x_j} \\ &= \sum_{j \in [m]} c_j \frac{x_j^i}{x_j} = \pi^i(x^i, x^{-i}), \end{aligned}$$

where the last equality uses that p permutes only identical bins.

Suppose, for contradiction, that y is not a PNE. Then some player i has a feasible profitable deviation $\bar{y}^i$ against y^{-i} :

$$\pi^i(\bar{y}^i, y^{-i}) < \pi^i(y^i, y^{-i}).$$

Define the corresponding deviation against x^{-i} by

$$\bar{x}_j^i := \bar{y}_{p(j)}^i \quad \forall j \in [m].$$

The demand constraint is preserved. For capacity feasibility, for every $j \in [m]$,

$$\bar{x}_j^i + x_j^{-i} = \bar{y}_{p(j)}^i + y_{p(j)}^{-i} \leq u_{p(j)} = u_j,$$

so $\bar{x}^i$ is feasible against x^{-i} . The same change of variables as above gives

$$\pi^i(\bar{x}^i, x^{-i}) = \pi^i(\bar{y}^i, y^{-i}) < \pi^i(y^i, y^{-i}) = \pi^i(x^i, x^{-i}),$$

contradicting that x is a PNE. Therefore y is a PNE, satisfies (4.43), and preserves every player's cost. $\square$

Technique (ii): Symmetry-generated equilibrium inequalities. For a best-response load vector $\hat{s}$ with support $\mathcal{B}_i \subseteq D$ in an identical-bin group D , we generate additional CEIs by permuting positive loads across bins in D . Each permuted deviation $\hat{s}'$ yields a globally valid equilibrium inequality of the form (4.11).

Remark 4.28 (Global validity of symmetric EIs). Since the bins in D are identical ($u_j = u_k$, $c_j = c_k$ for all $j, k \in D$), any permutation $\hat{s}'$ of the positive components of $\hat{s}$ across D is again a strategy in $\mathcal{X}_i$ (the demand and bound constraints are symmetric across D), so each permuted CEI is globally valid for all α -PNE states by Proposition 4.6; its activation term accounts for the profiles against which $\hat{s}'$ is feasible, and its coefficients evaluate the state-dependent cost of $\hat{s}'$.

Technique (iii): Player anchoring. For regular ISBP instances (identical bins and identical items) with $w \leq u$, we break player-label symmetry by requiring that designated players have positive allocation in designated bins. Let $k := \min\{n, \lceil nw/u \rceil\}$. We impose, in terms of the one-hot variables of Section 4.5.1.1, that player i places a positive number of units in bin i :

$$\sum_{t=1}^u \sum_{s=1}^{\min\{t,w\}} y_{i,s}^{i,t} \geq 1 \quad \forall i = 1, \dots, k. \quad (4.44)$$

Theorem 4.29 (Validity of player anchoring). *Consider a regular ISBP instance with common item size w , common bin capacity u , and $w \leq u$. Let $k := \min\{n, \lceil nw/u \rceil\}$. For every PNE x , there exists a PNE y satisfying both (4.43) and (4.44). Moreover, y is obtained from x by relabeling identical bins and identical players. Hence y has the same social cost as x , and the vector of individual player costs of y is a permutation of that of x .*

Since GZR optimizes the social objective, Theorem 4.29 shows that adding constraints (4.43) and (4.44) preserves the optimal value over the set of PNE. Individual player costs are only permuted, so these constraints would not be valid for a player-specific objective.

Proof. Let x be a PNE. By Theorem 4.27, we may relabel the identical bins to obtain a PNE $\bar{x}$ satisfying (4.43). This relabeling preserves feasibility and every player's cost.

Because the instance is regular, it is also uniform in the sense of Hao and Michini [84]—all bins have the same capacity and constant, load-independent costs. Hence, by Lemma 1 of Hao and Michini [84], $\bar{x}$ uses exactly $\lceil \frac{nw}{u} \rceil$ bins. Moreover, $w \leq u$ implies $\lceil \frac{nw}{u} \rceil \leq n$, so $k = \lceil nw/u \rceil$. Since $\bar{x}$ satisfies the bin-load ordering constraints, its nonempty bins are precisely $1, \dots, k$.

Consider the bipartite graph H whose left vertex set is $[k]$, whose right vertex set is $[n]$, and

in which bin j is adjacent to player i if and only if $\bar{x}_j^i > 0$. We verify Hall's condition:

$$|N_H(B)| \geq |B| \quad \forall B \subseteq [k],$$

where $N_H(B)$ denote the neighborhood of B in H . If $|N_H(B)| \leq |B| - 1$, then the total load in the bins of B is at most $|N_H(B)|w$, because only the players in $N_H(B)$ use those bins. The remaining $k - |B|$ nonempty bins have total capacity $(k - |B|)u$. Consequently,

$$\begin{aligned} nw &= \sum_{j \in B} \bar{x}_j + \sum_{j \in [k] \setminus B} \bar{x}_j \\ &\leq |N_H(B)|w + (k - |B|)u \\ &\leq (|B| - 1)w + (k - |B|)u \\ &\leq (|B| - 1)u + (k - |B|)u \\ &= (k - 1)u. \end{aligned}$$

This contradicts $k = \lceil nw/u \rceil$, which implies $(k - 1)u < nw$. Therefore,

$$|N_H(B)| \geq |B| \quad \forall B \subseteq [k].$$

By Hall's marriage theorem, there is an injection $\sigma : [k] \rightarrow [n]$ such that

$$\bar{x}_j^{\sigma(j)} > 0 \quad \forall j \in [k].$$

Extend σ to a permutation ρ of $[n]$ satisfying $\rho(j) = \sigma(j)$ for every $j \in [k]$ (here each bin index $j \in [k]$ is treated as a player index, which is possible because $k \leq n$), and relabel the players by defining

$$y^i := \bar{x}^{\rho(i)} \quad \forall i \in [n].$$

Since the players are identical, this relabeling preserves feasibility and the PNE property. It also leaves every bin load unchanged, so y continues to satisfy (4.43). Furthermore, for every $i \in [k]$,

$$y_i^i = \bar{x}_i^{\rho(i)} = \bar{x}_i^{\sigma(i)} > 0,$$

and hence y satisfies (4.44).

Thus y is obtained from x by relabeling identical bins and identical players. The bin relabeling preserves every player's cost, while the player relabeling merely permutes those costs. Therefore y has the same social cost as x , and its vector of individual player costs is a permutation of that of x . $\square$

4.B Comprehensive Computational Results

This appendix provides detailed per-configuration results for the four games studied in Section 4.6. Across the four games, the disaggregated results exhibit a consistent qualitative pattern: difficulty is driven by the strength of the coupling between players—larger item weights relative to bin capacity for ISBP, tighter edge capacities for NFG, stronger weight–profit correlation for GKG, and a larger share of adversarial interaction coefficients for GKGB—while the price of stability remains close to one whenever a PNE is found. Within each table, time-limit terminations concentrate in the most tightly coupled cells, whereas the remaining configurations are solved quickly and at social optimality. Tables 4.6 and 4.7 report ISBP results disaggregated by (n, m) for regular and non-regular instances, respectively. Tables 4.8 and 4.9 report GKG scalability results across $n = 2$ to $n = 50$ players, stratified by the number of items m and weight–profit correlation level (uncorrelated, weakly correlated, strongly correlated). Table 4.10 reports GKGB results by interaction type (B, BC, C) and problem size. Table 4.11 reports NFG results by capacity level (loose, tight), number of players, and

number of layers. All column definitions follow the performance metrics described in Section 4.6: *Count* is the number of instances; *BRD* and *GZR* are the numbers of instances for which each method found a PNE; α_{brd} is the average BRD approximation ratio; T_{BRD} and T_{GZR} are average wall-clock times in seconds; *POS* is the average price of stability; $\#_{\text{CEI}}$ is the average number of CEIs added; and *OPT*, *INF*, *TL* are the numbers of instances where GZR terminated with status **OPTIMAL**, **INFEASIBLE**, or **TIME_LIMIT**, respectively.

Table 4.6: Comprehensive results for ISBP regular instances by (n, m) .

Type	n	m	Count	BRD	GZR	α_{brd}	T_{BRD}	T_{GZR}	POS	$\#_{\text{CEI}}$	OPT	INF	TL
Reg	4	2	47	47	47	1.000	0.03	0.1	1.000	0.0	47	0	0
Reg	4	3	45	45	45	1.000	0.03	0.1	1.000	0.0	45	0	0
Reg	4	4	39	39	39	1.000	0.03	0.1	1.000	0.0	39	0	0
Reg	4	5	18	18	18	1.000	0.05	0.3	1.000	0.0	18	0	0
Reg	4	6	16	16	16	1.000	0.04	0.2	1.000	0.0	16	0	0
Reg	4	7	10	10	10	1.000	0.05	0.3	1.000	37.4	10	0	0
Reg	5	3	47	47	47	1.000	0.04	0.1	1.000	0.0	47	0	0
Reg	5	4	40	33	33	1.010	0.21	0.8	1.000	14.7	33	7	0
Reg	5	5	33	33	33	1.000	0.05	0.2	1.000	0.0	33	0	0
Reg	5	6	14	14	14	1.000	0.07	0.6	1.000	0.0	14	0	0
Reg	5	7	14	14	14	1.000	0.07	0.4	1.000	25.1	14	0	0
Reg	5	8	9	9	9	1.000	0.06	1.0	1.000	56.9	9	0	0
Reg	6	4	42	42	42	1.000	0.06	0.2	1.000	0.0	42	0	0
Reg	6	5	32	32	32	1.000	0.08	1.6	1.000	57.8	32	0	0
Reg	6	6	27	27	27	1.000	0.06	0.2	1.000	0.0	27	0	0
Reg	6	7	11	11	11	1.000	0.11	0.7	1.000	0.0	11	0	0
Reg	6	8	13	13	13	1.000	0.10	0.6	1.000	0.0	13	0	0
Reg	6	9	10	10	10	1.000	0.09	2.1	1.000	126.3	10	0	0
Reg	7	5	37	36	36	1.000	0.13	4.7	1.000	28.8	36	1	0
Reg	7	6	26	22	22	1.007	0.33	31.3	1.000	257.1	22	3	1
Reg	7	7	24	24	24	1.000	0.09	0.2	1.000	0.0	24	0	0
Reg	7	8	8	8	8	1.000	0.15	0.8	1.000	0.0	8	0	0
Reg	7	9	12	12	12	1.000	0.15	1.1	1.000	0.0	12	0	0
Reg	7	10	9	9	9	1.000	0.13	0.9	1.000	0.0	9	0	0
Reg	8	6	31	24	24	1.010	0.59	69.5	1.000	205.3	24	4	3
Reg	8	7	21	21	21	1.000	0.20	1.7	1.000	155.4	21	0	0
Reg	8	8	21	21	21	1.000	0.13	0.3	1.000	0.0	21	0	0
Reg	8	9	7	7	7	1.000	0.27	1.3	1.000	0.0	7	0	0
Reg	8	10	11	11	11	1.000	0.17	1.5	1.000	21.0	11	0	0
Reg	8	11	8	8	8	1.000	0.20	1.3	1.000	0.0	8	0	0
Reg	9	7	22	22	22	1.000	0.23	0.4	1.000	0.0	22	0	0
Reg	9	8	19	17	17	1.004	0.50	95.6	1.000	921.5	16	0	3
Reg	9	9	19	19	19	1.000	0.13	0.3	1.000	0.0	19	0	0
Reg	9	10	6	6	6	1.000	0.32	1.3	1.000	0.0	6	0	0
Reg	9	11	9	9	9	1.000	0.24	1.9	1.000	0.0	9	0	0
Reg	9	12	9	9	9	1.000	0.19	3.9	1.000	1.2	9	0	0
Reg	10	8	21	20	20	1.001	0.44	48.6	1.000	907.6	20	0	1
Reg	10	9	15	15	15	1.000	0.39	40.7	1.000	877.2	14	0	1
Reg	10	10	18	18	18	1.000	0.18	0.4	1.000	0.0	18	0	0
Reg	10	11	5	4	5	1.002	1.00	4.8	1.000	627.0	5	0	0
Reg	10	12	9	9	9	1.000	0.46	1.9	1.000	1.2	9	0	0
Reg	10	13	7	7	7	1.000	0.43	2.3	1.000	0.0	7	0	0

Table 4.7: Comprehensive results for ISBP non-regular instances by (n, m) .

Type	n	m	Count	BRD	GZR	α_{brd}	T_{BRD}	T_{GZR}	POS	$\#_{\text{CEI}}$	OPT	INF	TL
Non	4	2	40	40	40	1.000	0.0	0.1	1.000	0.1	40	0	0
Non	4	3	45	45	45	1.000	0.0	0.1	1.000	0.0	45	0	0
Non	4	4	39	39	39	1.000	0.0	0.1	1.000	0.1	39	0	0
Non	4	5	18	18	18	1.000	0.0	0.0	1.000	0.0	18	0	0
Non	4	6	16	16	16	1.000	0.0	0.1	1.000	0.0	16	0	0
Non	4	7	10	10	10	1.000	0.0	0.1	1.000	0.0	10	0	0
Non	5	3	47	47	47	1.000	0.0	0.1	1.000	0.0	47	0	0
Non	5	4	40	40	40	1.000	0.0	0.1	1.000	0.0	40	0	0
Non	5	5	33	33	33	1.000	0.0	0.2	1.000	0.5	33	0	0
Non	5	6	14	14	14	1.000	0.0	0.2	1.000	3.2	14	0	0
Non	5	7	14	14	14	1.000	0.0	0.3	1.000	0.0	14	0	0
Non	5	8	9	9	9	1.000	0.1	0.3	1.000	0.9	9	0	0
Non	6	4	42	42	42	1.000	0.0	0.3	1.000	0.1	42	0	0
Non	6	5	32	31	32	1.003	0.1	1.1	1.000	5.0	32	0	0
Non	6	6	27	27	27	1.000	0.1	1.0	1.000	2.7	27	0	0
Non	6	7	11	11	11	1.000	0.1	1.3	1.000	10.1	11	0	0
Non	6	8	13	13	13	1.000	0.1	0.8	1.000	3.8	13	0	0
Non	6	9	10	10	10	1.000	0.1	60.5	1.001	178.7	9	0	1
Non	7	5	37	37	37	1.000	0.1	1.8	1.000	5.7	37	0	0
Non	7	6	26	26	26	1.000	0.1	1.3	1.000	1.8	26	0	0
Non	7	7	24	24	24	1.000	0.1	2.0	1.000	13.9	24	0	0
Non	7	8	8	8	8	1.000	0.2	3.7	1.000	67.3	8	0	0
Non	7	9	12	12	12	1.000	0.2	27.9	1.000	199.3	12	0	0
Non	7	10	9	9	9	1.000	0.2	1.9	1.000	11.6	9	0	0
Non	8	6	31	31	31	1.000	0.1	3.1	1.000	5.7	31	0	0
Non	8	7	21	21	21	1.000	0.2	5.1	1.000	0.3	21	0	0
Non	8	8	21	21	21	1.000	0.2	4.1	1.000	20.0	21	0	0
Non	8	9	7	7	7	1.000	0.3	4.9	1.000	4.9	7	0	0
Non	8	10	11	11	11	1.000	0.2	111.4	1.003	518.0	9	0	2
Non	8	11	8	8	8	1.000	0.3	8.1	1.000	25.3	8	0	0
Non	9	7	22	22	22	1.000	0.2	12.2	1.000	9.3	22	0	0
Non	9	8	19	19	19	1.000	0.2	26.0	1.000	68.4	19	0	0
Non	9	9	19	19	19	1.000	0.3	8.3	1.000	13.0	19	0	0
Non	9	10	6	6	6	1.000	0.4	22.5	1.000	62.2	6	0	0
Non	9	11	9	9	9	1.000	0.2	1.3	1.000	0.1	9	0	0
Non	9	12	9	9	9	1.000	0.4	6.6	1.000	24.9	9	0	0
Non	10	8	21	21	21	1.000	0.2	12.8	1.000	24.6	21	0	0
Non	10	9	15	15	15	1.000	0.3	14.7	1.000	8.1	15	0	0
Non	10	10	18	18	18	1.000	0.4	18.3	1.000	51.1	18	0	0
Non	10	11	5	5	5	1.000	0.6	12.0	1.000	0.0	5	0	0
Non	10	12	9	9	9	1.000	0.5	15.0	1.000	8.6	9	0	0
Non	10	13	7	7	7	1.000	0.8	13.2	1.000	0.7	7	0	0

Table 4.8: Comprehensive GKG scalability results for $n = 2$ to $n = 20$.

n	m	Corr	Count	BRD	GZR	α_{brd}	T_{BRD}	T_{GZR}	POS	# _{CEI}	OPT	TL
2	30	uncorr	2	2	2	1.000	0.0	0.2	1.000	0.5	2	0
		weakly	2	2	2	1.000	0.0	0.6	1.000	1.0	2	0
		strongly	2	2	2	1.000	0.0	1.2	1.000	5.0	2	0
	50	uncorr	2	2	2	1.000	0.2	0.9	1.000	5.0	2	0
		weakly	2	2	2	1.000	0.3	2.5	1.000	7.5	2	0
		strongly	2	2	2	1.000	0.4	3.4	1.000	13.5	2	0
	30	uncorr	2	2	2	1.000	0.1	0.3	1.000	3.5	2	0
		weakly	2	2	2	1.000	0.1	2.4	1.000	6.5	2	0
		strongly	2	2	2	1.000	0.1	8.3	1.000	14.0	2	0
3	50	uncorr	2	2	2	1.000	0.1	1.5	1.000	14.0	2	0
		weakly	2	2	2	1.000	0.9	3.6	1.000	9.5	2	0
		strongly	2	2	2	1.000	1.1	14.7	1.000	19.5	2	0
	30	uncorr	2	2	2	1.000	0.1	1.1	1.000	10.0	2	0
		weakly	2	2	2	1.000	0.1	2.0	1.000	22.0	2	0
		strongly	2	2	2	1.000	0.1	80.3	1.000	51.0	2	0
	50	uncorr	2	2	2	1.000	0.2	1.0	1.000	13.0	2	0
		weakly	2	2	2	1.000	0.4	8.0	1.000	38.0	2	0
		strongly	2	2	2	1.000	0.8	307.3	1.000	79.0	1	1
5	30	uncorr	2	2	2	1.000	0.1	0.7	1.000	23.5	2	0
		weakly	2	2	2	1.000	0.0	1.1	1.000	20.5	2	0
		strongly	2	2	2	1.000	0.2	71.0	1.000	30.0	2	0
	50	uncorr	2	2	2	1.000	0.3	1.3	1.000	14.0	2	0
		weakly	2	2	2	1.000	0.4	33.0	1.000	246.0	2	0
		strongly	2	2	2	1.000	0.4	316.1	1.000	135.5	1	1
	30	uncorr	2	2	2	1.000	0.1	0.6	1.000	12.0	2	0
		weakly	2	2	2	1.000	0.1	2.7	1.000	64.0	2	0
		strongly	2	2	2	1.000	0.1	304.4	1.000	50.0	1	1
8	50	uncorr	2	2	2	1.000	0.2	2.4	1.000	43.5	2	0
		weakly	2	2	2	1.000	0.5	41.8	1.000	359.5	2	0
		strongly	2	2	2	1.000	0.4	379.2	1.000	67.5	1	1
	30	uncorr	2	2	2	1.000	0.3	1.4	1.000	19.0	2	0
		weakly	2	2	2	1.000	0.2	13.1	1.000	78.0	2	0
		strongly	2	2	2	1.000	0.5	306.0	1.000	22.0	1	1
	50	uncorr	2	2	2	1.000	0.4	3.3	1.000	28.0	2	0
		weakly	2	2	2	1.000	1.0	60.0	1.000	161.0	2	0
		strongly	2	2	2	1.000	1.0	600.1	1.000	219.0	0	2
10	30	uncorr	2	2	2	1.000	0.3	1.6	1.000	14.5	2	0
		weakly	2	2	2	1.000	0.2	26.9	1.000	185.0	2	0
		strongly	2	2	2	1.000	0.2	306.8	1.000	41.5	1	1
	50	uncorr	2	2	2	1.000	0.4	4.2	1.000	36.0	2	0
		weakly	2	2	2	1.000	0.7	69.4	1.000	158.0	2	0
		strongly	2	2	2	1.000	1.7	316.6	1.000	74.5	1	1

Table 4.9: Comprehensive GKG scalability results for $n = 25$ to $n = 50$.

n	m	Corr	Count	BRD	GZR	α_{brd}	T_{BRD}	T_{GZR}	POS	$\#_{\text{CEI}}$	OPT	TL
25	30	uncorr	2	2	2	1.000	0.3	3.6	1.000	25.5	2	0
		weakly	2	2	2	1.000	0.3	47.3	1.000	73.5	2	0
		strongly	2	2	2	1.000	0.3	308.6	1.000	45.5	1	1
	50	uncorr	2	2	2	1.000	0.8	4.0	1.000	40.5	2	0
		weakly	2	2	2	1.000	1.2	108.6	1.000	232.5	2	0
		strongly	2	2	2	1.000	0.9	600.1	1.000	326.5	0	2
	30	uncorr	2	2	2	1.000	0.3	3.5	1.000	28.5	2	0
		weakly	2	2	2	1.000	0.4	20.6	1.000	84.0	2	0
		strongly	2	2	2	1.000	0.5	320.8	1.000	91.5	1	1
30	50	uncorr	2	2	2	1.000	0.7	9.1	1.000	43.5	2	0
		weakly	2	2	2	1.000	1.2	207.8	1.000	110.5	2	0
		strongly	2	2	2	1.000	2.5	600.2	1.001	247.5	0	2
	30	uncorr	2	2	2	1.000	0.3	2.1	1.000	32.0	2	0
		weakly	2	2	2	1.000	0.2	19.1	1.000	151.0	2	0
		strongly	2	2	2	1.000	0.2	312.8	1.000	137.0	1	1
	50	uncorr	2	2	2	1.000	0.5	6.1	1.000	39.5	2	0
		weakly	2	2	2	1.000	1.5	316.1	1.003	403.5	1	1
		strongly	2	2	2	1.000	1.1	412.3	1.000	241.0	1	1
35	30	uncorr	2	2	2	1.000	0.3	4.5	1.000	46.0	2	0
		weakly	2	2	2	1.000	0.5	27.9	1.000	89.5	2	0
		strongly	2	2	2	1.000	0.4	318.9	1.000	134.5	1	1
	50	uncorr	2	2	2	1.000	0.7	4.9	1.000	49.0	2	0
		weakly	2	2	2	1.000	0.7	316.1	1.000	131.5	1	1
		strongly	2	2	2	1.000	1.7	600.2	1.000	327.0	0	2
	30	uncorr	2	2	2	1.000	0.4	4.8	1.000	46.0	2	0
		weakly	2	2	2	1.000	0.7	30.9	1.000	61.5	2	0
		strongly	2	2	2	1.000	0.6	341.3	1.000	193.0	1	1
40	50	uncorr	2	2	2	1.000	1.1	8.2	1.000	49.0	2	0
		weakly	2	2	2	1.000	2.0	324.1	1.000	180.0	1	1
		strongly	2	2	2	1.000	1.4	600.3	1.000	442.0	0	2
	30	uncorr	2	2	2	1.000	0.6	5.4	1.000	51.0	2	0
		weakly	2	2	2	1.000	0.6	32.5	1.000	61.0	2	0
		strongly	2	2	2	1.000	1.1	16.4	1.000	45.0	2	0
	50	uncorr	2	2	2	1.000	1.1	8.0	1.000	61.0	2	0
		weakly	2	2	2	1.000	1.2	162.7	1.000	155.0	2	0
		strongly	2	2	2	1.000	2.4	429.9	1.000	198.0	1	1

Table 4.10: Comprehensive GKGB results.

Type.	n	m	Count	BRD	GZR	α_{brd}	T_{BRD}	T_{GZR}	POS	$\#_{\text{CEI}}$	OPT	INF	TL
Type B	2	25	3	3	3	1.000	0.0	0.1	1.005	11.7	3	0	0
Type B	2	50	3	3	3	1.000	0.0	0.1	1.002	17.0	3	0	0
Type B	2	75	3	3	3	1.000	0.0	12.6	1.017	190.7	3	0	0
Type B	2	100	3	3	3	1.000	0.0	4.5	1.010	102.7	3	0	0
Type B	3	25	3	3	3	1.000	0.0	0.2	1.007	10.0	3	0	0
Type B	3	50	3	3	3	1.000	0.0	1.1	1.003	41.7	3	0	0
Type B	3	75	3	3	3	1.000	0.0	125.5	1.007	189.0	3	0	0
Type B	3	100	3	3	3	1.000	0.0	400.2	1.043	1793.7	1	0	2
Type BC	2	25	3	3	3	1.000	0.0	0.4	1.031	57.7	3	0	0
Type BC	2	50	3	3	3	1.000	0.0	88.0	1.029	2092.0	3	0	0
Type BC	2	75	3	1	1	1.004	0.4	600.5	1.061	9044.0	0	0	3
Type BC	2	100	3	2	2	1.000	0.2	600.5	1.090	7394.0	0	0	3
Type BC	3	25	3	2	3	1.004	0.2	1.1	1.029	55.0	3	0	0
Type BC	3	50	3	2	2	1.005	0.2	600.2	1.087	3992.0	0	0	3
Type BC	3	75	3	2	2	1.002	0.3	600.2	1.117	5164.3	0	0	3
Type BC	3	100	3	0	0	1.006	0.7	600.3		5089.7	0	0	3
Type C	2	25	3	2	3	1.005	0.1	0.7	1.027	52.0	3	0	0
Type C	2	50	3	1	2	1.002	0.3	256.6	1.091	5854.7	1	1	1
Type C	2	75	3	1	1	1.004	0.3	600.5	1.126	7742.0	0	0	3
Type C	2	100	3	1	2	1.003	0.5	417.1	1.091	6818.7	1	0	2
Type C	3	25	3	3	3	1.000	0.2	17.3	1.075	447.0	3	0	0
Type C	3	50	3	0	0	1.026	1.0	600.1		3482.7	0	0	3
Type C	3	75	3	0	0	1.009	1.2	600.3		3870.0	0	0	3
Type C	3	100	3	0	0	1.011	1.1	600.5		3858.7	0	0	3

Table 4.11: Comprehensive NFG results by capacity, number of players, and layers.

Cap.	n	Layers	Count	BRD	GZR	α_{brd}	T_{BRD}	T_{GZR}	POS	$\#_{\text{CEI}}$	OPT	INF	TL
Loose	2	3	4	4	4	1.000	0.1	0.1	1.000	0.0	4	0	0
Loose	2	5	4	4	4	1.000	0.1	0.4	1.000	0.0	4	0	0
Loose	2	8	4	4	4	1.000	0.1	0.6	1.000	0.0	4	0	0
Loose	2	10	4	4	4	1.000	0.1	1.0	1.000	0.2	4	0	0
Loose	4	3	4	4	4	1.000	0.1	0.2	1.000	0.0	4	0	0
Loose	4	5	4	4	4	1.000	0.2	1.1	1.015	0.8	4	0	0
Loose	4	8	4	4	4	1.000	0.2	9.3	1.005	0.8	4	0	0
Loose	4	10	4	4	4	1.000	0.2	7.1	1.000	0.2	4	0	0
Loose	6	3	4	4	4	1.000	0.1	0.4	1.006	0.5	4	0	0
Loose	6	5	4	4	4	1.000	0.2	2.9	1.004	0.2	4	0	0
Loose	6	8	4	4	4	1.000	0.3	79.3	1.013	3.8	4	0	0
Loose	6	10	4	4	4	1.000	0.4	86.6	1.000	1.0	4	0	0
Loose	8	3	4	4	4	1.000	0.1	0.7	1.021	0.8	4	0	0
Loose	8	5	4	4	4	1.000	0.3	4.1	1.000	0.0	4	0	0
Loose	8	8	4	4	4	1.000	0.5	145.9	1.004	1.0	4	0	0
Loose	8	10	4	4	4	1.000	0.6	258.3	1.014	2.2	3	0	1
Tight	2	3	4	4	4	1.000	0.0	0.1	1.000	0.2	4	0	0
Tight	2	5	4	4	4	1.000	0.1	0.3	1.078	0.2	4	0	0
Tight	2	8	4	4	4	1.000	0.1	2.9	1.020	0.8	4	0	0
Tight	2	10	4	4	4	1.000	0.2	1.1	1.000	0.2	4	0	0
Tight	4	3	4	4	4	1.000	0.1	0.2	1.000	0.0	4	0	0
Tight	4	5	4	4	4	1.000	0.2	1.1	1.025	1.0	4	0	0
Tight	4	8	4	4	4	1.000	0.2	28.0	1.000	0.5	4	0	0
Tight	4	10	4	4	4	1.000	0.2	235.3	1.007	2.5	4	0	0
Tight	6	3	4	4	4	1.000	0.1	0.4	1.006	0.5	4	0	0
Tight	6	5	4	4	4	1.000	0.2	3.0	1.004	0.8	4	0	0
Tight	6	8	4	4	4	1.000	0.5	305.0	1.017	17.8	3	0	1
Tight	6	10	4	4	4	1.000	0.6	254.5	1.003	11.8	3	0	1
Tight	8	3	4	4	4	1.000	0.1	0.7	1.021	0.8	4	0	0
Tight	8	5	4	4	4	1.000	0.3	5.1	1.000	0.0	4	0	0
Tight	8	8	4	4	4	1.000	0.5	210.7	1.014	6.8	3	0	1
Tight	8	10	4	4	4	1.000	0.7	469.6	1.050	3.8	1	0	3

Chapter 5

Cooperative Integer Programming Games: Core Stability and Optimal Coalition Structures

Manuscript Note. This chapter is based in part on the following manuscript:

H. Lee, R. Hildebrand, İ. E. Büyüктаhtakın, and J. Yang, “Cooperative Integer Programming Games: Core Stability and Optimal Coalition Structures.” Working paper. Peer reviewed and accepted for presentation at the 2026 INFORMS Optimization Society Conference, with the camera-ready version available at <https://zenodo.org/records/19380553>, and invited for expedited review at *INFORMS Journal on Optimization*.

The manuscript has been adapted and expanded for inclusion in this dissertation; notation, exposition, and organization have been revised for consistency with the remaining chapters. Code and data are available at <https://github.com/HyunwooLee0429/CIPG>.

5.1 Introduction

In many multi-agent systems, each player faces an individual resource-allocation problem naturally modeled as an integer program, say a knapsack or a capacity-constrained selection problem. Acting alone, a player may be unable to activate certain indivisible items or

projects because their budget is insufficient; by pooling budgets, a coalition can satisfy these integer requirements jointly and share the resulting value continuously. Municipalities funding indivisible infrastructure projects are a canonical example, and similar settings arise in joint procurement, collaborative R&D, and shared logistics. In each, what a group achieves depends on who joins it and not on what the players outside it do. That assumption is what makes a characteristic function the right object.

A *cooperative game* in characteristic function form is a pair (N, V) consisting of a finite player set N and a function $V : 2^N \rightarrow \mathbb{R}$ with $V(\emptyset) = 0$, where $V(c)$ is the value a coalition c can secure on its own [117]. The classical questions are which coalitions form and how they divide what they earn, and the central stability notion is the *Core* of c , the set of payoff vectors

$$\left\{ v \in \mathbb{R}^{|c|} : \sum_{i \in c} v_i = V(c), \quad \sum_{i \in s} v_i \geq V(s) \quad \forall s \subseteq c \right\}, \quad (5.1)$$

that is, payoffs that are *efficient* and *stable* against subcoalition deviation. Throughout we assume *transferable utility*: the worth $V(c)$ is denominated in a common divisible numéraire that the members of c may redistribute among themselves without loss, which is what makes (5.1) a statement about payoff vectors rather than about allocations of the underlying items. In the budget-pooling settings that motivate the model—joint procurement, co-funded infrastructure, shared licensing—the worth of a coalition is a monetary return that a consortium can split by contract, so transferability is the natural reading. Where side payments are not available, the coalition values computed here remain valid but the Core must be replaced by its non-transferable-utility counterpart.

The classical theory takes V as given. We introduce a variant in which it must be computed. A *cooperative integer programming game* (CIPG) is a cooperative game whose values are optimal values of integer programs, so each player's feasible plans live in a mixed-integer set and evaluating even one $V(c)$ can require solving an NP-hard problem. A second difficulty

is independent of the first: V has $2^{|N|} - 1$ entries, and one stability requirement refers to many at once.

Writing down $V(c)$ is itself a modeling decision. The program a player solves alone is given, but there is no canonical way to extend it to a coalition. Members' decisions may interact, their constraints may combine, and collaborating may open up plans that no member could have chosen alone. Each of these has to be settled and written into the constraints, and every answer produces a different V . We show how much rides on the choice: superadditivity and nonemptiness of the Core hold or fail with it, and so does the question of whether the grand coalition is simply optimal.

We propose a resource allocation setting in which those choices are made explicit. Certain resources combine when players collaborate, certain decisions are then taken jointly by the coalition, and the rest stay local to each player. Computing in this regime is our focus.

A *coalition structure* (CS) over c is a partition of c . Following Kóczy [95, Def. 3.30] we work with payoff-configurations, pairs $(v, CS(N))$ efficient within each block and individually rational, and add Core stability on each formed coalition [14]: $CS(N) = \{c_1, \dots, c_k\}$ with payoff vector v is an SCS if, for each c_ℓ , the restriction of v to c_ℓ lies in the Core of the game restricted to c_ℓ . This is weaker than the coalition structure core [95, Def. 3.31], which requires $\sum_{i \in s} v_i \geq V(s)$ for *every* $s \subseteq N$; Section 5.D shows that the weaker notion is what makes our question well posed. Because (5.1) is enforced *within* each coalition, an SCS guards against fission: no subcoalition has an incentive to break away. It does not guard against fusion, so separate coalitions may still benefit from merging [143]. We call any SCS that maximizes total payoff $\sum_{i \in N} v_i$ an OSCS, and our algorithms optimize over SCS-feasible solutions directly.

We investigate three questions: which structural properties of V can be established in CIPGs;

how to model and compute an *optimal coalition structure* (OCS); and how to characterize SCSs and *optimize* over them to obtain an OSCS.

5.1.1 Literature Review

Coalition-structure generation asks for a partition of N maximizing social welfare when the values $V(c)$ are already available, listed explicitly or produced by a simple formula [122, 123, 132]. A parallel literature equips games with coalition structures with their own stability notions [4, 6, 14, 95]; we adopt the coalition-structure stability of that line in Section 5.3.2.

Once the values are not given, the question becomes where they come from. The closest classical answer is the *linear production game* of Owen [113], in which players pool resource endowments and each coalition is worth the optimal value of a linear program; Granot [75] extends this to a generalized linear production model that unifies a broad class of cooperative games. In that setting the Core of the grand coalition is never empty: LP duality reads a Core allocation directly off the dual prices on the pooled resources, so the grand coalition is stable and optimal by construction.

CIPGs are the integer counterpart, and that guarantee fails: no dual construction survives integrality, and evaluating a single $V(c)$ is NP-hard. Optimal dual prices certify nothing once coalition-level decisions are integer, so the duality argument for Core nonemptiness has no analogue; evaluating a single $V(c)$ becomes NP-hard; and our experiments contain a full-pooling instance whose grand coalition admits no Core allocation (Section 5.5). The scope differs as well. Linear production games allocate within the grand coalition, whereas we optimize over coalition *structures* and enforce Core stability inside each formed coalition.

A second answer defines $V(c)$ through a fixed combinatorial problem on a graph or network,

which lets many Core-related questions be settled with specialized combinatorial structure [19, 40, 54]. The *integer minimization games* of Caprara and Letchford [32] subsume these, with $c(S)$ the value of an integer program parameterized by the coalition's incidence vector, and compute cost shares for the grand coalition; subadditivity is automatic there, and a CIPG's coalition policy can break its mirror, making the partition itself the decision (Remark 5.17 in the e-companion).

A third answer, and the one closest to practice, reads $V(c)$ off an operational optimization model. Göthe-Lundgren et al. [74] define $V(c)$ as the optimal value of a vehicle routing problem and compute Core and nucleolus quantities by constraint generation, evaluating $V(s)$ only when a violated inequality demands it; this is the separation-as-optimization idea we develop in Section 5.4.1. Such routing games need not be balanced, so their Core can be empty [41], which is itself an argument for optimizing over partitions instead of assuming the grand coalition. Özener et al. [114] allocate inventory-routing cost by Core-based dual methods with IP-defined values.

Closest to our OCS problem, Guajardo and Rönnqvist [77] propose OR models for choosing the coalition structure itself in collaborative logistics, assessing stability separately for each candidate structure. We instead embed the coalition-level IPs and the Core constraints in a single mixed-integer model with exact single-level MIP separation, which makes the partition a decision variable and enforces stability inside every formed coalition instead of verifying it afterwards. For broader surveys see Guajardo and Rönnqvist [78] and Gansterer and Hartl [72].

Optimization-based characteristic functions arise across application domains in which indivisibilities and player-specific integer decisions are essential: water resources [140], electricity transmission cost allocation [137], waste management [64], and shared energy storage [141].

For the knapsack setting in particular, the axiomatic literature studies a *single* knapsack held by the grand coalition and derives allocation rules and Core properties there [7, 51, 60], whereas the Cooperative Knapsack Game of Section 5.5 gives every player their own knapsack and every coalition a pooled one.

On the non-cooperative side, integer programming games model strategic interaction in which each player solves a mixed-integer program and payoffs couple through bilinear or shared-constraint structures, with pure Nash equilibria computed by zero-regret cutting planes and best-response algorithms [37, 58, 98]. These models form no coalitions.

Across these literatures the coalition values are given, LP-generated, graph-generated, read off an operational model, or priced for a fixed grand coalition. In a CIPG they are integer-programming values generated inside the search over coalition structures, which fuses three problems usually treated separately: computing $V(c)$ is NP-hard, choosing the partition is combinatorial, and certifying stability needs values for exponentially many subcoalitions. Sections 5.2 and 5.4 address the three together.

5.1.2 Contributions and Outline

We introduce *cooperative integer programming games* (CIPGs), the integer counterpart of the linear production games of Owen [113]. Integrality removes the dual-price construction that guarantees a nonempty Core in the linear case, and evaluating a single coalition value is already NP-hard.

- **Structure of the characteristic function.** We show that the coalition policy, not the CIPG framework, decides whether the grand coalition is optimal. Full pooling makes V superadditive (Proposition 5.2); restricted bounds break it and make the partition a decision worth making.

- **Two OCS formulations.** We give a mixed-integer formulation of the optimal coalition structure problem with an aggregated and a disaggregated linearization, and compare them theoretically and computationally. The disaggregated model leaves the integer problem unchanged but yields a relaxation that is never weaker and is strictly tighter on every instance we solve (Proposition 5.3).
- **Stability.** Gated stability inequalities enforce Core constraints inside each formed coalition. Theorem 5.5 shows that the resulting mixed-integer set projects exactly onto the stable payoffs, so maximizing over it returns an OSCS. A lifted variant dominates it pointwise (Proposition 5.8).
- **Computation and experiments.** We separate the inequalities without dualization, through a single-level MIP, and warm-start the solver with a primal heuristic that returns provably stable partitions. The heuristic is the largest most effective component: it accelerates the solves and makes the disaggregated formulation usable at scale. Section 5.5 instantiates the framework as a Cooperative Knapsack Game and reports the study.

The remainder of this chapter is organized as follows. Section 5.2 formalizes CIPGs and asks when the coalition structure is a decision worth making. Section 5.3 builds the mixed-integer model: the OCS formulation with its two linearizations, the stability inequalities and their lifted strengthening, and the exactness result that ties them together. Section 5.4 solves that model, covering separation oracles, the iterative and lazy-constraint architectures, the SCS-feasible warm start, and the per-coalition payoff refinement, summarized in Algorithm 5.14. Section 5.5 reports the computational study on cooperative knapsack games, including the comparison with an exhaustive baseline, and Section 5.6 concludes. Proofs omitted from the main text are in the e-companion (Section 5.B).

5.2 Cooperative Integer Programming Games

Two objects specify a CIPG: the problem each player solves alone, and the policy that turns a set of players into a single problem they solve together.

Definition 5.1 (Cooperative integer programming game). A *cooperative integer programming game* consists of a player set N , a vector of decision variables x^i for each player $i \in N$, and, for every coalition $c \subseteq N$, an objective f^c and a mixed-integer set $\mathcal{X}^c$ in the members' variables $x^c := (x^i)_{i \in c}$ together with auxiliary variables y^c that the coalition controls jointly, giving

$$V(c) = \max \{ f^c(x^c, y^c) : (x^c, y^c) \in \mathcal{X}^c \}, \quad V(\emptyset) = 0. \quad (5.2)$$

For a singleton this is player i 's own problem. We assume every program is feasible and bounded, so that V is finite and (N, V) is a cooperative game.

A coalition structure is itself a decision, so the model must be able to switch a player's decisions on and off according to whether that player joins a given coalition, and keeping each member's x^i identifiable is what makes that possible. The auxiliary variables y^c belong to the coalition and are not inherited when membership changes.

Taking f^c linear costs no generality as long as it is mixed-integer representable, since a piecewise-linear objective becomes linear once its epigraph is written out with auxiliary variables, which $\mathcal{X}^c$ is free to carry.

The definition becomes computationally interesting when the family $\{\mathcal{X}^c\}$ is described by a *coalition policy* rather than listed. Any one number is the optimum of some trivial integer program, so a listed family would admit every cooperative game with rational values, and an algorithm could do nothing but fill in $2^{|N|}$ entries one at a time. A policy instead builds $\mathcal{X}^c$ from the players in c , which is what lets $V(s)$ be reached for a subset never examined before.

It also carries the modeling decision described in Section 5.1: which constraints combine, which decisions the coalition takes jointly, what stays local to a player, and which plans only collaboration makes available. Different policies give different games on the same players.

One commitment is built into the definition. Since $\mathcal{X}^c$ depends on c alone, a coalition acts in isolation: what it achieves is unaffected by how the remaining players arrange themselves, and no resource is contested between two coalitions of the same partition. That places the game in characteristic function form rather than partition function form, and it is why a partition $CS(N)$ is worth exactly $\sum_{c \in CS(N)} V(c)$, the quantity Section 5.3.1 maximizes.

5.2.1 A resource allocation game

For the rest of this chapter we fix one coalition policy. Each player has a single budget constraint, which combines across a coalition, together with private constraints that never combine. Section 5.C of the e-companion returns to the general case.

For each player $i \in N$ and item $j \in R$, let $x_j^i \in \mathbb{Z}_+$ be the amount of item j chosen by player i , with profit coefficient $p_j^i \geq 0$ and common weight $a_j > 0$. Player i has budget b^i , a capacity u_j^i limiting how much of item j it may take, and possibly additional private constraints $B^i x^i \leq d^i$. When a coalition forms its members pool their budgets $\{b^i\}_{i \in c}$ and retain their private constraints, and a *pooling parameter* $\alpha \in (0, 1]$ caps the coalition's total use of each common item at α times the sum of its members' own capacities. Since α fixes V , it is part of the instance, not of any formulation of it; Section 5.2.2 examines what different values do.

Not every player has access to every item. For each item $j \in R$ let $I_j \subseteq N$ be the players

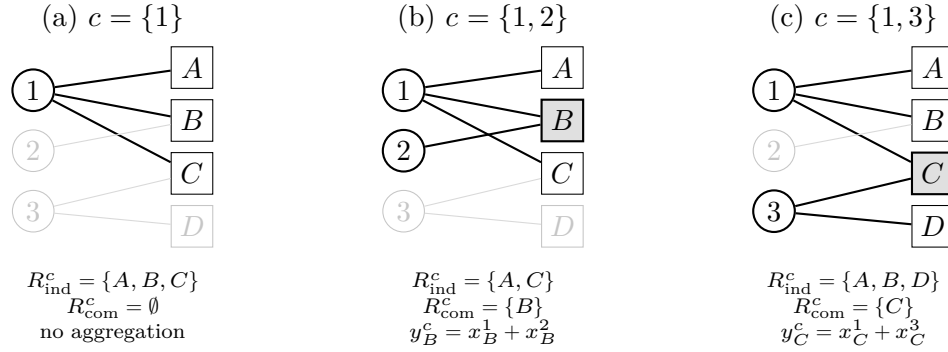


Figure 5.1: Availability and the coalition-dependent split of the item set. Circles are players, squares are items, and an edge means availability. Members of c are drawn in black, and shaded squares are the common items, reachable by two or more members. Item B is common in (b) but individual-specific in (c), and item C the other way round: the split depends on the coalition, not on the item. Note also that D is out of reach until player 3 joins.

for whom j is available. Given a coalition $c \subseteq N$,

$$\begin{aligned}
 I_j^c &:= I_j \cap c, & R^c &:= \{j \in R : I_j^c \neq \emptyset\}, \\
 R_{\text{ind}}^c &:= \{j \in R^c : |I_j^c| = 1\}, & R_{\text{com}}^c &:= \{j \in R^c : |I_j^c| \geq 2\},
 \end{aligned} \tag{5.3}$$

so R_{ind}^c collects the items available to exactly one member of c , which we call *individual-specific*, and R_{com}^c those available to two or more, which we call *common*.

This designation is coalition-dependent, as Figure 5.1 illustrates: an item available only to players 1 and 3 is individual-specific in coalition $\{1, 2\}$ and common in $\{1, 3\}$. The sets (5.3) are therefore recomputed for every coalition under consideration, in the OCS formulations, in the separation oracles, and in the experiments alike.

For common items we introduce coalition-level auxiliary variables

$$y_j^c = \sum_{i \in I_j^c} x_j^i, \quad j \in R_{\text{com}}^c, \tag{5.4}$$

which aggregate total usage of item j in coalition c . Integrality is placed on y_j^c and dropped

from the player-level x_j^i for $j \in R_{\text{com}}^c$, which reflects the two-stage nature of a coalition decision: the coalition first commits to how many units of each common item to activate, an integer choice recorded by y_j^c , and the resulting total is then divided among the contributing members, a continuous one recorded by $x_j^i \geq 0$ with $\sum_i x_j^i = y_j^c$. Items in R_{ind}^c keep their integrality, since one player decides them.

Dropping integrality on those x_j^i does not change $V(c)$. A common item's x_j^i appears only in the objective, in (5.5c) and in its own bound, so for fixed integer y_j^c the division among members is an interval problem whose optimum is attained at an integer point. This relies on (5.5d) not tying a member's common items to its other decisions.

The coalition value $V(c)$ is the optimal value of

$$V(c) = \max \sum_{j \in R_{\text{com}}^c} \sum_{i \in I_j^c} p_j^i x_j^i + \sum_{j \in R_{\text{ind}}^c} \sum_{i \in I_j^c} p_j^i x_j^i \quad (5.5a)$$

$$\text{s.t.} \quad \sum_{j \in R_{\text{com}}^c} a_j y_j^c + \sum_{j \in R_{\text{ind}}^c} \sum_{i \in I_j^c} a_j x_j^i \leq \sum_{i \in c} b^i, \quad (5.5b)$$

$$y_j^c = \sum_{i \in I_j^c} x_j^i, \quad \forall j \in R_{\text{com}}^c, \quad (5.5c)$$

$$B^i x^i \leq d^i, \quad \forall i \in c, \quad (5.5d)$$

$$0 \leq x_j^i \leq u_j^i, \quad \forall i \in c, j \in R^c, \quad (5.5e)$$

$$0 \leq y_j^c \leq \alpha \sum_{i \in I_j^c} u_j^i, \quad \forall j \in R_{\text{com}}^c, \quad (5.5f)$$

$$y_j^c \in \mathbb{Z}_+, \quad \forall j \in R_{\text{com}}^c, \quad x_j^i \in \mathbb{Z}_+, \quad \forall i \in c, j \in R_{\text{ind}}^c. \quad (5.5g)$$

Taking $c = \{i\}$ recovers the individual problem, since $|I_j \cap \{i\}| \leq 1$ for every j forces $R_{\text{com}}^{\{i\}} = \emptyset$. We write its value $V(\{i\})$ and use it throughout as the singleton value.

5.2.2 Structural properties of the characteristic function V

The coalition structure is a decision only if some partition can beat the grand coalition, and whether one ever does is settled here rather than by the CIPG framework. Under full pooling it never does, so the problem is vacuous; under restricted pooling it can, and that is the regime the rest of this chapter addresses. Two properties of V carry the argument, a third bears on how payoffs are divided.

Call V *superadditive* if $V(c \cup d) \geq V(c) + V(d)$ for all disjoint coalitions c, d , *subadditive* if the reverse inequality holds, and *convex* if $V(c) + V(d) \leq V(c \cup d) + V(c \cap d)$ for all c, d . Repeated merging shows that the grand coalition is an OCS in a superadditive game, and symmetrically that the singleton partition is an OCS in a subadditive one. Convexity is strictly stronger than superadditivity and matters later rather than here: on a convex game the Shapley value lies in the Core (Section 5.G.1 of the e-companion), so convexity would make Section 5.4.4 free.

(i) **Full pooling**, $\alpha = 1$. A coalition may then use the entire sum of its members' capacities on each common item, and (5.5f) is implied by (5.5c) and (5.5e), so it never binds on its own.

Proposition 5.2 (Superadditivity). *Assume full pooling, $\alpha = 1$. Then V is superadditive.*

Proof. Proof Let $c, d \subseteq N$ be disjoint and take optimal solutions (x^c, y^c) for $V(c)$ and (x^d, y^d) for $V(d)$. Define a candidate for $c \cup d$ by $\hat{x}_j^i = x_j^{c,i}$ if $i \in c$, $\hat{x}_j^i = x_j^{d,i}$ if $i \in d$, $\hat{x}_j^i = 0$ otherwise, and $\hat{y}_j^{c \cup d} = y_j^c + y_j^d$. Because c and d are disjoint, $0 \leq \hat{x}_j^i \leq u_j^i$ and $0 \leq \hat{y}_j^{c \cup d} \leq \sum_{i \in I_j^c} u_j^i + \sum_{i \in I_j^d} u_j^i = \sum_{i \in I_j^{c \cup d}} u_j^i$. The budget constraint for $c \cup d$ holds by summing the budgets of c and d , and every private constraint $B^i x^i \leq d^i$ is unaffected since each x^i is unchanged. The objective value of $(\hat{x}, \hat{y})$ is $V(c) + V(d)$, so $V(c \cup d) \geq V(c) + V(d)$. $\square$

Under full pooling Proposition 5.2 identifies the OCS *in advance*, so the full-pooling runs of Section 5.5 measure how quickly each formulation *certifies* an optimum rather than how quickly it finds one. It also explains why the lifted stability cut never falls back to the basic cut in that regime: superadditivity forces the synergy condition $V(s) \geq \sum_{i \in s} V(\{i\})$ for every subset (Section 5.3.3). Finally, it is the reason the restricted-bound setting of part (ii) exists at all: without it no nontrivial coalition structure could be optimal.

Superadditivity does not bring convexity with it. Capacity saturation makes marginal contributions shrink as coalitions grow: in a three-player instance with a single item too expensive for any one player, the second member of a coalition is worth the full item and the third is worth nothing, which breaks convexity while leaving superadditivity intact (Example 5.15 in the e-companion).

Non-convexity is therefore a feature of CIPGs rather than an artifact of our instances, and it is what makes the Shapley value leave the Core on the larger pooled coalitions of Section 5.5.

(ii) Restricted pooling, $\alpha < 1$. A coalition then cannot fully exploit that sum. No later result depends on the linear form: the stability inequalities, the separation and the heuristic need only that the cap depends on which members are present, so a different policy, in the extreme a binary activation $y_j^c \in \{0, 1\}$ for a shared resource, would change V but not the machinery. Such a bound arises whenever the limit attaches to the applicant rather than to the activity. Public funding programs cap awards per recipient, so two municipalities that could each win a grant separately receive a single award when they apply as one consortium, and that award is smaller than the two combined. Regulatory holding limits behave the same way: spectrum aggregation caps and airport slot rules bound what one licensee may hold, so carriers that merge or bid jointly face the cap once rather than twice. In each case cooperation is worth having for other reasons, but it costs the coalition part of what its members were separately entitled to.

Superadditivity now fails. With a binary common item and two players, each alone sets $x_1^i = 1$ and obtains value 1, while the coalition $\{1, 2\}$ faces a single shared unit $y_1^{\{1,2\}} \in \{0, 1\}$ and is worth only $1 < 1 + 1$. In this regime the OCS is no longer guaranteed to be the grand coalition; it becomes instance-dependent and nontrivial partitions can beat $\{N\}$, as assumed from Section 5.3.1 onward.

5.3 A Mixed-Integer Formulation

We now build a model whose solutions are exactly the stable coalition structures. It comes in two layers. Section 5.3.1 selects a partition and solves the coalition programs of every formed group, which gives an optimal coalition structure but enforces no stability; Sections 5.3.2 and 5.3.3 add a family of inequalities that cut off the partitions in which some subcoalition would rather deviate. Together the two layers describe the SCS payoffs exactly (Theorem 5.5), so an OSCS is obtained by maximizing over them.

5.3.1 OCS formulations: aggregated and disaggregated

We now formulate a mixed-integer program that selects a coalition structure and solves the coalition-level integer programs for each formed group. We reuse the item set R and the availability sets I_j from Section 5.2.1. Because the split of (5.3) into common and individual items depends on which coalition forms, we do not fix global sets R_{com} and R_{ind} ; the classification is determined endogenously by indicator variables.

Coalitions are represented by *group indices*. Since at most $|N|$ groups can form, we take $G := N$ as the index set and say that the players with $z_{ig} = 1$ form coalition $c(g)$. Table 5.4 in the e-companion collects the notation of the optimization models for reference; notation

local to the separation problem and to the heuristic is introduced where those are stated, in Sections 5.4.1 and 5.4.3.

The variables are group-indexed copies of those in (5.5): $x_j^{i,g} \geq 0$ and $y_j^g \in \mathbb{Z}_+$ replace x_j^i and y_j^c for the coalition realized in group g , with $z_{ig} \in \{0, 1\}$ recording the assignment, $w_j^g \in \{0, 1\}$ marking item j as common in group g , $\pi_g \geq 0$ the profit group g realizes, and $v_i \geq 0$ the payoff to player i . We write AGG_α for the *aggregated* formulation of the OCS_α problem: the subscript matters, since changing α changes V and so changes the problem, not just its formulation. Its constraints follow the order of (5.5), so the two can be read side by side.

objective, the partition, and profit accounting:

$$\max \quad \sum_{i \in N} v_i \quad (5.6a)$$

$$\text{s.t.} \quad \sum_{g \in G} z_{ig} = 1, \quad \forall i \in N, \quad (5.6b)$$

$$\pi_g = \sum_{j \in R} \sum_{i \in I_j} p_j^i x_j^{i,g} = \sum_{i \in N} v_i z_{ig}, \quad \forall g \in G, \quad (5.6c)$$

coalition program for each group, exactly (5.5b)–(5.5f):

$$\sum_{j \in R} a_j y_j^g \leq \sum_{i \in N} b^i z_{ig}, \quad \forall g \in G, \quad (5.6d)$$

$$y_j^g = \sum_{i \in I_j} x_j^{i,g}, \quad \forall g \in G, \quad \forall j \in R, \quad (5.6e)$$

$$B^i x_j^{i,g} \leq d^i z_{ig}, \quad \forall g \in G, \quad \forall i \in N, \quad (5.6f)$$

$$0 \leq x_j^{i,g} \leq u_j^i z_{ig}, \quad \forall g \in G, \quad \forall j \in R, \quad \forall i \in I_j, \quad (5.6g)$$

$$y_j^g \leq \sum_{i \in I_j} u_j^i z_{ig}, \quad \forall g \in G, \quad \forall j \in R, \quad (5.6h)$$

constraints that endogenous partitioning requires, with domains as in (5.5g):

$$z_{ig} = 0, \quad \forall g \in G, \forall i \in N \text{ with } i < g, \quad (5.6i)$$

$$\sum_{i \in N} z_{ig} \leq |N| z_{gg}, \quad \forall g \in G, \quad (5.6j)$$

$$v_i \geq 0, \quad y_j^g \in \mathbb{Z}_+, \quad z_{ig} \in \{0, 1\}, \quad \forall i \in N, \forall g \in G, \forall j \in R, \quad (5.6k)$$

needed only when $\alpha < 1$; omitted under full pooling:

$$2w_j^g \leq \sum_{i \in I_j} z_{ig} \leq (|I_j| - 1)w_j^g + 1, \quad \forall g \in G, \forall j \in R, \quad (5.6l)$$

$$y_j^g \leq \alpha \sum_{i \in I_j} u_j^i z_{ig} + M_j(1 - w_j^g), \quad \forall g \in G, \forall j \in R, \quad (5.6m)$$

$$w_j^g \in \{0, 1\}, \quad \forall g \in G, \forall j \in R. \quad (5.6n)$$

The objective (5.6a) maximizes total payoff. Constraint (5.6b) puts each player in exactly one group, so the nonempty groups induce a partition of N . The chain (5.6c) says that what a group produces is what it pays out: summing it over g and using (5.6b) gives $\sum_g \pi_g = \sum_i v_i$, so one could equivalently maximize total profit $\sum_{g \in G} \pi_g$. Since π_g is the objective value of a feasible, not necessarily optimal, production plan, it satisfies $\pi_g \leq V(c(g))$ for the realized coalition $c(g) = \{i \in N : z_{ig} = 1\}$, with equality at any optimal solution. Constraints (5.6d) to (5.6m), with the domains in (5.6k), embed for each group a coalition-level integer program consistent with (5.5), and in the same order: budgets of assigned members are pooled in (5.6d), items aggregate through (5.6e), private rows are preserved by (5.6f), usage is bounded by (5.6g), and (5.6m) carries the coalition policy, matching (5.5f) one for one. The indicators w_j^g it refers to are fixed by (5.6l) below.

The α -restriction (5.6m) carries the coalition policy into the formulation. It is released by a big- M term keyed to w_j^g because it should bind only on common items: an item reaching a single member is already capped by (5.6g). At $\alpha = 1$ it is redundant, since summing

(5.6g) over $i \in I_j$ with (5.6e) gives $y_j^g \leq \sum_{i \in I_j} u_j^i z_{ig}$ already, so one formulation covers both regimes. The final block of (5.6) collects everything this needs, so under full pooling it is simply omitted: our implementation does so, removing $|G| \cdot |R|$ binaries, which is 1500 at $n = 30$. What remains is then exactly (5.5) lifted to group indices, with (5.6h) in the role of (5.5f). That row is implied by (5.6e) and (5.6g) and is stated only to make the correspondence visible. The constant M_j has to be large enough to make (5.6m) redundant when $w_j^g = 0$. In that case the upper bound in (5.6l) leaves at most one member of I_j in the group, so y_j^g cannot exceed that member's own bound u_j^i by (5.6e) and (5.6g), and $M_j = (1-\alpha) \max_{i \in I_j} u_j^i$ suffices. Any larger value is also valid but only weakens the relaxation.

The two-sided bound (5.6l) makes $w_j^g = 1$ exactly when two or more players of I_j are assigned to group g : the upper bound forces $w_j^g = 1$ once a second player joins, and the lower bound forbids it before that, which is the endogenous version of the classification (5.3). Their purpose is to let the restricted bounds of Section 5.2.2 apply to common items only, which is where they mean anything: an item reaching a single member is already capped at u_j^i by (5.6g), and there is no sharing to restrict.

The parameter interpolates between the two regimes of Section 5.2.2. At $\alpha = 1$ the bound coincides with the full-pooling bound, so Proposition 5.2 applies, V is superadditive, and the grand coalition is an OCS. Smaller α is the restricted regime in which superadditivity fails and the partition becomes a decision worth making. Section 5.5 reports both, at $\alpha = 1$ and $\alpha = 0.5$.

Constraints (5.6i) and (5.6j) implement a group-leader symmetry-breaking scheme: group g contains only players with indices $i \geq g$, and player g belongs to group g whenever the group is nonempty, so each nonempty group is led by its smallest-indexed member and every coalition structure has a canonical representation.

Any optimal solution to (5.6) therefore yields an OCS.

Constraint (5.6c) involves the product $v_i z_{ig}$ and is bilinear. We linearize it with standard big- M techniques: for each pair (i, g) introduce $\phi_{ig} = v_i z_{ig}$ with $\phi_{ig} \leq v_i$, $\phi_{ig} \leq M z_{ig}$, $\phi_{ig} \geq v_i - M(1 - z_{ig})$, and $\phi_{ig} \geq 0$, where M bounds any payoff. Since a player's payoff cannot exceed the value of a coalition it belongs to, a computation-free bound is $M = \sum_{i \in N} \sum_{j \in R: i \in I_j} p_j^i u_j^i$. The right-hand side of (5.6c) becomes $\sum_{i \in N} \phi_{ig}$. We call this the *aggregated formulation*, or AGG, with LP relaxation F_{agg} .

5.3.1.1 Disaggregated formulation.

The aggregated formulation introduces $3|N| \cdot |G|$ McCormick constraints and relies on a big- M subtraction in $\phi_{ig} \geq v_i - M(1 - z_{ig})$, which weakens the LP relaxation. A *disaggregated* alternative, or DIS, removes every McCormick constraint. Disaggregating a variable controlled by a multiple-choice constraint and linking the copies is a standard reformulation technique [71, 79]; what is specific here is that it renders the McCormick system redundant outright, and the size of the effect measured in Section 5.5. Replace the single payoff variable v_i by *per-group* variables $v_{ig} \geq 0$, the payoff player i receives from group g . These are tied to the total, efficiency becomes purely linear, and the on/off structure is carried by a single variable upper bound:

$$\underbrace{v_i = \sum_{g \in G} v_{ig}}_{\text{linking}}, \forall i \in N; \quad \underbrace{\pi_g = \sum_{i \in N} v_{ig}}_{\text{efficiency}}, \forall g \in G; \quad \underbrace{v_{ig} \leq M z_{ig}}_{\text{upper bound}}, \forall i \in N, g \in G. \quad (5.7)$$

with the same computation-free bound M , which now appears only in the variable upper bound and never in a subtraction. The per-player quantity $U_i = \sum_{j: i \in I_j} p_j^i u_j^i$ would not do: a Core allocation may reward a player beyond its own profit potential, for instance one whose contribution is purely budget (Example 5.16 in the e-companion). The disaggregated model

replaces the second equality of (5.6c), and the McCormick linearization, by (5.7), keeping every other constraint of (5.6), including the α -restriction. We call it DIS_α , write F_{disagg} for its LP relaxation and, setting $\phi_{ig} := v_{ig}$, compare the two in the coordinates (v, ϕ, z, x, y, π) .

Proposition 5.3 (Same integer problem, tighter relaxation). *The aggregated and disaggregated formulations have the same mixed-integer feasible points. Their relaxations satisfy $F_{\text{disagg}} \subseteq F_{\text{agg}}$, and the inclusion can be strict. Moreover, adding the linking constraint $v_i = \sum_g \phi_{ig}$ to F_{agg} recovers F_{disagg} exactly.*

A byproduct of the proof (Section 5.B of the e-companion) is that under disaggregation every McCormick constraint is redundant, a net reduction of $2|N| \cdot |G| - |N|$ constraints. Strictness rests on a single fractional point, but the practical effect is larger: on all 30 instances of Section 5.5 the disaggregated root bound is strictly tighter, by margins between 45 and 98%.

5.3.2 Stability inequality for SCS

The OCS formulation (5.6) maximizes total coalition value over all partitions of N but does not enforce stability: a formed coalition may admit a blocking subcoalition. We now introduce a family of *stability inequalities* restricting (5.6) to *stable coalition structures* (SCS), in which every formed coalition is in the Core with respect to itself.

Each group index $g \in G$ corresponds to a potential coalition $c(g) := \{i \in N : z_{ig} = 1\}$. For each formed coalition, Core stability requires $\sum_{i \in s} v_i \geq V(s)$ for all $s \subseteq c(g)$. We encode these inequalities only when the players of s are together in some coalition, and we do so without coalition-indicator variables by gating on the assignment variables. For each subset

$s \subseteq N$ and each group $g \in G$ we impose the *stability inequality*

$$\sum_{i \in s} v_i \geq V(s) - M_s \left(|s| - \sum_{i \in s} z_{ig} \right), \quad (5.8)$$

where $M_s \geq \max\{V(s), 0\}$. The gate term $|s| - \sum_{i \in s} z_{ig}$ vanishes exactly when all of s is assigned to group g , in which case (5.8) is the Core inequality for s ; otherwise the right-hand side drops to at most zero and the inequality is inactive. Validity must hold for *every* pair (g, s) , including pairs with $s \not\subseteq c(g)$, because a cut persists in the model once generated.

Write $\mathcal{F}$ for the mixed-integer feasible set of AGG_α .

Proposition 5.4 (Validity and exact separation). *Let $s \subseteq N$ and $g \in G$, and suppose $M_s \geq \max\{V(s), 0\}$ and $V(\{i\}) \geq 0$ for all $i \in s$; both hold in our model with $M_s = V(s)$, since the empty solution is feasible and profits are nonnegative.*

- (i) (Validity) *Every mixed-integer point of $\mathcal{F}$ that encodes an SCS satisfies (5.8) for the pair (g, s) .*
- (ii) (Exact separation) *A mixed-integer point of $\mathcal{F}$ with $s \subseteq c(g)$ and $\sum_{i \in s} v_i < V(s)$ violates (5.8).*

Separation is itself NP-complete (Section 5.D), so no essentially cheaper oracle should be expected. Since the separation oracle computes $V(s)$ exactly, we set $M_s = V(s)$, the tightest choice in this big- M family when coalition values are nonnegative; Section 5.3.3 strengthens it further. We call (5.8) the *gated stability inequality*, the bracket being the gate, closed exactly when all of s lands in group g . Validity for every pair lets us impose it for all of them, so define

$$\mathcal{F}^{\text{stab}} := \{(v, x, y, z, \pi) \in \mathcal{F} : (5.8) \text{ holds for all } g \in G \text{ and all } s \subseteq N\}. \quad (5.9)$$

Let $\mathcal{S} \subseteq \mathbb{R}^N$ be the set of payoff vectors that admit some stable coalition structure: $v \in \mathcal{S}$ when there is a partition $CS(N)$ of N such that, for every $c \in CS(N)$, the restriction of v to c lies in the Core of the game restricted to c . Note that $\mathcal{S}$ is defined without reference to (5.6).

Theorem 5.5 (Exact extended formulation). *Under the assumptions of Proposition 5.4, the mixed-integer system $(v, x, y, z, \pi) \in \mathcal{F}^{\text{stab}}$ is an exact extended formulation for SCS payoffs:*

$$\text{proj}_v(\mathcal{F}^{\text{stab}}) = \mathcal{S}. \quad (5.10)$$

In particular, maximizing $\sum_{i \in N} v_i$ over $\mathcal{F}^{\text{stab}}$ yields an optimal stable coalition structure (OSCS).

Proof. Proof ($\subseteq$) Let $(v, x, y, z, \pi) \in \mathcal{F}^{\text{stab}}$ be mixed-integer feasible. The assignment variables define a partition $CS(N) = \{c(g) : \sum_i z_{ig} > 0\}$. Constraints (5.6d) to (5.6f) and the domains (5.6k) ensure feasibility of the coalition-level integer programs, with (5.6m) supplying the coalition bound (5.5f), so (5.6c) gives $\sum_{i \in c(g)} v_i = \pi_g \leq V(c(g))$ whenever $c(g)$ is nonempty. By Proposition 5.4(ii) the family (5.8) implies $\sum_{i \in s} v_i \geq V(s)$ for all $s \subseteq c(g)$. Taking $s = c(g)$ supplies the reverse inequality, so each formed coalition is efficient as well as internally stable, and $v \in \mathcal{S}$.

($\supseteq$) Conversely, let $v \in \mathcal{S}$, with coalition structure $CS(N)$ and coalition decisions x, y, π that are efficient and internally stable. Encode the structure canonically by assigning each coalition the group index of its smallest member, which respects (5.6i) and (5.6j). The resulting point satisfies all OCS constraints, including the nonnegativity in (5.6k): internal stability applied to the singleton $\{i\}$ gives $v_i \geq V(\{i\}) \geq 0$, the last inequality because the empty plan is feasible for player i . By internal stability with Proposition 5.4(i) it satisfies (5.8) for every pair, so $v \in \text{proj}_v(\mathcal{F}^{\text{stab}})$. $\square$

Note that $\mathcal{S}$ need not be a single polyhedron in v -space; it is in general a union of Core polytopes corresponding to different partitions of N . The system $\mathcal{F}^{\text{stab}}$ is therefore an extended formulation for SCS payoffs rather than a direct polyhedral description in payoff space.

Proposition 5.6 (Existence of SCS). *Assume $V(c)$ is finite for every $c \subseteq N$. Then an SCS exists: the singleton partition $CS(N) = \{\{i\} : i \in N\}$ with payoffs $v_i = V(\{i\})$.*

Proof. Proof For each singleton coalition $\{i\}$, efficiency requires $v_i = V(\{i\})$ and internal stability requires $v_i \geq V(\{i\})$ for the only nonempty subcoalition. These are compatible, so the Core of the restricted game $(\{i\}, V_{\{i\}})$ is nonempty. Collecting these singletons yields an SCS. $\square$

By Proposition 5.6 the feasible region $\mathcal{F}^{\text{stab}}$ is nonempty, so by Theorem 5.5 solving (5.6) augmented with (5.8) optimizes over SCS-feasible solutions and returns an OSCS.

5.3.3 Lifted stability cuts

Strengthening a valid inequality by reading structure out of the problem itself is a recurring device in integer programming, and the cuts below follow that pattern. Dynamic-programming information has been used to derive inequalities for the capacitated lot-sizing problem [87], partial-objective structure has been used to lift inequalities and drive separation in its multi-item version [28], and scenario dominance has been used to tighten risk-averse multistage stochastic mixed-integer programs [22, 23]. Those inequalities address different problem classes and none encodes an equilibrium or stability condition, so they are not predecessors of what follows; they are cited as methodological context for the same move, which

here is to read the synergy structure of the characteristic function out of the singleton values and use it to lift the stability inequality.

The basic inequality (5.8) can be strengthened using the singleton values. For $s \subseteq c(g)$ write $\Delta(s) := V(s) - \sum_{i \in s} V(\{i\})$ for the *synergy* of s , the value gained from cooperation beyond what its members achieve alone. When $\Delta(s) \geq 0$ we replace (5.8) by the *lifted stability cut*

$$\sum_{i \in s} v_i \geq \Delta(s) \left(\sum_{i \in s} z_{ig} - |s| + 1 \right) + \sum_{i \in s} V(\{i\}) z_{ig}. \quad (5.11)$$

Proposition 5.7 (Validity of the lifted cut). *Let $s \subseteq N$ with $\Delta(s) \geq 0$ and $V(\{i\}) \geq 0$ for all $i \in s$. Then (5.11) is valid for every mixed-integer point of $\mathcal{F}^{\text{stab}}$.*

Proposition 5.8 (Pointwise dominance). *Under the hypotheses of Proposition 5.7 and with $M_s = V(s)$, the right-hand side of (5.11) is at least that of (5.8) at every point of the box $z_{ig} \in [0, 1]$, $i \in s$. Consequently (5.11) implies (5.8) as a constraint on (v, z) .*

Remark 5.9 (Dominance and limits of lifted cuts). When all players of s belong to group g , the lifted cut reduces to the Core inequality $\sum_{i \in s} v_i \geq V(s)$, matching (5.8). Elsewhere in the box, Proposition 5.8 shows (5.11) dominates (5.8) pointwise, so the lifted family is at least as strong.

This dominance should not be read as a tightening of the root LP relaxation. Inequalities of the payoff-stability form $\sum_{i \in s} v_i \geq f_s(z)$ bound the payoff variables from below while the objective of (5.6) maximizes total payoff, which is why one should not expect them to move the root bound; we confirm computationally in Section 5.5 that they do not. Any benefit of (5.11) over (5.8) is realized inside branch and bound, where the dominated inequality can be discarded. Validity requires $\Delta(s) \geq 0$ and $V(\{i\}) \geq 0$; when either fails, our implementation falls back to (5.8). By Proposition 5.2 the fallback never triggers under full pooling.

Table 5.1: The four OSCS configurations tested in Section 5.5.

Configuration	Cut management	Separation for $ c > \tau$	Warm start
cutplane	re-solve the model	enumeration (all violated)	—
lazy	branch and cut (lazy)	enumeration (first violated)	—
lazy_MIP	branch and cut (lazy)	separation MIP (most violated)	—
lazy_MIP_ws	branch and cut (lazy)	separation MIP (most violated)	SCS-feasible heuristic

Note. For coalitions with $|c| \leq \tau$, every configuration enumerates all subsets and adds all violated cuts; the column “Separation for $|c| > \tau$ ” is where the configurations differ. Each configuration is tested under both the AGG and DIS formulations.

Nothing in Sections 5.3.2 and 5.3.3 used the knapsack form of (5.5): those inequalities reference the instance only through the values $V(s)$ and $V(\{i\})$. The separation of Section 5.4.1 does use the rows, and requires the subset indicators to enter only through right-hand sides and variable bounds, with the player variables bounded. Section 5.C of the e-companion states the general MILP setting and checks each component.

5.4 Computing an OSCS

The formulation of Section 5.3 has exponentially many stability inequalities, one for every pair of a group and a subset of players. They cannot all be stated up front, so they must be *separated*: generated on demand at candidate solutions. Our methods combine choices along three axes, taken in turn below: the separation oracle (Section 5.4.1), when the model is re-solved (Section 5.4.2), and the warm start (Section 5.4.3). Table 5.1 lists the four resulting configurations, each run under both the AGG and DIS formulations of Section 5.3.1.

5.4.1 Separation strategies

Given an incumbent $(\bar{v}, \bar{z})$ with realized coalitions $c(g) = \{i \in N : \bar{z}_{ig} = 1\}$, the separation oracle must find a subset $s \subseteq c(g)$ with $\sum_{i \in s} \bar{v}_i < V(s)$, or prove that none exists. We use

two oracles, combined through a cardinality threshold τ .

Enumeration. For $|c(g)| \leq \tau$ we enumerate all $2^{|c(g)|} - 1$ nonempty subsets and test each. This is exponential in the coalition size but exact. We use $\tau = 9$ throughout, and the two considerations pull against each other: a larger threshold covers more of the low-cardinality cuts that stay active through the search, for reasons taken up at the end of this section, while each increment of τ doubles the number of subsets to scan. At $\tau = 9$ the $2^9 - 1 = 511$ subsets of a coalition are cheap to check against cached coalition values, and past that point the cost outruns the extra coverage.

Optimization-based separation. For a group g , separation seeks the most violated sub-coalition,

$$\max_{s \subseteq c(g)} \left\{ V(s) - \sum_{i \in s} \bar{v}_i \right\}. \quad (5.12)$$

Written through the value function this looks like one coalition program per subset. It is not: choosing the subset and solving its coalition program can be done in a single model. Introduce binaries $\sigma_i \in \{0, 1\}$ for $i \in c(g)$, with $\sigma_i = 1$ exactly when $i \in s$, and let them drive the right-hand sides and variable bounds of (5.5). Because σ never touches the objective of the coalition program, the selection and the production decisions can be optimized together.

Both oracles enumerate below the threshold and differ only above it. The configuration that keeps enumerating, in order of increasing cardinality, and stops at the first violated subset we call *lazy*; the one that solves (5.13) for the most violated subset we call *lazy_MIP*. Each adds a single cut per group, so *lazy* trades more rounds for cheaper callbacks.

Fix $c(g)$ and let $R_{\text{ind}}^{c(g)}$ and $R_{\text{com}}^{c(g)}$ be the sets of (5.3) evaluated at $c = c(g)$, and let $R_i := \{j \in R : i \in I_j\}$ be the items available to player i . The decision variables are the subset indicators σ_i , the player-level usage x_j^i , and the coalition-level usage y_j of common items.

Mirroring (5.5), integrality falls on x_j^i for individual items and on y_j for common ones, while x_j^i for common items stays continuous:

$$\max \quad \sum_{i \in c(g)} \sum_{j \in R_i} p_j^i x_j^i - \sum_{i \in c(g)} \bar{v}_i \sigma_i, \quad (5.13a)$$

$$\text{s.t.} \quad \sum_{j \in R_{\text{ind}}^{c(g)}} \sum_{i \in I_j \cap c(g)} a_j x_j^i + \sum_{j \in R_{\text{com}}^{c(g)}} a_j y_j \leq \sum_{i \in c(g)} b^i \sigma_i, \quad (5.13b)$$

$$y_j = \sum_{i \in I_j \cap c(g)} x_j^i, \quad \forall j \in R_{\text{com}}^{c(g)}, \quad (5.13c)$$

$$B^i x^i \leq d^i \sigma_i, \quad \forall i \in c(g), \quad (5.13d)$$

$$0 \leq x_j^i \leq u_j^i \sigma_i, \quad \forall i \in c(g), \forall j \in R_i, \quad (5.13e)$$

$$y_j \leq \sum_{i \in I_j \cap c(g)} u_j^i \sigma_i, \quad \forall j \in R_{\text{com}}^{c(g)}, \quad (5.13f)$$

$$x_j^i \in \mathbb{Z}_+, \quad \forall i \in c(g), j \in R_{\text{ind}}^{c(g)} \cap R_i; \quad y_j \in \mathbb{Z}_+, \quad \forall j \in R_{\text{com}}^{c(g)}; \quad \sigma_i \in \{0, 1\}, \quad \forall i \in c(g), \quad (5.13g)$$

needed only when $\alpha < 1$; omitted under full pooling:

$$2\kappa_j \leq \sum_{i \in I_j \cap c(g)} \sigma_i \leq (|I_j \cap c(g)| - 1) \kappa_j + 1, \quad \forall j \in R_{\text{com}}^{c(g)}, \quad (5.13h)$$

$$y_j \leq \alpha \sum_{i \in I_j \cap c(g)} u_j^i \sigma_i + M_j(1 - \kappa_j), \quad \forall j \in R_{\text{com}}^{c(g)}, \quad (5.13i)$$

$$\kappa_j \in \{0, 1\}, \quad \forall j \in R_{\text{com}}^{c(g)}. \quad (5.13j)$$

The separation MIP must evaluate $V(s)$ under the *same* coalition policy as the formulation, and the common and individual designation depends on the selected subset itself: an item $j \in R_{\text{com}}^{c(g)}$ is common in s only when two or more of its users are selected. That is what the indicator κ_j records in (5.13h) and (5.13i) then enforces; κ_j is the separation counterpart of w_j^g , and the two pairs of constraints mirror (5.6l) and (5.6m). Under full pooling (5.13i) is implied by (5.13f), so nothing reads κ_j and both (5.13h) and (5.13i) are dropped, as in the

formulation.

We keep $M_j = \sum_{i \in I_j \cap c(g)} u_j^i$ here and the analogous constant in (5.6m), although $(1 - \alpha) \max_i u_j^i$ is also valid and considerably smaller. Both choices leave the mixed-integer feasible set unchanged, so every optimum we report is unaffected; only the relaxation differs. The reported times are therefore conservative, since the tighter constant can only strengthen the bound.

Proposition 5.10 (Single-level separation). *Fix a group g and an incumbent $(\bar{v}, \bar{z})$, and assume the player variables are bounded above by u . Then (5.13) and the separation problem (5.12) have the same optimal value, and every optimal σ^* of (5.13) yields a maximizer $s^* = \{i : \sigma_i^* = 1\}$ of (5.12). Hence the optimal value is nonpositive exactly when no subset of $c(g)$ violates stability at $\bar{v}$.*

Proof. Proof Fix $\sigma \in \{0, 1\}^{c(g)}$ and let $s = \{i : \sigma_i = 1\}$. Since u is finite, (5.13e) forces $x^i = 0$ for every $i \notin s$, and (5.13d) then reads $B^i x^i \leq 0$, which $x^i = 0$ satisfies; for $i \in s$ both revert to the original bounds. Constraint (5.13b) pools exactly the budgets of s , and (5.13h) and (5.13i) make the common and individual designation the one induced by s . The remaining constraints are those of (5.5) for coalition s , so maximizing (5.13a) over (x, y) at this fixed σ gives $V(s) - \sum_{i \in s} \bar{v}_i$. Maximizing over σ as well therefore returns $\max_{s \subseteq c(g)} \{V(s) - \sum_{i \in s} \bar{v}_i\}$, which is (5.12); the maximizers correspond. The last claim follows because $s = \emptyset$ is feasible with objective zero. $\square$

The second part is what justifies accepting an incumbent: a nonpositive optimum is a *certificate* of internal stability, not a failure to find a cut. Its hypotheses are also exactly the conditions under which the framework extends beyond the resource allocation game (Section 5.C of the e-companion). One may additionally impose $\sum_{i \in c(g)} \sigma_i \geq 1$, since $s = \emptyset$ is feasible with objective zero and yields no cut. The full subset $s = c(g)$ must *not* be

excluded. Its inequality is the efficiency requirement $\sum_{i \in c(g)} v_i \geq V(c(g))$, and (5.6c) alone does not enforce it at a mixed-integer feasible point: it gives only $\sum_{i \in c(g)} v_i = \pi_g \leq V(c(g))$, since the group's production need not be optimal there. Separating $s = c(g)$ is what makes every accepted incumbent efficient, hence SCS-feasible. Our implementation omits it: the objective maximizes total payoff, which (5.6c) ties to production, so efficiency holds at any optimum; time-limit incumbents are verified directly instead.

Strength of the most violated cut. The cut built from the maximizer s^* is usually not the strongest cut for the rest of the search. A stability cut for s is non-vacuous only when its gate is closed, that is when all of s is assigned to one group, and the number of coalition structures in which that happens falls rapidly with $|s|$. Since the violation tends to grow with cardinality, the most violated subset is often large, and its cut rarely binds again later in the tree, whereas low-cardinality cuts such as $v_i \geq V(\{i\})$ are weaker at the incumbent but activate in far more configurations. This inverts classical cutting-plane practice, where a most violated cover inequality, strengthened by lifting, is valid for the whole polytope and can bind anywhere in the tree [50, 76]. Hence the hybrid: enumeration below τ , where low-cardinality cuts keep paying, and the separation MIP above it, where only the certificate is available.

Algorithm 5.11 SEPARATE($\bar{v}, \bar{z}$), run by the solver at an integer-feasible point

Require: incumbent $(\bar{v}, \bar{z})$; enumeration threshold τ (we use $\tau = 9$)

Ensure: lazy stability cuts for every violated subcoalition found

```

1: for each nonempty realized coalition  $c(g)$  do
2:   if  $|c(g)| \leq \tau$  then
3:     for every violated nonempty  $s \subseteq c(g)$ , add (5.11) as a lazy cut, or (5.8) where the lifted
       cut is invalid ▷ up to  $2^\tau - 1$  cuts
4:   else
5:     solve (5.13), built from  $c(g)$  and  $\bar{v}$ ; if its value is positive, add that same cut at its
       maximizer  $s^*$  ▷ one cut
6:
7:
```

5.4.2 Iterative re-solving versus lazy constraints

The baseline is the classical iterative cutting-plane loop, which we call *cutplane*: solve (5.6) to optimality, enumerate the subcoalitions of every formed group, add all violated stability cuts, and re-solve, stopping when no violation remains, at which point the incumbent is a provably stable optimum. Its drawback is equally classical. Every round discards the previous branch-and-bound tree, and separation happens only at optimal solutions, so the tree is rebuilt each round.

Our remaining configurations embed separation in the branch-and-bound tree using lazy constraints. Whenever an integer-feasible solution $(\bar{v}, \bar{z})$ is found, at a MIPSOL callback, we extract the implied coalition structure $CS(N) = \{c(g) : \bar{z}_{ig} = 1\}$, run Algorithm 5.11 for each $c(g)$, and add any violated stability inequalities through Gurobi’s lazy-constraint callback `cbLazy` [80]. For each violated subset s we add the lifted cut (5.11) when the synergy condition $\Delta(s) \geq 0$ holds and the basic cut (5.8) otherwise; since the lifted cut dominates the basic one wherever it is valid, this is our default throughout. Every accepted incumbent then satisfies the separated inequalities, and the search optimizes over stable coalition structures directly.

5.4.3 An SCS-feasible primal heuristic

The warm start we supply to `lazy_MIP` is *provably* SCS-feasible: every coalition it returns carries a verified nonempty Core before the partition is assembled. Algorithm 5.12 states the procedure, which takes the two phase budgets T_1 and T_2 as parameters; the values we use are reported in Section 5.5.

Two features are worth pointing out. First, the pool $\mathcal{P}$ collects *every* coalition that passes a Core check, not only the merges that are executed, so one pass leaves Phase 2 with far

more material than the greedy merge path alone. Second, the Core check splits at the same threshold τ used for separation. For $|c| \leq \tau$ it enumerates every proper nonempty subcoalition and tests (5.1) directly. For $|c| > \tau$ it generates rows instead: solve the Core LP over the subcoalitions found so far, call (5.13) at the resulting allocation, and add the most violated subcoalition, stopping when separation finds no violation, which certifies a nonempty Core, or when the LP goes infeasible, which certifies an empty one. If the budget expires before either, the check returns *inconclusive* and the merging loop treats it as a failure, so a coalition enters $\mathcal{P}$ only with a proof.

Phase 2 selects from the pool by solving the set partitioning problem

$$\max \sum_{c \in \mathcal{P}} V(c) \chi_c \quad \text{s.t.} \quad \sum_{c \in \mathcal{P}: i \in c} \chi_c = 1 \quad \forall i \in N, \quad \chi_c \in \{0, 1\}, \quad (5.14)$$

which is always feasible because $\mathcal{P}$ contains the singletons.

Algorithm 5.12 SCS-feasible warm start

Require: player set N , threshold τ , phase budgets T_1 and T_2

Ensure: coalition structure CS^* and payoffs $\tilde{v}$ encoding an SCS

- 1: evaluate $V(\{i\})$ for every $i \in N$
 - 2: $CS \leftarrow \{\{i\} : i \in N\}$; $\mathcal{P} \leftarrow \{\{i\} : i \in N\}$; $(CS^*, \tilde{v}) \leftarrow (CS, (V(\{i\}))_{i \in N})$ $\triangleright$ incumbent, already an SCS

 - 3: **Phase 1: Core-verified merging**, within budget T_1
 - 4: **while** $|CS| \geq 2$ and T_1 is not exhausted **do**
 - 5: $\mathcal{C} \leftarrow \{(c_1, c_2) \subseteq CS : V(c_1 \cup c_2) - V(c_1) - V(c_2) > 0\}$; **if** $\mathcal{C} = \emptyset$ **then break**
 - 6: **for** $(c_1, c_2) \in \mathcal{C}$ in order of decreasing gain **do**
 - 7: **if** $\text{CORENONEMPTY}(c_1 \cup c_2)$ returns **true** **then** $\mathcal{P} \leftarrow \mathcal{P} \cup \{c_1 \cup c_2\}$ $\triangleright$ kept whether or not the merge is taken
 - 8:
 - 9: **if** no candidate passed **then break**
 - 10: $(c_1^*, c_2^*) \leftarrow$ passing pair of largest gain; $CS \leftarrow (CS \setminus \{c_1^*, c_2^*\}) \cup \{c_1^* \cup c_2^*\}$
 - 11:

 - 12: **Phase 2: set partitioning**, within budget T_2
 - 13: $CS' \leftarrow$ best solution of (5.14) over $\mathcal{P}$ found within T_2 , seeded with the singleton partition
 - 14: **if** a Core allocation $\tilde{v}'$ is obtained for every $c \in CS'$ **then** $(CS^*, \tilde{v}) \leftarrow (CS', \tilde{v}')$ $\triangleright$ incumbent replaced only when complete **return** $(CS^*, \tilde{v})$
-

Proposition 5.13 (Warm start: feasibility and cost). *Algorithm 5.12 evaluates the $|N|$ singleton values during initialization. From the end of initialization onward it maintains a pair $(CS^*, \tilde{v})$ encoding an SCS, and returns that pair whenever it stops, for any budgets $T_1, T_2 \geq 0$. Beyond initialization it performs at most $\binom{|N|+1}{3}$ Core checks, and each check on a coalition of size at most τ evaluates at most $2^\tau - 2$ coalition values.*

The initialization cost cannot be avoided: Core membership constrains the singleton sub-coalitions of every formed coalition, so no output can be certified without $V(\{i\})$ for each i . Past that point the heuristic is *anytime*, and the budgets control how good the partition is rather than whether one is returned. The cost bound makes $O(|N|^3)$ Core checks, so $O(|N|^3 2^\tau)$ calls to an NP-hard oracle, polynomial in the number of players for fixed τ , against the $2^{|N|} - 1$ values the exhaustive baseline of Section 5.5.4 needs before it can begin. It is loose in practice: at $|N| = 30$ it permits 4,495 checks and the runs use 210 at $\alpha = 0.5$ and 774 at $\alpha = 1.0$.

Limits of the guarantee. If no positive-gain pair passes a Core check in the first round the heuristic returns the singleton partition, worth $\sum_i V(\{i\})$; we claim no approximation ratio, and none is available without further assumptions on V . The guarantee is conservative by construction, since an inconclusive Core check counts as a failure. It is also only as sharp as the LP tolerance: the near-degenerate Cores of Table 5.7 sit close to the boundary at which that tolerance decides the answer.

5.4.4 Per-coalition solution concepts and payoff refinement

The system $(v, x, y, z, \pi) \in \mathcal{F}^{\text{stab}}$ fixes the coalition structure but not the payoffs within a coalition: by construction of SCS the restricted game (c, V_c) with $V_c(s) := V(s)$ has

a nonempty Core, and any of its Core allocations may replace the one the solver returns without changing the structure or the total welfare. We therefore report, for every formed coalition, the Shapley value [136]; the nucleolus [134], which lies in the Core whenever it is nonempty and hence for every formed coalition, computed by the standard sequential-LP scheme [100]; and the least-core value $\varepsilon(c)$, whose sign certifies Core nonemptiness and whose magnitude measures the slack of the tightest Core constraint. Formed coalitions are small enough that this is cheap. Definitions are in Section 5.G, the computation in Section 5.F, and the results in Section 5.5.

5.4.5 Algorithm summary

Algorithm 5.14 assembles the pieces into `lazy_MIP_ws`. Its outer loop is ordinary branch and cut, and we state it only so that the order of operations is in one place; the content is in Algorithms 5.11 and 5.12. The formulation is an input, not a fixed choice: nothing in the separation or the warm start depends on how the payoffs are linearized, and Section 5.5.1 measures the difference between AGG and DIS. One step is easy to get wrong. Line 2 must relabel CS^* canonically, because the symmetry-breaking constraints (5.6i)–(5.6j) admit exactly one group labeling of each partition, so a warm start that labels coalitions any other way is infeasible and the solver discards it, losing the whole benefit of Algorithm 5.12. The remaining configurations of Table 5.1 drop the warm start, giving `lazy_MIP`, or replace the MIP separation in line 5 of Algorithm 5.11 with first-violation enumeration.

Algorithm 5.14 OSCS via lazy constraints with an SCS-feasible warm start (lazy_MIP_ws)

Require: CIPG instance with pooling parameter α ; an OCS_α formulation, AGG_α or DIS_α ; threshold τ ; time limit T ; phase budgets T_1 and T_2

Ensure: an OSCS, or the best SCS-feasible incumbent found

- 1: $(CS^*, \tilde{v}) \leftarrow$ Algorithm 5.12 with budgets T_1, T_2
 - 2: encode CS^* as $\tilde{z}$ canonically, giving each coalition the group index of its smallest member, as the symmetry-breaking scheme of Section 5.3.1 requires
 - 3: warm start the formulation at $(\tilde{v}, \tilde{z})$ and solve it with the time remaining, separating stability cuts with Algorithm 5.11 at every integer-feasible incumbent (MIPSOL) **return** best incumbent, an OSCS if optimality is certified
-

5.5 Experimental design and results

We instantiate the framework as a *Cooperative Knapsack Game* (CKG): each player solves an individual knapsack of the form (5.5) with $c = \{i\}$, and coalitions pool their budgets under the α -restriction (5.6m). Instances use $|R| = 50$ items, availability probability $q = 0.5$, profits $p_j^i \sim U\{10, 100\}$, weights $a_j \sim U\{5, 10\}$, capacities $b^i = \rho \sum_{j \in R_i} a_j$ with $\rho = 0.3$ where $R_i = \{j : i \in I_j\}$, and $u_j^i = 1$, so each player solves a 0–1 knapsack. There are no private constraints, so $B^i x^i \leq d^i$ is vacuous. We report both regimes, $\alpha \in \{0.5, 1.0\}$, one instance per pair. Every configuration starts from the grand coalition, which lazy_MIP_ws replaces with the heuristic’s partition; we checked coalition-level efficiency for every reported incumbent afterwards by re-solving $V(c)$.

The implementation, the instance generators, and the scripts that reproduce every table and figure in this chapter are available at <https://github.com/HyunwooLee0429/CIPG>. All runs used an Intel Core i9-13900F (24 cores) with 64 GB of RAM, in Python with Gurobi 13.0.0 [80] on up to 16 threads, and the enumeration threshold $\tau = 9$ throughout. For lazy_MIP_ws we give Algorithm 5.12 the budgets $T_1 = 2T/5$ and $T_2 = T/5$, leaving the solver at least $2T/5$ and more when the heuristic finishes early; we settled on this split because it worked well across our instances, and the guarantee of Proposition 5.13 holds for

any other choice. Every OSCS configuration uses the lifted cuts (5.11) by default and falls back to (5.8) when the synergy condition fails. The full grid ran in one session, and the lazy_MIP_ws configurations were re-run in a second session after a coalition value cache was shared between the heuristic and the solver, so timings are comparable across methods and formulations.

Experiment 1 covers $n = 2, 4, \dots, 20$ with $T = 1800$ seconds and tests five methods under both formulations: the OCS formulation (5.6) without stability constraints, as a baseline, and the four OSCS configurations of Table 5.1. Experiment 2 covers $n = 22, 24, \dots, 30$ with $T = 3600$ seconds and drops cutplane, for reasons given in Section 5.5.2. Throughout, “TL” marks a timeout and the adjacent Gap column gives the gap remaining at that point.

5.5.1 Does disaggregation pay?

For the OCS problem, yes, and by a wide margin at the root. Table 5.2 reports both on all 30 instances. Its “Better” column ranks the formulation with the smaller time when both certify, the certifying formulation when only one does, and the smaller reported gap when both time out, with “—” when the two coincide. Timeout rows report best incumbents rather than proven optima.

Root bounds. The disaggregated bound is tighter on every instance, and the margin grows from 45.6% at $n = 2$ to 98.1% at $n = 30$. The growth has a closed form. In F_{agg} the only upper bound on a payoff is $v_i \leq \phi_{ig} + M(1 - z_{ig})$, so at $z_{ig} = 1/n$ the relaxation admits $v_i \approx M(1 - 1/n)$ for every player. Payoffs decouple from production and $\text{LP}_{\text{AGG}} \approx (n-1)M$, which is quadratic in n because M grows linearly; the observed ratio between $n = 30$ and $n = 10$ is 9.79 against a predicted 9.67. The linking constraint in (5.7) removes exactly this mechanism: total payoff equals total fractional production, which the pooled resources

Table 5.2: OCS formulation comparison, aggregated (AGG) vs. disaggregated (DIS): integer performance and root LP bounds.

n	α	OCS (AGG)			OCS (DIS)			Better	Root LP bounds			
		Obj	Time	Gap	Obj	Time	Gap		LP _{AGG}	LP _{DIS}	Tight. (%)	IP gap (%)
2	0.5	1340	0.04	—	1340	0.01	—	DIS	2,531	1,376.5	45.6	2.72
2	1.0	1365	0.01	—	1365	0.00	—	DIS				0.84
4	0.5	3037	0.3	—	3037	0.1	—	DIS	13,795	3,114.0	77.4	2.54
4	1.0	3105	0.09	—	3105	0.06	—	DIS				0.29
6	0.5	3622	0.3	—	3622	0.4	—	AGG	29,319	3,699.9	87.4	2.15
6	1.0	3696	0.1	—	3696	0.09	—	DIS				0.10
8	0.5	5487	1.5	—	5487	1.4	—	DIS	61,197	5,614.7	90.8	2.33
8	1.0	5609	0.4	—	5609	0.2	—	DIS				0.10
10	0.5	7483	11	—	7483	4.1	—	DIS	110,032	7,567.9	93.1	1.13
10	1.0	7564	1.0	—	7564	0.6	—	DIS				0.05
12	0.5	8244	14	—	8244	12	—	DIS	150,727	8,331.4	94.5	1.06
12	1.0	8325	1.1	—	8325	3.4	—	AGG				0.08
14	0.5	8855	57	—	8855	43	—	DIS	193,826	8,954.8	95.4	1.13
14	1.0	8953	1.2	—	8953	0.3	—	DIS				0.02
16	0.5	10250	94	—	10250	113	—	AGG	266,031	10,344.9	96.1	0.93
16	1.0	10341	55	—	10341	TL	0.02	AGG				0.04
18	0.5	11434	TL	0.23	11425	TL	0.38	AGG	346,662	11,563.0	96.7	1.13
18	1.0	11560	TL	0.01	11560	TL	0.02	AGG				0.03
20	0.5	12906	TL	0.17	12896	TL	0.30	AGG	441,402	12,981.3	97.1	0.58
20	1.0	12978	TL	0.02	12978	TL	0.02	—				0.03
22	0.5	15135	TL	0.79	15135	TL	0.73	DIS	567,079	15,342.2	97.3	1.37
22	1.0	15341	3.3	—	15341	0.9	—	DIS				0.01
24	0.5	16374	TL	0.32	16353	TL	0.31	DIS	672,802	16,445.6	97.6	0.44
24	1.0	16439	TL	0.04	16439	TL	0.04	—				0.04
26	0.5	17906	TL	0.67	17896	TL	0.67	—	779,708	18,036.9	97.7	0.73
26	1.0	18033	TL	0.02	18033	TL	0.02	—				0.02
28	0.5	18768	TL	0.73	18760	TL	0.64	DIS	922,279	18,945.4	97.9	0.94
28	1.0	18941	TL	0.02	18941	TL	0.02	—				0.02
30	0.5	20708	TL	0.79	20716	TL	0.75	DIS	1,077,559	20,876.5	98.1	0.77
30	1.0	20876	6.8	—	20876	1.6	—	DIS				0.00

Note. Obj, wall-clock time (s), and MIP gap (%) at timeout (TL); “—” in a Gap column means solved to optimality. “Tight.” = $(\text{LP}_{\text{AGG}} - \text{LP}_{\text{DIS}}) / \text{LP}_{\text{AGG}}$; “IP gap” is the relative gap of LP_{DIS} to the best known integer OCS value of the row; root bounds are reported once per n . The “Better” convention and remaining details are given in the text. Time limit: 1800 s ($n \leq 20$); 3600 s ($n \geq 22$).

bound, so LP_{DIS} sits just above the integer value. In absolute terms it lies within 0.84% of the best known integer value at $\alpha = 1.0$ and within 2.72% at $\alpha = 0.5$, where the LP does not see the α -restriction, while the aggregated bound exceeds it by more than a factor of 50 at $n = 30$. The two α values share a root bound: the indicators w_j^g relax and the restriction goes inactive, so the table reports one per n (the disaggregated bounds differ by less than 0.06% between the two α values; the table reports $\alpha = 1.0$).

The margin measures the formulations as stated; a global bound on the objective closes most of it but is counterproductive inside branch and bound, as Section 5.E explains.

Adding all basic and lifted stability cuts for subsets of size up to two a priori leaves the root LP value unchanged on every instance, as argued in Remark 5.9.

Solve times. The runtime margins are far more modest than the bound margins. DIS is faster in 14 of the 17 instances where both formulations certify. At $\alpha = 1.0$ the α -restriction reduces to the full-pooling bound, so V is superadditive by Proposition 5.2 and the grand coalition is known to be optimal in advance; those rows therefore measure certification speed rather than search. DIS usually certifies faster, taking under two seconds at $n = 22$ and $n = 30$ where AGG takes several times as long. The exception is $n = 16$, where AGG certifies in 55 seconds and DIS times out. At $n \in \{18, 20, 24, 26, 28\}$ neither formulation certifies within the limit, both stalling at gaps between 0.01 and 0.04%: the last fraction of a percent is hard for both. Where both certify they agree.

The reversal for OSCS. Once stability constraints enter, the ranking inverts: within lazy_MIP, AGG is faster in 9 of the 15 cells where both certify (Table 5.8 in the e-companion). More seriously, without a warm start DIS collapses at scale: on seven of the eight configurations with $n \geq 24$, and already at $n = 20$, $\alpha = 0.5$, it returns near-trivial

singleton partitions with objectives of 303 to 375 against expected values of 12,900 to 20,900, leaving gaps above 4,000%. The mechanism is model size. Disaggregation adds $|N| \cdot |G|$ payoff variables and $|N|$ linking constraints, which with the exponentially many lazy cuts overwhelms the solver’s ability to find any feasible solution. The warm start removes that burden and the collapse never occurs, but AGG is the safer default.

Table 5.3: OSCS separation strategy comparison (AGG formulation).

n	α	cutplane				lazy				lazy_MIP				lazy_MIP_ws							
		Obj	Time	Gap	Cuts	Obj	Time	Gap	Cuts	Obj	Time	Gap	Cuts	Obj	Time	Gap	Cuts	Heur	HGap	CS	avg.
2	0.5	1340	0.01	—	0	1340	0.01	—	1	1340	0.01	—	1	1340	0.09	—	0	0.0	<0.01	2	1.0
2	1.0	1365	0.02	—	1	1365	0.02	—	2	1365	0.01	—	2	1365	0.01	—	0	0.0	<0.01	1	2.0
4	0.5	3037	0.8	—	2	3037	0.2	—	7	3037	0.2	—	7	3037	0.2	—	0	0.0	<0.01	3	1.3
4	1.0	3105	0.5	—	14	3105	0.1	—	14	3105	0.1	—	14	3105	0.1	—	0	0.0	<0.01	1	4.0
6	0.5	3622	1.0	—	4	3622	0.5	—	35	3622	0.5	—	35	3622	0.6	—	0	0.0	<0.01	4	1.5
6	1.0	3696	0.8	—	54	3696	0.4	—	54	3696	0.4	—	54	3696	0.3	—	0	0.1	<0.01	1	6.0
8	0.5	5487	3.2	—	4	5487	2.2	—	158	5487	2.3	—	158	5487	2.0	—	0	0.1	<0.01	4	2.0
8	1.0	5606	5.4	—	243	5606	1.6	—	225	5606	1.8	—	225	5606	2.1	—	250	0.6	0.11	2	4.0
10	0.5	7483	13	—	4	7483	12	—	136	7483	12	—	137	7483	4.0	—	0	0.2	<0.01	6	1.7
10	1.0	7564	15	—	917	7564	9.2	—	257	7564	9.4	—	257	7564	3.1	—	0	0.6	<0.01	2	5.0
12	0.5	8244	29	—	8	8244	18	—	90	8244	19	—	91	8244	17	—	0	0.2	<0.01	6	2.0
12	1.0	8325	756	—	3004	8325	77	—	1347	8325	75	—	1348	8325	13	—	0	2.1	<0.01	2	6.0
14	0.5	8855	188	—	23	8855	150	—	615	8855	147	—	616	8855	72	—	44	0.2	0.08	6	2.3
14	1.0	8953	352	—	12488	8953	140	—	2076	8953	133	—	2316	8953	172	—	3272	3.6	0.15	2	7.0
16	0.5	10250	337	—	15	10250	167	—	353	10250	162	—	354	10250	74	—	10	0.5	<0.01	9	1.8
16	1.0	10341 [‡]	TL	n/a	36765	10341	TL	0.02	3094	10341	TL	0.02	4533	10341	1800	—	2712	24.1	<0.01	2	8.0
18	0.5	11434 [‡]	TL	n/a	0	11425	TL	0.38	187	11425	TL	0.39	188	11427	TL	0.40	19	0.7	0.12	10	1.8
18	1.0	11560 [‡]	TL	n/a	0	11557	TL	0.05	6378	11559	TL	0.03	4809	11560	TL	0.03	3030	28.4	0.10	3	6.0
20	0.5	12906 [‡]	TL	n/a	0	12905	TL	0.27	114	12905	TL	0.28	115	12908	TL	0.12	41	1.4	0.09	10	2.0
20	1.0	12978 [‡]	TL	n/a	0	12975	TL	0.05	6920	12970	TL	0.09	8692	12978	TL	0.02	6600	17.4	0.12	3	6.7
22	0.5				15125	TL	0.91	62	15125	TL	0.91	63	15151	TL	0.72	23	1.8	<0.01	11	2.0	
22	1.0				15322	TL	0.13	9654	15327	TL	0.10	8698	15340	TL	0.01	7821	98.5	<0.01	3	7.3	
24	0.5				16293	TL	0.76	67	16278	TL	0.83	70	16315	TL	0.64	32	2.2	0.07	12	2.0	
24	1.0				16435	TL	0.06	7793	16437	TL	0.05	5125	16431	TL	0.09	6561	10.3	0.05	5	4.8	
26	0.5				17819	TL	1.1	94	17816	TL	1.1	98	17862	TL	0.87	28	3.0	<0.01	13	2.0	
26	1.0				17996	TL	0.23	6840	18007	TL	0.17	13090	18021	TL	0.09	7327	106.4	0.04	5	5.2	
28	0.5				18735	TL	0.89	76	18735	TL	0.89	77	18776	TL	0.60	30	3.6	<0.01	14	2.0	
28	1.0				18908	TL	0.20	14429	18916	TL	0.16	13588	18930	TL	0.08	4179	65.5	<0.01	6	4.7	
30	0.5				20653	TL	1.1	363	20653	TL	1.1	364	20676	TL	0.94	28	4.5	<0.01	17	1.8	
30	1.0				20732	TL	0.70	18228	20854	TL	0.11	14181	20866	TL	0.05	5393	60.1	<0.01	6	5.0	

Note. **Bold** marks the best entry in each row: the fastest certified time where any configuration certifies, and otherwise the smallest remaining gap, since at a timeout every configuration reports the same elapsed time. Ties are all marked. Obj is each configuration’s own best incumbent; “—” means solved to optimality; “n/a” means no single MIP bound is available; blank cells were not run ($n \geq 22$); [‡] marks incumbents from an interrupted cutting-plane loop, verified SCS-feasible a posteriori but carrying no MIP bound (see text). Heur is the heuristic runtime (s) and HGap its gap (%) to the best known SCS-feasible solution; $|CS|$ and avg. $|c|$ describe the lazy_MIP_ws incumbent. Time limits as in Table 5.2.

5.5.2 Solution strategies for OSCS

Table 5.3 compares the four OSCS configurations under AGG; the e-companion gives DIS.

Separation strategies. All four methods return identical optima wherever they certify, which is the basic correctness check on the separation approaches; at timeout the incumbents differ and each column reports its own. The choice between enumeration and MIP separation matters only when coalitions grow past the threshold. At $\alpha = 0.5$ the formed structures are fragmented, mostly pairs and triples, so $\tau = 9$ absorbs nearly all separation work and lazy and lazy_MIP are indistinguishable. At $\alpha = 1.0$ coalitions are larger and the separation MIP earns its cost: at $n = 30$ it closes the gap to 0.11% against 0.70% for first-violation enumeration. This is the behavior anticipated in Section 5.4.1, where the separation MIP's value was argued to lie in its certificate rather than its cut.

The warm start. The heuristic is the single most effective component; per-instance diagnostics are in Table 5.10 of the e-companion. It is fastest or tied-fastest in 11 of the 15 cells that solve to optimality, and at $n = 16$, $\alpha = 1.0$ it is the only configuration that certifies optimality at all. From $n = 18$ onward nothing certifies, but it holds the smallest gap in 12 of the 14 timeout cells. At $n = 16$, $\alpha = 0.5$ it finishes in 74 seconds against 167, 162 and 337 for the three cold-started configurations. It is cheap where it needs to be, taking at most 4.5 seconds at $\alpha = 0.5$ and up to 106 seconds at $\alpha = 1.0$ where the Core checks run on large pooled coalitions, and its objective is within 0.15% of the best known stable solution. Where it does not pay the cold solve was already short, the only material case being $n = 14$, $\alpha = 1.0$; Table 5.3 has the detail.

The iterative baseline. The zero cut counts for cutplane at $n \in \{18, 20\}$ record that its first re-solve never completed inside the time limit; at $n = 16$, $\alpha = 1.0$ many rounds completed, but the loop never reached a violation-free incumbent. In all five cases the reported incumbent comes from an interrupted loop and carries no MIP bound, which is why those entries are marked $\ddagger$; each was verified SCS-feasible a posteriori. This is the cost of discarding the branch-and-bound tree every round, and it is why Experiment 2 drops cutplane.

Lifted versus basic cuts. Running `lazy_MIP` with lifted against basic cuts on all Experiment 1 instances, the two certify identical optima on the 15 solved cells, with a median runtime ratio of 0.99 and no systematic winner on the five double timeouts (Table 5.6 in the e-companion). This is what Remark 5.9 predicts: neither family moves the root bound, so any advantage is path-dependent. The fallback separates the two regimes structurally, never triggering at $\alpha = 1.0$, where Proposition 5.2 forces $\Delta(s) \geq 0$, and firing on 62.8% of added cuts at $\alpha = 0.5$.

5.5.3 Effect of the coalition policy

Section 5.2.2 argued that the coalition policy, not the CIPG framework, decides whether the partition is a decision worth making. The solutions show this, and they show something further: the policy also decides whether fairness and stability collide, because it controls how large the formed coalitions get.

Structures. Figure 5.2(a) places the computed structures against the two a reader might expect, all singletons and the grand coalition, and neither regime returns either. Under restricted pooling the count runs just above the all-pairs line, so the answers are close to a

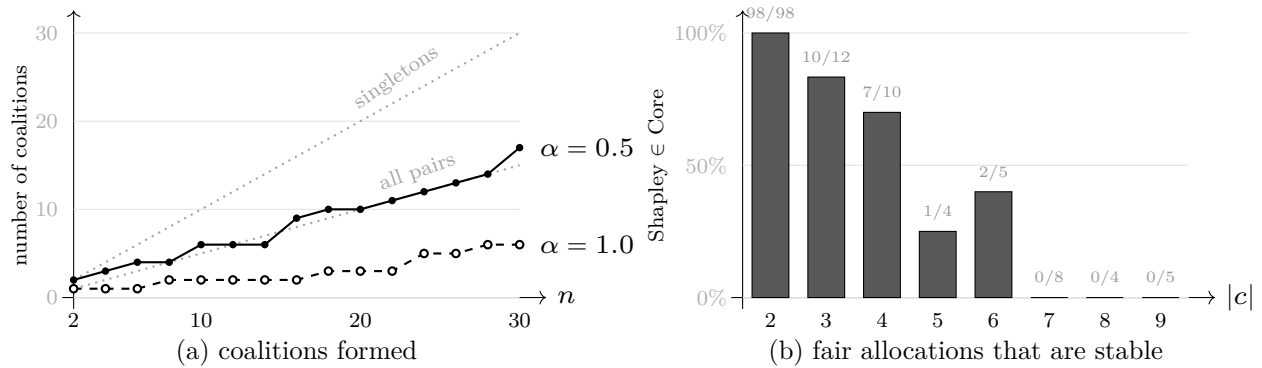


Figure 5.2: The coalition policy sets the size of the formed coalitions, and size decides whether a fair allocation is also stable. (a) Coalitions in the lazy_MIP_ws incumbent (Table 5.3), against the two trivial answers. Restricted pooling gives slightly more coalitions than a perfect pairing, and full pooling stays far above the grand coalition from $n = 8$ on, so neither regime returns a trivial structure. (b) Share of the 146 formed coalitions of each size whose Shapley value lies in the Core, with the count above each bar (Table 5.7).

matching with a few players left alone; what is being decided is which players to pair, not how many coalitions to form. Under full pooling the count sits far from the grand coalition from $n = 8$ onward, and since superadditivity makes the grand coalition the OCS in that regime, every one of those splits is forced by stability alone. Per-instance detail is in Table 5.11.

Stability is free at $\alpha = 0.5$: OCS and OSCS coincide on every instance solved to optimality, that is for $n \leq 16$. Beyond $n = 16$ neither run is certified, so the incumbents are not comparable, and in three cells the OSCS incumbent is the larger. The agreement has a structural reason: a pair is worth forming exactly when $V(\{i, j\}) \geq V(\{i\}) + V(\{j\})$, and in that case a stable two-way split always exists, so every OCS pair is stabilizable. At $\alpha = 1.0$ the splits are certified through $n = 16$, and only once does one cost anything: at $n = 8$ the OSCS gives up three units, $5,609 \rightarrow 5,606$. That near-zero cost is a property of these instances, not of CIPGs. With three players, $V(\{i\}) = 0$, every pair worth 10 and $V(N) = 12$, the pair inequalities force a total payoff of at least $15 > 12$, so the grand coalition's Core is empty and stability costs nearly 17%.

Payoffs. The same regimes decide whether a fair allocation is also stable. Refining every OCS solution as in Section 5.4.4 treats 171 coalitions, 146 of size at least two (Table 5.7 in the e-companion). The nucleolus is Core-feasible in all 146, as the SCS guarantee requires. The Shapley value is not: it lies in the Core in 118 of the 146, and membership falls off with size until it reaches zero (Figure 5.2(b), where each bar carries its own denominator; the dip at $|c| = 5$ rests on four coalitions). The contrast across regimes is therefore sharp, 100 of 102 at $\alpha = 0.5$ against 18 of 44 at $\alpha = 1.0$, and it is the empirical counterpart of Example 5.15: CKGs are superadditive but not convex, and the Shapley value is guaranteed to lie in the Core only for convex games. Every violation is under 0.232% of $V(c)$, so the Shapley value is nearly stable even when it fails, but a manager who wants both fairness and stability under full pooling should take the nucleolus.

5.5.4 Cost of the framework

Against exhaustive enumeration. As an implementation-independent check we ran a baseline that computes all $2^n - 1$ coalition values, finds the OCS by dynamic programming over partitions, and certifies the OCS with Core-nonemptiness LPs. Wherever it completes, its optimal values and partitions agree exactly with Tables 5.2 and 5.3 (Table 5.5 in the e-companion).

Through $n \approx 14$ to 16 the two routes are comparable. The difference is entirely in the scaling: value enumeration alone is $\Theta(2^n)$, and the OCS search stage reaches more than 25 years at $n = 30$. Our pipeline evaluates 766 of the $2^{30} - 1 \approx 1.07 \times 10^9$ coalition values at $n = 30$, $\alpha = 0.5$, and returns certified sub-1% solutions inside the hour. At small n under full pooling the Core checks touch every subset anyway, so the saving is asymptotic.

Brute force is also all or nothing. Interrupted, it yields neither a usable incumbent nor a

bound, whereas every incumbent the lazy framework accepts carries a certified gap, and every incumbent lazy_MIP_ws returned was verified SCS-feasible, so the pipeline degrades gracefully with n .

5.5.5 Implementation recommendations

Six choices arise. Use the *aggregated* model (5.6) for an OSCS and the *disaggregated* model (5.7) for an OCS; without a warm start, disaggregation collapses at scale. Add stability as *lazy constraints* in one branch-and-bound tree, not by re-solving. Separate by *enumeration* up to $\tau = 9$ and by the *separation MIP* (5.13) above it, which alone returns a certificate. Prefer the *lifted cut* (5.11), falling back to (5.8) when $\Delta(s) < 0$: no faster, but dominant and free. *Always warm start* with Algorithm 5.12, which matters most. And allocate with the *nucleolus*.

Limitations. We claim two things only within limits. First, OSCS optimality is certified only to $n = 16$, and at $\alpha = 1.0$ only by the warm-started configuration, right at the limit; the others certify to $n = 14$. Beyond that, quality rests on the reported gaps, below 1% for lazy_MIP_ws under AGG and 1.1% for the other AGG configurations, and closing them at scale calls for decomposition or branch-and-price. Second, we use one instance per (n, α) pair to span a wide range of sizes and both regimes; a randomized study across seeds and $(|R|, q, \rho)$ would strengthen the conclusions.

5.6 Conclusion

Across the study, what makes a cooperative integer programming game hard is the coalition policy laid over it rather than the integrality of the coalition programs. Under full pooling the grand coalition is optimal in advance and the computational problem is certification. Under a restricted bound the partition becomes a decision worth making, and the difficulties follow: superadditivity is lost, stability needs separation, and DIS collapses without a warm start.

Two directions seem most promising. Since payoff-space stability cuts leave the root bound unchanged, tightening it needs valid inequalities in the assignment and item variables, such as cover inequalities for (5.6d). And SCS guards against fission but not fusion; merge-proofness on top of Core stability remains open [143].

Appendix

5.A Notation

5.B Omitted Proofs and Examples

Example 5.15 (Superadditive but not convex). Take $N = \{1, 2, 3\}$ with one item of weight $a = 4$ and profit $p^i = 4$, available to all three players with $u^i = 1$ each, and budgets $b^i = 2$. Under full pooling a coalition c has budget $2|c|$ and bound $y^c \leq |c|$, so

$$V(\{i\}) = 0, \quad V(\{i, j\}) = 4, \quad V(N) = 4,$$

Table 5.4: Notation for the OCS formulation.

<i>Sets and indices</i>	
N	set of players; index $i \in N$
$G := N$	set of potential group indices; at most $ N $ groups form
R	item set; index $j \in R$
$I_j \subseteq N$	players for whom item j is available
<i>Parameters</i>	
p_j^i	profit of item j for player i
a_j	weight of item j , common across players
b^i	budget of player i
u_j^i	upper bound on usage of item j by player i
B^i, d^i	data of the private constraints $B^i x^i \leq d^i$
$\alpha \in (0, 1]$	pooling parameter; the fraction of the full-pooling bound a coalition may use on a common item
<i>Decision variables</i>	
$z_{ig} \in \{0, 1\}$	1 if player i is assigned to group g
$x_j^{i,g} \geq 0$	usage of item j by player i in group g
$y_j^g \in \mathbb{Z}_+$	aggregate usage of item j in group g
$w_j^g \in \{0, 1\}$	1 if item j is common in group g , that is, available to ≥ 2 assigned players
$\pi_g \geq 0$	total profit generated by group g
$v_i \geq 0$	payoff allocated to player i
$\phi_{ig} \geq 0$	linearization auxiliary, $\phi_{ig} = v_i z_{ig}$ (aggregated model)
$v_{ig} \geq 0$	payoff player i receives from group g (disaggregated model)

since two units would cost $8 > 6$. The game is superadditive, as Proposition 5.2 requires. It is not convex: taking $c = \{1, 2\}$ and $d = \{1, 3\}$ gives

$$V(c) + V(d) = 8 > 4 = V(c \cup d) + V(c \cap d).$$

The second member of a coalition is worth 4 and the third is worth nothing.

Example 5.16 (A budget-only player must be paid beyond its own profit potential). Consider three players with budgets $b^1 = b^2 = 1$ and $b^3 = 10$, an item of profit 8 and weight 10 available only to player 1, an identical item available only to player 2, and a common item of profit 2 and weight 2 available to players 1 and 2. Under full pooling, $V(\{1\}) = V(\{2\}) = V(\{3\}) = 0$, $V(\{1, 2\}) = 2$, $V(\{1, 3\}) = V(\{2, 3\}) = 8$, and $V(N) = 10$. Summing the Core inequalities for the two mixed pairs gives $v_1 + v_2 + 2v_3 \geq 16$, and with $v_1 + v_2 + v_3 = 10$

this forces $v_3 \geq 6$; e.g., $(2, 2, 6)$ is a Core allocation. Since player 3 has no items, $U_3 = 0$, so the cap $v_3 \leq U_3$ would cut off *every* Core allocation of the grand coalition.

Proof. Proof of Proposition 5.3. (*Same mixed-integer feasible points.*) Fix an integer assignment z and let $g^*(i)$ be the unique group with $z_{i,g^*(i)} = 1$. In the aggregated model, McCormick at integer z gives $\phi_{i,g^*(i)} = v_i$ and $\phi_{ig} = 0$ for $g \neq g^*(i)$, so efficiency becomes $\pi_g = \sum_{i: g^*(i)=g} v_i$. In the disaggregated model, the upper bound in (5.7) forces $v_{ig} = 0$ for $g \neq g^*(i)$, and linking gives $v_{i,g^*(i)} = v_i$, so efficiency yields the same expression. Hence the two formulations have the same mixed-integer feasible points.

($F_{\text{disagg}} \subseteq F_{\text{agg}}$.) We show every aggregated constraint is satisfied by points of F_{disagg} . (i) The upper bound $v_{ig} \leq Mz_{ig}$ is exactly the McCormick constraint $\phi_{ig} \leq Mz_{ig}$. (ii) The efficiency $\pi_g = \sum_i v_{ig}$ is $\pi_g = \sum_i \phi_{ig}$, which is the aggregated efficiency constraint. (iii) The remaining two McCormick constraints are implied by the linking constraint $v_i = \sum_g v_{ig}$: first, $\phi_{ig} = v_{ig} \leq \sum_{g'} v_{ig'} = v_i$ since $v_{ig'} \geq 0$; second, $v_{ig} = v_i - \sum_{g' \neq g} v_{ig'} \geq v_i - \sum_{g' \neq g} Mz_{ig'} = v_i - M(1 - z_{ig})$, using $v_{ig'} \leq Mz_{ig'}$ and the partition constraint $\sum_{g'} z_{ig'} = 1$. Since the linking constraint is *not* present in F_{agg} , every point of F_{disagg} satisfies all constraints of F_{agg} , so $F_{\text{disagg}} \subseteq F_{\text{agg}}$.

(Strict inclusion): For suitable instance data, consider $|N| = 2$ with $M = 100$, and take $z_{11} = 1$, $z_{21} = z_{22} = 0.5$, $v_2 = 50$; this satisfies the partition and symmetry-breaking constraints, and it suffices that the instance admits fractional item usage generating group profits of 50 in each group (two items with sufficiently large profit and capacity suffice, since the x -variables are continuous in the LP relaxation). In F_{agg} : $\phi_{21} \leq \min(v_2, Mz_{21}) = 50$, $\phi_{21} \geq v_2 - M(1 - z_{21}) = 0$, and similarly for ϕ_{22} . Setting $\phi_{21} = \phi_{22} = 50$ contributes $100 > 50 = v_2$ to the group values: the LP relaxation double-counts player 2's payoff across groups. In F_{disagg} : $v_{21} + v_{22} = v_2 = 50$, so player 2's payoff is counted only once.

Finally, adding the linking constraint $v_i = \sum_g \phi_{ig}$ to F_{agg} yields exactly the constraint set of F_{disagg} , which proves the last assertion. $\square$

Proof. Proof of Proposition 5.4. (i) Let $k := |s| - \sum_{i \in s} z_{ig} \in \mathbb{Z}_{\geq 0}$. If $k = 0$, then $s \subseteq c(g)$ and (5.8) is the Core inequality $\sum_{i \in s} v_i \geq V(s)$ for s within $c(g)$, which holds by SCS-feasibility. If $k \geq 1$, then the right-hand side of (5.8) equals $V(s) - M_s k \leq V(s) - M_s \leq 0$, while every SCS-encoding point satisfies $v_i \geq V(\{i\}) \geq 0$ for each player through singleton stability within the player's own coalition, so $\sum_{i \in s} v_i \geq 0$. (ii) With $s \subseteq c(g)$ we have $k = 0$, so (5.8) reads $\sum_{i \in s} v_i \geq V(s)$, violated by assumption. $\square$

Proof. Proof of Proposition 5.7. Fix a mixed-integer point of $\mathcal{F}^{\text{stab}}$ and a group g , and let $k := \sum_{i \in s} z_{ig} \in \{0, \dots, |s|\}$. If $k = |s|$, then $s \subseteq c(g)$ and the right-hand side of (5.11) equals $\Delta(s) + \sum_{i \in s} V(\{i\}) = V(s)$, so (5.11) is the Core inequality for s within $c(g)$, which holds by SCS-feasibility. If $k \leq |s| - 1$, then $k - |s| + 1 \leq 0$ and $\Delta(s) \geq 0$ imply

$$\Delta(s)(k - |s| + 1) + \sum_{i \in s} V(\{i\}) z_{ig} \leq \sum_{i \in s} V(\{i\}) z_{ig} \leq \sum_{i \in s} V(\{i\}),$$

using $z_{ig} \leq 1$ and $V(\{i\}) \geq 0$. Every player i belongs to some group $g'(i)$, and the singleton member of the stability family (5.8) for that group yields $v_i \geq V(\{i\})$. Summing over $i \in s$ gives $\sum_{i \in s} v_i \geq \sum_{i \in s} V(\{i\})$, which dominates the right-hand side above. $\square$

Proof. Proof of Proposition 5.8. With $M_s = V(s)$ the right-hand side of (5.8) equals $V(s)(\sum_{i \in s} z_{ig} - |s| + 1)$. Subtracting it from the right-hand side of (5.11) and using $V(s) = \Delta(s) + \sum_{i \in s} V(\{i\})$ gives

$$\sum_{i \in s} V(\{i\}) \left[z_{ig} - \left(\sum_{\ell \in s} z_{\ell g} - |s| + 1 \right) \right] = \sum_{i \in s} V(\{i\}) \left[(|s| - 1) - \sum_{\ell \in s \setminus \{i\}} z_{\ell g} \right] \geq 0,$$

since each bracket is a sum of $|s| - 1$ terms of the form $1 - z_{\ell g} \geq 0$ and $V(\{i\}) \geq 0$. $\square$

Proof. Proof of Proposition 5.13. *Termination.* Each executed merge strictly decreases $|CS|$, so the loop of Algorithm 5.12 runs at most $|N| - 1$ times. Each Core check terminates: for $|c| \leq \tau$ it enumerates finitely many subcoalitions, and for $|c| > \tau$ each round of row generation adds a subcoalition of c not generated before, of which there are finitely many. Both loops are additionally capped by T_1 , and Phase 2 by T_2 .

The incumbent is an SCS at the end of initialization. The singleton partition with $\tilde{v}_i = V(\{i\})$ is an SCS by Proposition 5.6.

The invariant is maintained. The incumbent is modified only in the last step before the return, and only when a Core allocation has been obtained for every coalition of CS' . Every $c \in \mathcal{P}$ other than a singleton entered $\mathcal{P}$ only after **CORENONEMPTY**(c) returned true, which certifies that (c, V_c) has a nonempty Core, and $CS' \subseteq \mathcal{P}$ is a partition of N because $\mathcal{P}$ contains the singletons, so (5.14) is feasible and its seeded solution is available even if T_2 expires. The replacement therefore satisfies $\sum_{i \in c} \tilde{v}_i = V(c)$ and $\sum_{i \in s} \tilde{v}_i \geq V(s)$ for every $s \subseteq c$ and every $c \in CS'$, which is the definition of an SCS. If the allocations are not all obtained, the previous incumbent is returned unchanged.

Initialization is necessary. An SCS requires $\sum_{i \in s} \tilde{v}_i \geq V(s)$ for every subcoalition s of every formed coalition, and $s = \{i\}$ is such a subcoalition, so $V(\{i\})$ is needed for every i regardless of the partition returned.

No argument after initialization uses the size of T_1 or T_2 : a smaller budget yields a smaller pool, never an uncertified one, because an inconclusive check is treated as a failure.

Cost. Each executed merge reduces $|CS|$ by one, so the loop passes through partitions of size $|N|, |N| - 1, \dots, 2$ and considers at most $\binom{k}{2}$ candidate pairs when $|CS| = k$. Summing, $\sum_{k=2}^{|N|} \binom{k}{2} = \binom{|N|+1}{3}$, and each candidate receives at most one Core check. A check on $|c| \leq \tau$ enumerates the $2^{|c|} - 2 \leq 2^\tau - 2$ proper nonempty subcoalitions of c ; above the threshold

the row-generation loop adds a distinct subcoalition each round and is bounded by T_1 . $\square$

5.C Extension to General MILP-Based CIPGs

Sections 5.2 and 5.3 were written for the resource allocation game of Section 5.2.1. With the model and its cut families in place we can say exactly how much of that structure they use.

Two components use none of it. The stability inequalities (5.8) and (5.11) reference the instance only through the values $V(s)$ and $V(\{i\})$, so Propositions 5.4, 5.7 and 5.8 and Theorem 5.5 hold verbatim for any CIPG in the sense of Definition 5.1. The formulation comparison of Section 5.3.1.1 involves only the assignment and payoff variables, never the underlying player variables, so Proposition 5.3 is unchanged.

Two components are written in the rows of that model and are restated rather than inherited; We give the two restatements component by component. Throughout, consider CIPGs in which each player solves a bounded MILP in the sense of Definition 5.1, with designated poolable (budget-type) rows and private structural rows.

- *OCS formulations.* Introduce group-indexed copies of each player's variables and aggregate the right-hand sides of the poolable rows across the members assigned to a group, exactly as in (5.6); private rows are gated by the assignment variables as in (5.6). The McCormick and disaggregated linearizations, and hence Proposition 5.3, carry over unchanged, because they involve only the assignment and payoff variables—never the underlying player variables.
- *Stability inequalities.* The basic cut (5.8) and the lifted cut (5.11) reference the instance only through the values $V(s)$ and $V(\{i\})$, so their validity (Propositions 5.4 and 5.7) is independent of how those values are generated.

- *Separation.* The single-level reformulation of the separation MIP (5.13) requires only that the subset-selection variables enter the inner coalition problem through right-hand sides and variable bounds, and that player variables are bounded—both hold for the general model.
- *Heuristic and refinement.* The SCS-feasible heuristic and the Shapley/nucleolus refinement access the instance solely through a $V(c)$ oracle and small LPs.

Consequently, instantiating the framework for another MILP class (e.g., facility-location or assignment-type games) changes only the inner coalition-level MILPs, not the formulation, the cuts, or the algorithms; the CKG of Section 5.5 is one such instantiation.

Remark 5.17 (Relation to integer minimization games). Under full pooling the resource allocation game of Section 5.2.1 fits the maximization mirror of the (mixed-integer) *integer minimization games* of Caprara and Letchford [32]: the rows are fixed, and the subset enters only through right-hand sides and variable bounds that are linear in its incidence vector, which is exactly the condition under which the separation MIP (5.13) exists. The proof of Proposition 5.2 mirrors that of their Observation 1, the automatic subadditivity of the class. Restricted pooling leaves the class: the common-or-individual designation makes the presence of the α -row depend on the subset itself, which is what the indicators w_j^g and κ_j encode, and it is also what makes the coalition structure a nontrivial decision. The two frameworks also answer an empty Core differently: γ -budget-balanced cost shares relax efficiency for a fixed grand coalition, while an SCS restores exact efficiency by refining the partition; the two relaxations are orthogonal and could be combined.

One further transfer is useful in computation. The mirror of their Observation 7 states that, within a formed coalition, the Core inequality for a subset s is redundant whenever s splits into s_1, s_2 with $V(s_1) + V(s_2) \geq V(s)$, since summing the inequalities for s_1 and s_2 dominates

it. The subset enumerations of Section 5.4.1 and the Core checks of Algorithm 5.12 therefore need only the *indecomposable* subsets, and in the gated family (5.8) the same reduction is valid at integer incumbents in the presence of the singleton cuts: with the gate closed the summed cuts for s_1 and s_2 dominate, and with it open the right-hand side is nonpositive.

5.D Complexity of the Stability Problems

A CIPG instance is the compact data of Section 5.2.1; the $2^{|N|} - 1$ coalition values are not part of the input but are generated from it, and that encoding is the source of the difficulty. When the values are supplied explicitly, an optimal coalition structure follows by dynamic programming over subsets in $O(3^{|N|})$ time [123, 132], polynomial in the size of that exponential input; this is the baseline of Section 5.5.4. We assume the data are rational and every coalition program feasible and bounded.

Proposition 5.18 (Evaluation and partition selection). *Deciding $V(c) \geq K$ is NP-complete, and so is deciding whether some coalition structure satisfies $\sum_{c \in CS(N)} V(c) \geq K$.*

Proof. Proof. For $V(c) \geq K$, guess a plan for c and verify its value; hardness is **KNAPSACK** with $|c| = 1$. For the partition problem, guess a coalition structure and one plan per block: a structure has at most $|N|$ blocks, and an understated coalition value can only understate the objective, so no guessed plan need be optimal. Hardness again follows with $|N| = 1$. $\square$

Membership is the informative half: one *feasible* plan per block certifies the bound, and no guess need be optimal. That property fails once stability is imposed, and it is why Section 5.3.1 can be a compact mixed-integer program while Section 5.3.2 cannot.

Proposition 5.19 (Separation). *Deciding whether some $s \subseteq c$ satisfies $\sum_{i \in s} \bar{v}_i < V(s)$ is NP-complete.*

Proof. Proof. Guess s together with a plan for s whose value exceeds $\sum_{i \in s} \bar{v}_i$; a feasible plan suffices, since exhibiting one proves $V(s) > \sum_{i \in s} \bar{v}_i$. Hardness is **KNAPSACK** with $|c| = 1$. $\square$

Separation of the stability family is therefore not a subroutine a better implementation would render polynomial, and its natural certificate—a subset with a plan that beats the incumbent’s payoff—is exactly the pair returned by (5.13).

Proposition 5.20 (Verification and membership). *Deciding whether a given pair $(CS(N), v)$ is an SCS lies in DP and is hard for both NP and coNP; it therefore lies in neither class unless $\text{NP} = \text{coNP}$. Deciding whether a given v admits some SCS is likewise coNP-hard.*

Proof. Proof. *Membership.* The pair is an SCS iff for every $c \in CS(N)$: (a) $\sum_{i \in c} v_i = V(c)$, and (b) $\sum_{i \in s} v_i \geq V(s)$ for all $s \subseteq c$. Condition (b), over at most $|N|$ blocks, is the complement of the separation problem and lies in coNP by Proposition 5.19. Given (b), the case $s = c$ supplies $\sum_{i \in c} v_i \geq V(c)$, so what remains of (a) is the residual assertion $V(c) \geq \sum_{i \in c} v_i$, a polynomial conjunction of value queries and hence in NP. The problem is the intersection of a coNP and an NP language, which is the class DP.

Hardness for NP. Take $|N| = 1$ with no private constraints and every item available, so (5.5) is a 0–1 knapsack, and set $v_1 = K$. Then $(\{\{1\}\}, v)$ is an SCS iff $V(\{1\}) = K$; reduce from **SUBSETSUM**, taking profits equal to weights and capacity equal to the target.

Hardness for coNP. An instance must supply an efficient v , which presupposes $V(c)$, so we build a gadget in which $V(c)$ is computable by inspection while a proper subcoalition’s value is not. Given a knapsack instance with capacity B , integer profits and target K , take $N = \{1, 2\}$ and $c = \{1, 2\}$ with every item available to player 1 only, so $I_j = \{1\}$

and $R_{\text{com}}^c = \emptyset$, and player 2 holding no items. Set $b^1 = B$ and b^2 large enough that $b^1 + b^2 \geq \sum_{j \in R} a_j u_j^1$. Then $V(c) = P := \sum_{j \in R} p_j^1 u_j^1$, $V(\{2\}) = 0$, and $V(\{1\})$ is the knapsack optimum at capacity B . Assume $K \leq P$ and put $v = (K - \frac{1}{2}, P - K + \frac{1}{2})$, so $v_1 + v_2 = P = V(c)$ and efficiency holds by construction. Of the subsets of c : $s = \{2\}$ requires $v_2 \geq 0$, which holds; $s = c$ holds with equality; and $s = \{1\}$ requires $K - \frac{1}{2} \geq V(\{1\})$, that is $V(\{1\}) < K$, the profits being integers. Hence $(\{c\}, v)$ is an SCS iff $V(\{1\}) < K$.

Hardness of membership. In the same gadget the only coalition structures are $\{\{1, 2\}\}$ and $\{\{1\}, \{2\}\}$; the second requires $v_2 = V(\{2\}) = 0$ while $v_2 \geq \frac{1}{2}$, so it is never an SCS. Therefore v admits an SCS iff $V(\{1\}) < K$. $\square$

Stability alone—condition (b)—is coNP-complete by Proposition 5.19. Efficiency alone is *not* in NP: $\sum_{i \in c} v_i = V(c)$ is an exact-optimum assertion. Only the residual assertion, once (b) has supplied one inequality, is an NP question, and it is the conjunction that leaves both classes. This is why Section 5.4.1 verifies time-limit incumbents by re-solving $V(c)$ for every formed coalition rather than reading efficiency off the model: (5.6c) gives only $\sum_{i \in c(g)} v_i = \pi_g \leq V(c(g))$, and closing that to equality is the NP half.

Corollary 5.21 (No compact description of $\mathcal{S}$). *Suppose that for every instance there is a system in the payoff variables v and auxiliary variables w , of size polynomial in the input, with polynomial-time checkable constraints, and with every $v \in \mathcal{S}$ the projection of some w of polynomial encoding length, whose projection onto v is exactly $\mathcal{S}$. Then $\text{NP} = \text{coNP}$.*

Proof. Proof. Under the hypothesis, deciding $v \in \mathcal{S}$ lies in NP: guess w and check the constraints. By the last part of Proposition 5.20 that decision problem is coNP-hard, so $\text{coNP} \subseteq \text{NP}$. $\square$

Theorem 5.5 describes $\mathcal{S}$ exactly, but only through a family of inequalities that cannot be

written down in advance. Corollary 5.21 says this is not an artifact of our encoding. Because it permits auxiliary variables and bounds the total size, it forbids every polynomial-size exact formulation, not only those in the payoff variables: exactness cannot be bought by trading inequalities for variables. What it does not exclude is $\mathcal{F}^{\text{stab}}$ itself, which uses polynomially many variables and exponentially many inequalities. What creates the gap is stability rather than integrality; the same obstruction appears for worth functions computable in polynomial time, where core membership over coalition structures is already coNP-complete.

Proposition 5.22 (Optimization over stable structures). *Deciding whether there is an SCS $(CS(N), v)$ with $\sum_{i \in N} v_i \geq K$ lies in Σ_2^P and is NP-hard.*

Proof. *Proof. Membership.* For each $c \in CS(N)$ the restriction of v to c lies in the Core of (c, V_c) , the polyhedron $\{w : \sum_{i \in c} w_i = V(c), \sum_{i \in s} w_i \geq V(s) \forall \emptyset \neq s \subsetneq c\}$. Singleton stability gives $w_i \geq V(\{i\}) \geq 0$ and efficiency gives $w_i \leq V(c)$, so the polytope is bounded and nonempty and has a vertex, which solves a nonsingular $|c| \times |c|$ subsystem with 0–1 coefficients and coalition values as right-hand sides. By Cramer’s rule and the Hadamard bound the vertex is polynomially encodable. Replacing v on each block by such a vertex preserves the structure and the total payoff, so the witness is polynomially balanced, and verifying it means one instance of Proposition 5.20, which lies in $\text{DP} \subseteq \text{P}^{\text{NP}}$. Hence the problem lies in $\exists \cdot \text{P}^{\text{NP}} = \Sigma_2^P$.

Hardness. Take $|N| = 1$; the only SCS payoff is $v_1 = V(\{1\})$. □

We do not settle the exact complexity of the OSCS problem and regard Σ_2^P -completeness as the natural conjecture; Corollary 5.21 does not depend on it. The obstruction is that the Δ_2^P -hardness constructions for core problems over coalition structures assume a polynomial-time worth function, and in that regime the second level is not reached.

Remark 5.23 (Where the difficulty lies on our instances). Proposition 5.18 is not a statement about the instances of Section 5.5. With $a_j \sim U\{5, 10\}$ the coalition programs are easy: Table 5.5 reports 262,143 values in 2,126.9 seconds at $n = 18$, $\alpha = 1.0$, about 8 milliseconds each, and 3,628.4 seconds at $\alpha = 0.5$, about 14 milliseconds each. The gain in Section 5.5.4 comes from evaluating 766 values rather than 10^9 , not from evaluating any one faster.

Remark 5.24 (Why fission stability, and not the coalition structure core). Our notion allows only subsets of a *formed* coalition to block. The coalition structure core [95, Def. 3.31] instead requires $\sum_{i \in s} v_i \geq V(s)$ for every $s \subseteq N$. Under that definition the question this chapter asks does not arise. Let $(CS(N), v)$ satisfy it. Efficiency on each block gives $\sum_{i \in N} v_i = \sum_{c \in CS(N)} V(c)$, while applying the requirement blockwise to an arbitrary partition $\mathcal{Q}$ gives $\sum_{i \in N} v_i = \sum_{T \in \mathcal{Q}} \sum_{i \in T} v_i \geq \sum_{T \in \mathcal{Q}} V(T)$. Taking the maximum over $\mathcal{Q}$,

$$\sum_{c \in CS(N)} V(c) = \sum_{i \in N} v_i \geq \max_{\mathcal{Q}} \sum_{T \in \mathcal{Q}} V(T), \quad (5.15)$$

so $CS(N)$ is already an optimal coalition structure. Under the coalition structure core every stable structure is welfare-maximal by definition, and the gap between an OCS and an OSCS—the quantity Section 5.5.3 measures—is identically zero.

The stronger notion is also frequently empty here. Call V *cohesive* if $\sum_{c \in CS(N)} V(c) \leq V(N)$ for every partition; superadditivity implies cohesion. Under cohesion (5.15) and the requirement at $s = N$ give $\sum_{i \in N} v_i = V(N)$, so v lies in the Core of the grand coalition. Under full pooling Proposition 5.2 makes V superadditive, and at $n = 8$, $\alpha = 1.0$ our certified runs report an optimal coalition structure of value 5,609 against a best stable structure of 5,606: no stable structure attains $V(N)$, so the grand coalition's Core is empty and with it the coalition structure core.

Fission stability is what makes an SCS always exist (Proposition 5.6) and the price of stability a computable quantity rather than a definitional zero. The cost is that our structures are not protected against a coalition spanning two blocks, which is the fusion direction we leave open.

5.E A Global Bound on the Objective

The root-bound margin of Section 5.5.1 compares AGG_α and DIS_α as stated. A single inequality narrows it. Since $\sum_{i \in N} v_i = \sum_{g \in G} \pi_g$ by (5.6b)–(5.6c), since $\pi_g \leq V_\alpha(c(g)) \leq V_{\alpha=1}(c(g))$, and since $V_{\alpha=1}$ is superadditive by Proposition 5.2,

$$\sum_{i \in N} v_i \leq V_{\alpha=1}(N) \tag{5.16}$$

is valid for both formulations at every α , at the cost of one coalition-value solve. At $n = 4$ it takes the aggregated root bound from 4.5 times the integer optimum to 1.02 at $\alpha = 0.5$ and to the optimum itself at $\alpha = 1.0$, which is tighter than the disaggregated root bound.

We do not use it, for two reasons. At $\alpha = 1.0$ the bound is not informative: Proposition 5.2 already identifies the grand coalition as an OCS, so computing $V_{\alpha=1}(N)$ solves the problem that (5.16) is meant to help solve. And bounding the objective directly leaves the relaxation tight at every node of the tree, which removes the differences in bound that branching uses to choose variables; node counts rose whenever we added it. The informative case is $\alpha < 1$ and the OCS problem, where the optimum lies strictly below $V_{\alpha=1}(N)$, and there (5.16) deserves a proper computational study.

5.F Additional Computational Results

Table 5.6 reports the lifted-versus-basic cut ablation and Table 5.8 the AGG-versus-DIS formulation effect for the OSCS methods; both are discussed in Section 5.5. Table 5.9 completes the ten configurations of Experiment 1 by reporting the cutplane and lazy runs under the disaggregated formulation. Table 5.10 reports per-instance diagnostics of the SCS-feasible heuristic (phase times, Core checks, pool sizes, and the seeded partition), complementing the Heur and HGap columns of Table 5.3. Table 5.7 summarizes the payoff refinement of Section 5.4.4 by coalition size. Table 5.5 reports the comparison with the exhaustive baseline of Section 5.5.

Exhaustive baseline: detailed comparison. In Table 5.5, “Evaluations (ours)” counts the distinct coalition values computed by heuristic plus solver (one shared cache), “Time, ours” is the wall-clock time of lazy_MIP_ws (AGG) for the entire OSCS computation, and the BF columns report the baseline’s two stages (V -enumeration measured to $n = 18$, OCS to $n = 16$, OSCS search to $n = 14$). At small n under full pooling the heuristic’s Core checks alone touch every subset, so the evaluation economy of Table 5.5 is an asymptotic phenomenon, emerging from $n \approx 10$ onward; brute force scales as $\Theta(2^n)$ in value evaluations alone, already requiring over 2,000 seconds at $n = 18$ and an extrapolated 10^7 seconds at $n = 30$, whereas the lazy framework queries V only where stability of a candidate partition is actually at stake. For exact certification, the two routes are in fact comparable through $n \approx 14$ – 16 —brute force is even competitive at $n = 16$, $\alpha = 1.0$, where its extrapolated total of roughly 18 minutes undercuts our time limit. The separation is in scaling: the OSCS-search stage passes from hours at $n = 20$ to days at $n = 22$ and to more than 25 years at $n = 30$. The deeper difference is that brute force is all-or-nothing: interrupted before completion, it yields neither a usable incumbent nor a bound, whereas every incumbent

accepted by the lazy framework carries a certified optimality gap, and every incumbent lazy_MIP_ws returned was verified SCS-feasible, so the pipeline degrades gracefully as n grows.

Lifted versus basic cuts: details. On the five double-timeout cells of the ablation, neither variant achieves systematically smaller final gaps (each is smaller on some cells; all gaps remain below 0.5%): since neither family tightens the root relaxation, any advantage must arise from path-dependent branch-and-bound effects, and none is systematic on these instances. The fallback counts are 0 of 342,585 added cuts at $\alpha = 1.0$ —structural, since the full-pooling bound makes V superadditive and the synergy condition $\Delta(s) \geq 0$ of Proposition 5.7 holds for every subset—versus 25,954 of 41,346 (62.8%) at $\alpha = 0.5$, both aggregated over all lazy, lazy_MIP, and lazy_MIP_ws runs of the two experiments, where restricted bounds create subadditive subsets.

Formulation effect: details. In the † cells of Table 5.8, the solver returns singleton partitions with near-zero objectives (303–375 versus the expected 12,900–20,900); the first such collapse already appears at $n = 20$, $\alpha = 0.5$, and the one remaining configuration with $n \geq 24$ times out normally with a sub-1% gap. The mechanism is model size: the disaggregated formulation introduces $|N| \cdot |G|$ additional payoff variables v_{ig} and linking constraints, which—combined with the exponentially many lazy stability cuts—overwhelm the solver’s ability to find feasible solutions at scale, and lazy_MIP_ws with DIS achieves gaps comparable to AGG (both under 1%) precisely because the warm start removes the feasibility burden.

Coalition structure patterns: details. The near-zero cost of stability has a structural explanation. At $\alpha = 0.5$, the formed coalitions are predominantly pairs, and a pair is worth

forming precisely when $V(\{i, j\}) \geq V(\{i\}) + V(\{j\})$ —in which case a stable two-way split always exists, so every OCS pair is automatically stabilizable. At $\alpha = 1.0$, the pooled coalitions in our instances are large enough that splitting them costs little or nothing (at $n = 10$, both the grand-coalition OCS and the two-coalition OSCS attain 7,564). This reflects our test instances rather than CIPGs in general: with $V(\{i\}) = 0$, each pair worth 10, and $V(N) = 12$, the three pair inequalities force a total payoff of at least $15 > 12$, so the grand coalition’s Core is empty and the cost of stability is nearly 17%.

Payoff refinement: details. The refinement enumerates all subset values $V(s)$ of each formed coalition with the coalition-level IP (under the same α -restricted policy) and solves the nucleolus by the sequential-LP scheme; formed coalitions are small, so at most $2^{|c|} - 1$ small IPs are required per coalition. Since large coalitions arise mainly under $\alpha = 1.0$, the Shapley Core-membership contrast across pooling regimes is sharp (100/102 at $\alpha = 0.5$ versus 18/44 at $\alpha = 1.0$)—an empirical reflection of the non-convexity discussed in Section [5.2.2](#).

Table 5.5: Coalition-value evaluations: proposed pipeline (lazy_MIP_ws, AGG) versus the exhaustive baseline. Measured V -enumeration times differ by α —at $n = 18$, 3,628 seconds at $\alpha = 0.5$ against 2,127 at $\alpha = 1.0$ —and the brute-force columns report the $\alpha = 1.0$ series, which understates the baseline’s cost at $\alpha = 0.5$.

n	$2^n - 1$	Evaluations (ours)		Time, ours (s)		BF time (s)	
		$\alpha = 0.5$	$\alpha = 1.0$	$\alpha = 0.5$	$\alpha = 1.0$	V -enum.	OSCS search
2	3	3	3	0.09	0.01	0.0	0.0
4	15	12	15	0.2	0.1	0.1	0.0
6	63	28	63	0.6	0.3	0.3	0.1
8	255	51	255	2.0	2.1	1.3	0.2
10	1,023	81	321	4.0	3.1	6.4	1.3
12	4,095	123	1,009	17	13	26.5	10.1
14	16,383	184	6,151	72	172	112.8	76.5
16	65,535	213	6,076	74	1800	509.5	≈ 580
18	262,143	271	6,025	TL (0.40%)	TL (0.03%)	2,126.9	$\approx 4,392$
20	1,048,575	357	14,732	TL (0.12%)	TL (0.02%)	$\approx 8,878$	$\approx 33,224$
22	4,194,303	418	15,840	TL (0.72%)	TL (0.01%)	$\approx 37,061$	$\approx 251,597$
24	16,777,215	478	13,567	TL (0.64%)	TL (0.09%)	$\approx 154,706$	$\approx 1,905,298$
26	67,108,863	573	17,448	TL (0.87%)	TL (0.09%)	$\approx 645,792$	$\approx 14,428,445$
28	268,435,455	666	11,727	TL (0.60%)	TL (0.08%)	$\approx 2,695,743$	$\approx 109,263,786$
30	1,073,741,823	766	11,840	TL (0.94%)	TL (0.05%)	$\approx 11,252,893$	$\approx 827,433,238$

Note. Brute-force (BF) times are measured up to the stage caps (V -enumeration to $n = 18$, OSCS search to $n = 14$) and geometrically extrapolated beyond (marked $\approx$); TL entries in “Time, ours” report the remaining MIP gap. Column definitions are given in the text.

Table 5.6: Ablation: lifted stability cuts (5.11) versus basic cuts (5.8) within lazy_MIP (AGG, no warm start).

n	α	lifted				basic				Obj
		Time	Gap	Cuts	CB	Time	Gap	Cuts	CB	
2	0.5	0.01	—	1	8	0.02	—	1	8	1340
2	1.0	0.01	—	2	13	0.02	—	2	13	1365
4	0.5	0.2	—	7	17	0.2	—	7	17	3037
4	1.0	0.1	—	14	15	0.1	—	14	14	3105
6	0.5	0.5	—	35	18	0.5	—	35	18	3622
6	1.0	0.4	—	54	18	0.3	—	56	17	3696
8	0.5	2.3	—	158	20	7.7	—	151	18	5487
8	1.0	1.8	—	225	23	10	—	296	49	5606
10	0.5	12	—	137	26	10	—	156	30	7483
10	1.0	9.4	—	257	45	10	—	300	80	7564
12	0.5	19	—	91	33	19	—	118	25	8244
12	1.0	75	—	1348	105	82	—	1003	48	8325
14	0.5	147	—	616	46	109	—	326	39	8855
14	1.0	133	—	2316	133	61	—	1611	64	8953
16	0.5	162	—	354	40	118	—	223	35	10250
16	1.0	TL	0.02	4533	228	TL	0.04	2488	136	10341
18	0.5	TL	0.39	188	62	TL	0.47	73	59	11425
18	1.0	TL	0.03	4809	181	TL	0.03	2630	81	11559
20	0.5	TL	0.28	115	95	TL	0.27	107	83	12905
20	1.0	TL	0.09	8692	234	TL	0.03	6131	181	12970

Note. All Experiment 1 instances. Time (s), MIP gap (%) at timeout (“—” means solved to optimality), number of added cuts, and callback invocations (CB). Obj is the common incumbent (identical optima wherever both arms certify optimality).

Table 5.7: Payoff refinement summary over the 146 formed OSCS coalitions of size ≥ 2 .

$ c $	#Coalitions	Shapley $\in$ Core	Max. viol. (% of $V(c)$)	Thin Core
2	98	98	—	0
3	12	10	0.145	0
4	10	7	0.057	0
5	4	1	0.130	0
6	5	2	0.049	0
7	8	0	0.232	2
8	4	0	0.201	2
9	5	0	0.078	3
Total	146	118	0.232	7

Note. All instances ($n = 2, \dots, 30$, both α values). Shapley value and nucleolus of each restricted game are computed by enumerating all subset values $V(s)$ with the coalition-level IP. “Thin Core” counts coalitions with least-core $\varepsilon > -0.01$. The nucleolus is Core-feasible in all 146 cases.

Table 5.8: OSCS formulation effect: AGG vs. DIS within lazy_MIP and lazy_MIP_ws.

n	α	lazy_MIP					lazy_MIP_ws					Obj
		AGG		DIS		Better	AGG		DIS		Better	
		Time	Gap	Time	Gap		Time	Gap	Time	Gap		
2	0.5	0.01	—	0.01	—	DIS	0.09	—	0.01	—	DIS	1340
2	1.0	0.01	—	0.01	—	DIS	0.01	—	0.01	—	DIS	1365
4	0.5	0.2	—	0.3	—	AGG	0.2	—	0.2	—	AGG	3037
4	1.0	0.1	—	0.2	—	AGG	0.1	—	0.1	—	DIS	3105
6	0.5	0.5	—	0.6	—	AGG	0.6	—	0.5	—	DIS	3622
6	1.0	0.4	—	0.3	—	DIS	0.3	—	0.2	—	DIS	3696
8	0.5	2.3	—	6.5	—	AGG	2.0	—	2.9	—	AGG	5487
8	1.0	1.8	—	3.1	—	AGG	2.1	—	2.2	—	AGG	5606
10	0.5	12	—	3.0	—	DIS	4.0	—	2.9	—	DIS	7483
10	1.0	9.4	—	15	—	AGG	3.1	—	1.8	—	DIS	7564
12	0.5	19	—	74	—	AGG	17	—	5.6	—	DIS	8244
12	1.0	75	—	217	—	AGG	13	—	169	—	AGG	8325
14	0.5	147	—	99	—	DIS	72	—	66	—	DIS	8855
14	1.0	133	—	42	—	DIS	172	—	182	—	AGG	8953
16	0.5	162	—	259	—	AGG	74	—	TL	0.33	AGG	10250
16	1.0	TL	0.02	TL	0.06	AGG	1800	—	TL	0.04	AGG	10341
18	0.5	TL	0.39	TL	0.66	AGG	TL	0.40	TL	0.43	AGG	11425
18	1.0	TL	0.03	TL	0.05	AGG	TL	0.03	TL	0.12	AGG	11559
20	0.5	TL	0.28	†	†	AGG	TL	0.12	TL	0.33	AGG	12905
20	1.0	TL	0.09	TL	0.02	DIS	TL	0.02	TL	0.13	AGG	12970
22	0.5	TL	0.91	TL	1.0	AGG	TL	0.72	TL	0.72	—	15125
22	1.0	TL	0.10	TL	0.28	AGG	TL	0.01	TL	0.01	—	15327
24	0.5	TL	0.83	†	†	AGG	TL	0.64	TL	0.74	AGG	16278
24	1.0	TL	0.05	TL	0.08	AGG	TL	0.09	TL	0.11	AGG	16437
26	0.5	TL	1.1	†	†	AGG	TL	0.87	TL	0.91	AGG	17816
26	1.0	TL	0.17	†	†	AGG	TL	0.09	TL	0.11	AGG	18007
28	0.5	TL	0.89	†	†	AGG	TL	0.60	TL	0.86	AGG	18735
28	1.0	TL	0.16	†	†	AGG	TL	0.08	TL	0.08	—	18916
30	0.5	TL	1.1	†	†	AGG	TL	0.94	TL	0.96	AGG	20653
30	1.0	TL	0.11	†	†	AGG	TL	0.05	TL	0.05	—	20854

Note. Wall-clock time (s) and MIP gap (%) at timeout; “—” means solved to optimality. Obj is the lazy_MIP (AGG) incumbent. “Better” follows the convention of Table 5.2. †: solver degradation (see text).

Table 5.9: The cutplane and lazy configurations under the disaggregated formulation (DIS), $n \leq 20$.

n	α	cutplane (DIS)			lazy (DIS)		
		Time	Gap	Cuts	Time	Gap	Cuts
2	0.5	0.01	—	0	0.01	—	1
2	1.0	0.01	—	1	0.02	—	2
4	0.5	0.3	—	1	0.3	—	11
4	1.0	0.3	—	14	0.2	—	21
6	0.5	0.8	—	2	0.5	—	33
6	1.0	0.6	—	50	0.3	—	71
8	0.5	3.1	—	4	6.4	—	154
8	1.0	3.7	—	203	3.1	—	271
10	0.5	8.5	—	4	3.1	—	147
10	1.0	12	—	850	22	—	446
12	0.5	17	—	6	73	—	298
12	1.0	408	—	2586	218	—	987
14	0.5	231	—	23	101	—	147
14	1.0	117	—	12602	42	—	721
16	0.5	502	—	15	265	—	1029
16	1.0	TL	n/a	0	TL	0.04	2962
18	0.5	TL	n/a	0	TL	0.66	566
18	1.0	TL	n/a	0	TL	0.03	5952
20	0.5	TL	n/a	0	TL	0.63	2150
20	1.0	TL	n/a	0	TL	0.13	6302

Note. Completes the configurations announced in Section 5.5; the AGG counterparts appear in Table 5.3. Wall-clock time (s), MIP gap (%) at timeout (“—” means solved to optimality; “n/a” means no single MIP bound is available), and number of stability cuts.

Table 5.10: Diagnostics of the SCS-feasible heuristic (lazy_MIP_ws, AGG).

n	α	Phase 1 (s)	Phase 2 (s)	Core checks	Pool $ \mathcal{P} $	Heur. Obj	$ CS $	avg. $ c $
2	0.5	0.05	0.00	0	2	1340	2	1.0
2	1.0	0.00	0.00	1	3	1365	1	2.0
4	0.5	0.02	0.00	2	6	3037	3	1.3
4	1.0	0.03	0.00	9	13	3105	1	4.0
6	0.5	0.04	0.00	9	15	3622	4	1.5
6	1.0	0.13	0.00	24	30	3696	1	6.0
8	0.5	0.08	0.00	11	19	5487	4	2.0
8	1.0	0.63	0.00	50	52	5600	2	4.0
10	0.5	0.18	0.00	15	25	7483	6	1.7
10	1.0	0.65	0.00	74	81	7564	2	5.0
12	0.5	0.17	0.00	35	47	8244	6	2.0
12	1.0	2.13	0.00	112	118	8325	2	6.0
14	0.5	0.23	0.01	41	55	8848	8	1.8
14	1.0	3.61	0.01	147	150	8940	4	3.5
16	0.5	0.50	0.01	59	75	10250	9	1.8
16	1.0	24.1	0.01	199	198	10340	2	8.0
18	0.5	0.68	0.01	75	92	11420	10	1.8
18	1.0	28.4	0.01	285	267	11549	3	6.0
20	0.5	1.38	0.01	115	129	12896	11	1.8
20	1.0	17.3	0.01	329	304	12963	5	4.0
22	0.5	1.82	0.01	125	144	15151	11	2.0
22	1.0	98.5	0.01	469	413	15339	3	7.3
24	0.5	2.23	0.02	98	121	16304	13	1.8
24	1.0	10.3	0.02	464	434	16428	5	4.8
26	0.5	2.94	0.02	207	217	17862	13	2.0
26	1.0	106	0.06	725	564	18014	5	5.2
28	0.5	3.57	0.01	160	185	18776	14	2.0
28	1.0	65.5	0.02	725	614	18930	6	4.7
30	0.5	4.46	0.02	210	237	20676	17	1.8
30	1.0	60.1	0.02	774	708	20866	6	5.0

Note. Phase 1 is the Core-verified bottom-up merging (Section 5.4.3) and Phase 2 the set-partitioning IP over the collected pool $\mathcal{P}$ of Core-feasible coalitions; “Core checks” counts Core-nonemptiness LPs solved in Phase 1. Heur. Obj is the objective of the returned SCS-feasible partition ($|CS|$ coalitions of average size avg. $|c|$), which seeds lazy_MIP_ws. Phase 2 is negligible throughout because the pool sizes remain in the hundreds; total heuristic time is dominated by the Core checks on large pooled coalitions at $\alpha = 1.0$.

5.G Coalition Structure and Payoff Details

5.G.1 Per-coalition solution concepts: details

For each coalition $c \in CS(N)$, the restricted game (c, V_c) admits the usual cooperative solution concepts: (i) the Shapley value is well-defined for (c, V_c) ; (ii) the nucleolus of (c, V_c) exists and lies in its Core whenever the Core is nonempty, as is the case for every formed coalition of an SCS; and (iii) if (c, V_c) is convex (supermodular), then the Shapley value of (c, V_c) also belongs to its Core. Non-uniqueness of the OSCS payoffs is harmless: replacing the returned allocation $v^{(c)}$ by any other Core allocation $w^{(c)}$ of (c, V_c) preserves efficiency $\sum_{i \in c} w_i^{(c)} = V(c)$ and all internal Core inequalities $\sum_{i \in s} w_i^{(c)} \geq V(s)$, so the SCS constraints remain satisfied. The least-core value is

$$\varepsilon(c) := \min_{\varepsilon, \psi} \left\{ \varepsilon : \sum_{i \in c} \psi_i = V(c), \sum_{i \in s} \psi_i \geq V(s) - \varepsilon \quad \forall \emptyset \neq s \subsetneq c \right\},$$

where the quantification is over proper nonempty subsets; the Core of (c, V_c) is nonempty if and only if $\varepsilon(c) \leq 0$, and allocations attaining $\varepsilon(c) < 0$ lie in the relative interior of the Core.

Table 5.11 reports the optimal coalition structures found by OCS and OSCS for all small instances ($n \leq 12$), illustrating the structural patterns discussed in Section 5.5. Table 5.12 reports the corresponding per-coalition payoff refinement: the Shapley value and the nucleolus of each formed OSCS coalition of size at least two.

Table 5.11: Coalition structures for small instances ($n \leq 12$).

n	α	OCS (AGG)			OSCS (lazy_MIP_ws, AGG)		
		Obj	$ CS $	Coalition Structure	Obj	$ CS $	Coalition Structure
2	0.5	1340	2	$\{0\}, \{1\}$	1340	2	$\{0\}, \{1\}$
2	1.0	1365	1	$\{0, 1\}$	1365	1	$\{0, 1\}$
4	0.5	3037	3	$\{0, 3\}, \{1\}, \{2\}$	3037	3	$\{0, 3\}, \{1\}, \{2\}$
4	1.0	3105	1	$\{0, 1, 2, 3\}$	3105	1	$\{0, 1, 2, 3\}$
6	0.5	3622	4	$\{0, 1\}, \{2\}, \{3\}, \{4, 5\}$	3622	4	$\{0, 1\}, \{2\}, \{3\}, \{4, 5\}$
6	1.0	3696	1	$\{0, 1, 2, 3, 4, 5\}$	3696	1	$\{0, 1, 2, 3, 4, 5\}$
8	0.5	5487	4	$\{0, 5\}, \{1, 3\}, \{2, 6\}, \{4, 7\}$	5487	4	$\{0, 5\}, \{1, 3\}, \{2, 6\}, \{4, 7\}$
8	1.0	5609	1	$\{0, 1, 2, 3, 4, 5, 6, 7\}$	5606*	2	$\{0, 2, 4, 7\}, \{1, 3, 5, 6\}$
10	0.5	7483	6	$\{0\}, \{1, 3\}, \{2, 9\}, \{4, 7\}, \{5, 6\}, \{8\}$	7483	6	$\{0\}, \{1, 3\}, \{2, 9\}, \{4, 7\}, \{5, 6\}, \{8\}$
10	1.0	7564	1	$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$	7564	2	$\{0, 2, 4, 5, 6, 7, 8, 9\}, \{1, 3\}$
12	0.5	8244	6	$\{0, 7\}, \{1, 6\}, \{2, 9\}, \{3, 5\}, \{4, 8\}, \{10, 11\}$	8244	6	$\{0, 7\}, \{1, 6\}, \{2, 9\}, \{3, 5\}, \{4, 8\}, \{10, 11\}$
12	1.0	8325	1	$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$	8325	2	$\{0, 1, 3, 4, 5, 6, 8, 10, 11\}, \{2, 7, 9\}$

*Stability constraint reduces objective value.

Note. OCS: optimal coalition structure (AGG, no stability). OSCS: optimal stable coalition structure (lazy_MIP_ws, AGG).

Table 5.12: Per-coalition payoff refinement for small instances ($n \leq 12$).

n	α	$V(c)$	Coalition	Shapley ϕ	Nucleolus	$\phi \in \text{Core}$	ε
2	1.0	1365	{0,1}	(582,784)	(582,784)	Yes	-12.50
4	0.5	1596	{0,3}	(704,892)	(704,892)	Yes	-17.00
4	1.0	3105	{0,1,2,3}	(709,745,740,911)	(709,742,739,916)	Yes	-5.33
6	0.5	1364	{0,1}	(798,566)	(798,566)	Yes	-2.00
6	0.5	1077	{4,5}	(692,385)	(692,385)	Yes	-10.00
6	1.0	3696	{0,1,2,3,4,5}	(809,585,627,594,699,383)	(806,588,626,598,699,380)	No	-0.67
8	0.5	1507	{0,5}	(743,764)	(743,764)	Yes	-6.00
8	0.5	1333	{1,3}	(614,719)	(614,719)	Yes	-2.00
8	0.5	1198	{2,6}	(567,631)	(567,631)	Yes	-11.00
8	0.5	1449	{4,7}	(690,759)	(690,759)	Yes	-18.00
8	1.0	2790	{0,2,4,7}	(761,566,693,771)	(763,560,691,776)	No	-4.00
8	1.0	2816	{1,3,5,6}	(639,751,786,640)	(637,756,789,633)	Yes	-9.20
10	0.5	1578	{1,3}	(846,732)	(846,732)	Yes	-6.50
10	0.5	1355	{2,9}	(602,752)	(602,752)	Yes	-4.50
10	0.5	1450	{4,7}	(696,754)	(696,754)	Yes	-19.50
10	0.5	1487	{5,6}	(817,670)	(817,670)	Yes	-20.00
10	1.0	1602	{1,3}	(858,744)	(858,744)	Yes	-18.50
10	1.0	5962	{0,2,4,5,6,7,8,9}	(892,606,691,819,679,764,745,766)	(892,606,691,820,680,766,743,763)	No	0.00
12	0.5	1312	{0,7}	(636,676)	(636,676)	Yes	-5.50
12	0.5	1331	{1,6}	(796,535)	(796,535)	Yes	-18.00
12	0.5	1370	{2,9}	(659,711)	(659,711)	Yes	-16.00
12	0.5	1284	{3,5}	(672,612)	(672,612)	Yes	-11.00
12	0.5	1447	{4,8}	(662,784)	(662,784)	Yes	-25.50
12	0.5	1500	{10,11}	(706,794)	(706,794)	Yes	-4.50
12	1.0	2052	{2,7,9}	(663,680,708)	(668,676,707)	Yes	-5.67
12	1.0	6273	{0,1,3,4,5,6,8,10,11}	(644,809,673,653,624,547,805,711,806)	(640,811,672,653,624,550,807,710,806)	No	-0.67

Note. OSCP coalitions of size ≥ 2 . Payoff vectors are ordered by player index within each coalition and rounded to integers. ε is the least-core value ($\varepsilon \leq 0$ iff the Core is nonempty); $\phi \in \text{Core}$ indicates whether the Shapley value satisfies all Core inequalities.

Chapter 6

Conclusion

This dissertation studied multi-agent decision problems in which each player solves an integer program, and it advanced a single thesis across four chapters: the equilibrium of an integer programming game is a computable object that can be scaled, leveraged, generalized, and extended to cooperation. This concluding chapter states the methodological contributions collectively, reports what the Minnesota application delivered, and sets out the limitations that define the next round of work.

Methodological contributions. The dissertation develops a unified optimization framework for computing, selecting, improving, and stabilizing outcomes in multi-agent integer programming games. Across the non-cooperative, generalized, and cooperative settings, the central methodological contribution is the integration of best-response oracles with globally valid inequalities inside branch-and-cut methods that optimize directly *over* equilibrium and stability sets, rather than merely locating one feasible point in them. Four components realize this idea.

Scale. Modeling best-response play as a random walk on the best-response state graph and adding round-based random restarts yields RRR-BRD, whose Las Vegas guarantee makes the probability of reaching a pure Nash equilibrium tend to one as restarts accumulate. Hybridizing it with the zero-regret cutting-plane method yields BZR, which feeds high-quality incumbents to the exact solver and lifts tractable equilibrium computation from

the small instances of the prior literature to games with dozens of players and thousands of integer variables. On the Minnesota AIS game with 84 county players, BZR located multiple equilibria where the exact baseline found none within the time budget.

Leverage. Nash-informed planning treats a computed equilibrium not as a verdict but as an input: equilibrium payoffs become individual-rationality floors, and welfare is maximized above them. The welfare cost of that participation guarantee is named and priced—the price of individual rationality—and proved to be bounded by the price of stability when reference utilities are drawn from the welfare-best equilibrium. The construction requires neither a redesign of player utilities nor monetary transfers, which is what makes it usable where institutional structure is fixed by statute.

Generality. Conditional equilibrium inequalities encode the no-regret condition under shared constraints through an explicit activation mechanism, giving the first globally valid equilibrium inequalities for generalized integer programming games and a generalized zero-regret algorithm that optimizes over the generalized equilibrium set. Where exact equilibria may not exist, an α -bisection procedure returns certified approximate equilibria; where the conditional polytopes are well structured, the vertex equilibrium structure theorem converts that structure into optimality-preserving cuts. On the generalized knapsack benchmark the method finds a PNE for all 1,260 instances within ten minutes and certifies social optimality for all but one, against 30–50% termination within one hour for the prior intersection-cut approach, and it scales to 50 players and to a bilinear-cost extension; on regular integer splittable bin packing the structural cuts deliver a $1.55\times$ speedup.

Cooperation. Defining coalition values through pooled integer programs turns the characteristic function into an optimization object, so superadditivity, convexity, and non-emptiness of the Core become instance-dependent rather than assumed. A stability inequality keeps every formed coalition in its own Core, and lifted stability cuts, separation strategies, a

stability-feasible primal heuristic, and a payoff-refinement step complete the method. Optimality is certified with up to 16 players and gaps below 1% are reached at 30 players, while only a vanishing fraction of the exponentially many coalition values is evaluated. Stability proves nearly free: optimal and stable coalition structures almost always coincide, and the nucleolus and, in most cases, the Shapley value remain (nearly) stable even when integrality voids the duality arguments that guarantee it in the linear case.

Application and policy contributions. The Minnesota aquatic invasive species application demonstrates how these methods support coordination among autonomous public agencies whose discrete resource-allocation decisions generate cross-jurisdictional externalities. Minnesota’s inspection effort is planned independently by counties operating under separate statutory budgets, and boat traffic carries the consequences of those decisions across county lines; the resulting edge-weighted budgeted maximum coverage game spans roughly 9,000 lakes, with 84 county players in the equilibrium-computation study and 38 strategically active counties at the baseline funding level.

Three findings are worth stating precisely. First, the decentralized outcome is far from the best achievable: at the baseline budget the counties’ reference equilibrium protects 220 infested lakes and the Nash-informed recommendation protects 302, an increase of roughly 37%, while guaranteeing every county at least its equilibrium payoff and requiring no inter-county transfers. Second, the welfare cost of that guarantee is negligible: the realized price of individual rationality stays within 0.15% of the unconstrained statewide optimum across funding scenarios from 50% to 125% of current appropriations, and the same near-unit behavior recurs across 28 non-trivial instances of three further game classes. Third, the gap that matters is not the one usually assumed: at baseline, fragmented budget authority costs more statewide protection than strategic decentralization does, which reframes the policy

question from how to make counties behave less selfishly to how to let them pool or redirect budget without surrendering autonomy.

Read as a whole, the dissertation argues that equilibria and stable coalition structures of integer programming games are not only *definable* but *operational*: they can be computed at applied scale and used as inputs to downstream decisions. For a state agency this yields levers it can actually pull, together with operational guidance on directing capacity toward high-impact infested source lakes and on which county coalitions are worth forming. The ingredients—an objective, a computed equilibrium or coalition value, and a participation or stability constraint—are not specific to invasive species, and carry over wherever autonomous agents control discrete resources whose use spills across their boundaries.

Limitations. Four limitations bound what has been established. The equilibrium-computation guarantees are probabilistic and conditional on existence: RRR-BRD cannot by itself certify nonexistence, so failure to find an equilibrium within a time budget is evidence about difficulty rather than a proof. The welfare guarantee of Nash-informed planning is bounded by the price of stability but otherwise observational, the near-unit price of individual rationality being an empirical regularity rather than a theorem. For generalized games, the unary lifting that keeps conditional inequalities linear confines the method to small integer demands and capacities, and certifying that no α -equilibrium exists is empirically much harder than finding one. For cooperative games, optimality is certified only up to moderate player counts, and the stability notion adopted here guards against fission but not fusion.

Future research directions. Seven directions follow from this work, four tied to individual chapters and three cutting across them.

Complexity of restart-based equilibrium search. The Las Vegas guarantee of Chapter 2 rests

on the per-attempt success probability $q_\mu[R]$, the probability that one RRR-BRD attempt under restart law μ reaches a pure Nash equilibrium within R rounds, and the guarantee is vacuous when $q_\mu[R] = 0$. Because the strategy space is finite, $q_\mu[R]$ is computable in principle: the best-response state graph can be enumerated and the set of profiles from which a pure Nash equilibrium is reachable within R rounds traced exactly. The open questions are therefore about cost and structure rather than about decidability. What is the computational complexity of deciding $q_\mu[R] > 0$, and of computing $q_\mu[R]$ itself, when the game is given implicitly by the players' integer programs rather than by a state graph whose size is exponential in the number of players? Can $q_\mu[R]$ be approximated within a useful factor by sampling with provable guarantees, rather than estimated as we do? And can the smallest round cap R attaining a target success probability be determined without enumerating the graph? Answers would replace the current empirical estimate $\tilde{q}_\mu[R]$ with an a priori stopping rule.

Structural characterization of $q_\mu[R]$. Even where the general question is hard, the game classes of Chapter 2 may admit exact answers. Characterizing $q_\mu[R]$ —or bounding it in terms of instance parameters—for edge-weighted budgeted maximum coverage games and knapsack problem games would exploit their specific structure (submodular coverage on a directed network; separable knapsack feasibility) and would convert the convergence guarantee into a runtime bound stated in the data of the instance rather than in an unknown probability.

Sharper price of individual rationality, and mechanism design. Chapter 3 proves only that the price of individual rationality is bounded by the price of stability, while the observed values sit far below that bound. An analytical characterization of the realized price for well-structured or special IPG classes—identifying the network and budget conditions under which it provably stays near one, and constructing the classes where it can be large—is the natural next step. Beyond that, Nash-informed planning currently takes utilities and

institutions as fixed; extending it toward algorithmic mechanism design would ask what recommendation mechanisms can be designed, with provable participation and welfare guarantees, when the planner is allowed limited instruments.

Mixed-integer strategies and large-scale generalized games. The conditional equilibrium inequalities of Chapter 4 are developed for pure-integer strategy spaces, and extending them to mixed-integer strategies would cover a substantially larger class of applications. Scale is the second obstacle: the one-hot encoding that keeps the lifted inequalities linear grows with the integer demands and capacities, so more economical encodings of the activation mechanism, or a reformulation that avoids the encoding altogether, are what stand between the present method and large-scale generalized integer programming games.

Broader and larger cooperative games. The cooperative framework of Chapter 5 defines coalition values through pooled integer programs, and nothing in the construction requires those programs to be knapsacks. Instantiating it on richer combinatorial problems—vehicle routing, the traveling salesman problem, and inventory and lot-sizing models, where a coalition’s value is the optimal value of a hard combinatorial program—would test both the modeling framework and the separation machinery. In parallel, raising the player count at which optimality can be certified, through branch-and-price with on-the-fly coalition generation or stronger inequalities in the assignment and item variables, would move the method toward realistic consortium sizes.

Uncertainty. All four chapters treat payoffs, budgets, and interaction networks as known. Extending integer programming games to stochastic formulations, where players optimize expected utility over a distribution of parameters, and to robust or distributionally robust formulations, where they hedge against an ambiguity set, raises new questions at every level: what an equilibrium means under uncertainty, whether the equilibrium inequalities remain globally valid, and how the resulting formulations can be separated.

Dynamics in the application. The Minnesota study is single-period, whereas invasion is a process that unfolds over years and inspection budgets are appropriated annually. A multi-period formulation, or a multistage stochastic program in which infestation status evolves stochastically and capacity is re-allocated as it is observed, would capture the timing of prevention rather than only its placement, and would let the participation guarantee developed here be stated over a planning horizon instead of a single year.

Bibliography

- [1] Ben Amiet, Andrea Collevocchio, Marco Scarsini, and Ziwen Zhong. Pure Nash Equilibria and Best-response Dynamics in Random Games. *Mathematics of Operations Research*, 46(4):1552–1572, 2021.
- [2] Nicholas R Angell, Alex W Bajcz, Reuben P Keller, Amy C Kinsley, Nels Snellgrove, Ulrich Muellner, Petra Muellner, and Nicholas B D Phelps. AIS Explorer: Intervention impact—an application for planning more cost-effective AIS prevention programs. *Biological Invasions*, 27(11):234, 2025.
- [3] Elliot Anshelevich, Anirban Dasgupta, Jon Kleinberg, Éva Tardos, Tom Wexler, and Tim Roughgarden. The price of stability for network design with fair cost allocation. *SIAM Journal on Computing*, 38(4):1602–1623, 2008.
- [4] Krzysztof R Apt and Tadeusz Radzik. Stable partitions in coalitional games. *arXiv preprint cs/0605132*, 2006.
- [5] Krzysztof R Apt and Sunil Simon. A classification of weakly acyclic games. *Theory and Decision*, 78(4):501–524, 2015.
- [6] Krzysztof R Apt and Andreas Witzel. A generic approach to coalition formation. *International game theory review*, 11(03):347–367, 2009.
- [7] R Pablo Arribillaga and Gustavo Bergantiños. Cooperative and axiomatic approaches to the knapsack allocation problem. *Annals of Operations Research*, 318(2):805–830, 2022.

- [8] Itai Ashlagi, Dov Monderer, and Moshe Tennenholtz. On the value of correlation. *Journal of Artificial Intelligence Research*, 33:575–613, 2008.
- [9] Itai Ashlagi, Felix Fischer, Ian A. Kash, and Ariel D. Procaccia. Mix and match: A strategyproof mechanism for multi-hospital kidney exchange. *Games and Economic Behavior*, 91:284–296, 2015.
- [10] Atalay Atasu, Miklos Sarvary, and Luk N. Van Wassenhove. Remanufacturing as a marketing strategy. *Management Science*, 54(10):1731–1746, 2008.
- [11] Atalay Atasu, L. Beril Toktay, and Luk N. Van Wassenhove. How collection cost structure drives a manufacturer’s reverse channel choice. *Production and Operations Management*, 22(5):1089–1102, 2013.
- [12] Robert J. Aumann. Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, 1(1):67–96, 1974.
- [13] Robert J. Aumann. Correlated equilibrium as an expression of Bayesian rationality. *Econometrica*, 55(1):1–18, 1987.
- [14] Robert J Aumann and Jacques H Drèze. Cooperative games with coalition structures. *International Journal of Game Theory*, 3(4):217–237, 1974.
- [15] Fernando Bernstein and Awi Federgruen. Decentralized supply chains with competing retailers under demand uncertainty. *Management Science*, 51(1):18–29, 2005.
- [16] Dimitris Bertsimas, Vivek F. Farias, and Nikolaos Trichakis. The price of fairness. *Operations Research*, 59(1):17–31, 2011.
- [17] Dimitris Bertsimas, Vivek F. Farias, and Nikolaos Trichakis. On the efficiency-fairness trade-off. *Management Science*, 58(12):2234–2250, 2012.

- [18] Vittorio Bilò. On the packing of selfish items. In *Proceedings of the 20th IEEE International Parallel & Distributed Processing Symposium*, 2006.
- [19] Peter Borm, Herbert Hamers, and Ruud Hendrickx. Operations research games: A survey. *Top*, 9(2):139–199, 2001.
- [20] Yossi Bukchin and Eran Hanany. Decentralization cost in scheduling: A game-theoretic approach. *Manufacturing & Service Operations Management*, 9(3):263–275, 2007.
- [21] Sabah Bushaj, İ. Esra Büyüктаhtakın, and Robert G. Haight. Risk-averse multi-stage stochastic optimization for surveillance and operations planning of a forest insect infestation. *European Journal of Operational Research*, 299(3):1094–1110, 2022. doi: 10.1016/j.ejor.2021.08.035.
- [22] İ. Esra Büyüктаhtakın. Stage- t scenario dominance for risk-averse multi-stage stochastic mixed-integer programs. *Annals of Operations Research*, 309(1):1–35, 2022. doi: 10.1007/s10479-021-04388-3.
- [23] İ. Esra Büyüктаhtakın. Scenario-dominance to multi-stage stochastic lot-sizing and knapsack problems. *Computers & Operations Research*, 153:106149, 2023. doi: 10.1016/j.cor.2023.106149.
- [24] I Esra Büyüктаhtakın and Robert G Haight. A Review of Operations Research Models in Invasive Species Management: State of the Art, Challenges, and Future Directions. *Annals of Operations Research*, 271(2):357–403, 2018.
- [25] İ. Esra Büyüктаhtakın, Zhuo Feng, George Frisvold, Ferenc Szidarovszky, and Aaryn Olsson. A dynamic model of controlling invasive species. *Computers & Mathematics with Applications*, 62(9):3326–3333, 2011. doi: 10.1016/j.camwa.2011.08.037.

- [26] İ. Esra Büyüктаhtakın, Eyyüb Y. Kılış, Halil I. Cobuloglu, Gregory R. Houseman, and J. Tanner Lampe. An age-structured bio-economic model of invasive species management: Insights and strategies for optimal control. *Biological Invasions*, 17(9):2545–2563, 2015. doi: 10.1007/s10530-015-0893-4.
- [27] İ. Esra Büyüктаhtakın, Emmanuel des Bordes, and Eyyüb Y. Kılış. A new epidemics–logistics model: Insights into controlling the ebola virus disease in west africa. *European Journal of Operational Research*, 265(3):1046–1063, 2018. doi: 10.1016/j.ejor.2017.08.037.
- [28] İ. Esra Büyüктаhtakın, J. Cole Smith, and Joseph C. Hartman. Partial objective inequalities for the multi-item capacitated lot-sizing problem. *Computers & Operations Research*, 91:132–144, 2018. doi: 10.1016/j.cor.2017.11.006.
- [29] İsmet Esra Büyüктаhtakın, Zhuo Feng, George Frisvold, and Ferenc Szidarovszky. Invasive Species Control Based on a Cooperative Game. *Applied Mathematics*, 4: 54–59, 2013.
- [30] Gérard P. Cachon. Supply chain coordination with contracts. In Stephen C. Graves and A. G. de Kok, editors, *Handbooks in Operations Research and Management Science: Supply Chain Management*, volume 11, pages 227–339. Elsevier, Amsterdam, 2003.
- [31] Gérard P. Cachon and Serguei Netessine. Game theory in supply chain analysis. In David Simchi-Levi, S. David Wu, and Z. Max Shen, editors, *Handbook of Quantitative Supply Chain Analysis: Modeling in the E-Business Era*, pages 13–65. Springer, Boston, MA, 2004.
- [32] Alberto Caprara and Adam N Letchford. New techniques for cost sharing in combinatorial optimization games. *Math. Programming*, 124(1–2):93–118, 2010.

- [33] M. Carvalho. *Computation of equilibria on integer programming games*. PhD thesis, Universidade do Porto, Faculdade de Ciências, 2016.
- [34] Margarida Carvalho, Andrea Lodi, Joao Pedro Pedroso, and Ana Viana. Nash Equilibria in the Two-player Kidney Exchange Game. *Mathematical Programming*, 161: 389–417, 2017.
- [35] Margarida Carvalho, Andrea Lodi, and Joao Pedro Pedroso. Existence of Nash Equilibria on Integer Programming Games. In A. Ismael F. Vaz, João Paulo Almeida, José Fernando Oliveira, and Alberto Adrego Pinto, editors, *Operational Research: IO2017, Valença, Portugal, June 28-30 XVIII*, pages 11–23. Springer, 2018.
- [36] Margarida Carvalho, Andrea Lodi, and João P Pedroso. Computing Equilibria for Integer Programming Games. *European Journal of Operational Research*, 303(3): 1057–1070, 2022.
- [37] Margarida Carvalho, Gabriele Dragotto, Andrea Lodi, and Sriram Sankaranarayanan. Integer Programming Games: A Gentle Computational Overview. In *Tutorials in Operations Research: Advancing the Frontiers of OR/MS: From Methodologies to Applications*, pages 31–51. INFORMS, 2023.
- [38] Margarida Carvalho, Gabriele Dragotto, Andrea Lodi, and Sriram Sankaranarayanan. The Cut-and-Play algorithm: Computing Nash equilibria via outer approximations. *Operations Research*, 74(2):1070–1086, 2026. Published online July 8, 2025.
- [39] Bugra Caskurlu, Vahan Mkrtchyan, Ojas Parekh, and K Subramani. On Partial Vertex Cover and Budgeted Maximum Coverage Problems in Bipartite Graphs. In Josep Diaz, Ivan Lanese, and Davide Sangiorgi, editors, *IFIP International Conference on Theoretical Computer Science*, pages 13–26. Springer, 2014.

- [40] Georgios Chalkiadakis, Edith Elkind, and Michael Wooldridge. *Computational aspects of cooperative game theory*. Morgan & Claypool Publishers, 2011.
- [41] Pierre Chardaire. The core and nucleolus of games: A note on a paper by Göthe-Lundgren et al. *Math. Programming*, 90(1):147–151, 2001.
- [42] André Chassein, Sven O. Krumke, and Clemens Thielen. Capacitated network design games with weighted players. *Networks.*, 68(2):141–158, September 2016. ISSN 0028-3045.
- [43] Chandra Chekuri and Amit Kumar. Maximum Coverage Problem with Group Budget Constraints and Applications. In *International Workshop on Randomization and Approximation Techniques in Computer Science*, pages 72–83. Springer, 2004.
- [44] Chen Chen, Wenbo Cai, İ Esra Büyüктаhtakın, and Robert G. Haight. A Game-theoretic Approach to Incentivize Landowners to Mitigate an Emerald Ash Borer Outbreak. *IIEE Transactions*, 56(11):1131–1145, 2024. doi: 10.1080/24725854.2023.2247466.
- [45] Ho-Lin Chen and Tim Roughgarden. Network design with weighted players. In *Theory of Computing Systems*, volume 45, pages 302–324, 2009.
- [46] George Christodoulou and Elias Koutsoupias. The Price of Anarchy of Finite Congestion Games. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 67–73, 2005.
- [47] Michele Conforti, Gérard Cornuéjols, and Giacomo Zambelli. *Integer Programming*. Springer, Cham, 2014. doi: 10.1007/978-3-319-11008-0.
- [48] Miguel Constantino, Xenia Klimentova, Ana Viana, and Abdur Rais. New insights on

- integer-programming models for the kidney exchange problem. *European Journal of Operational Research*, 231(1):57–68, nov 2013.
- [49] Tobias Crönert and Stefan Minner. Equilibrium Identification and Selection in Finite Games. *Operations Research*, 72(2):816–831, 2024.
- [50] Harlan Crowder, Ellis L Johnson, and Manfred Padberg. Solving large-scale zero-one linear programming problems. *Oper. Res.*, 31(5):803–834, 1983.
- [51] Andreas Darmann and Christian Klamler. The Knapsack sharing game. *European Journal of Operational Research*, 238(2):457–462, 2014.
- [52] Constantinos Daskalakis, Paul W Goldberg, and Christos H Papadimitriou. The Complexity of Computing a Nash equilibrium. *Communications of the ACM*, 52(2):89–97, 2009.
- [53] Alberto Del Pia, Michael C. Ferris, and Carla Michini. Totally unimodular congestion games. *SIAM Journal on Discrete Mathematics*, 31(2):875–881, 2017.
- [54] Xiaotie Deng, Toshihide Ibaraki, and Hiroshi Nagamochi. Algorithmic aspects of the core of combinatorial optimization games. *Mathematics of Operations Research*, 24(3):751–766, 1999.
- [55] Shiman Ding and Philip M. Kaminsky. Centralized and decentralized warehouse logistics collaboration. *Manufacturing & Service Operations Management*, 22(4):812–831, 2020.
- [56] DNR. Minnesota Department of Natural Resources (DNR) Administrative Regions, June 2024. URL <https://www.dnr.state.mn.us/aboutdnr/regions.html>. Last accessed August 26, 2024.

- [57] DNR. Local Aquatic Invasive Species Prevention Aid, 2025. <https://www.dnr.state.mn.us/invasives/ais/prevention/index.html>.
- [58] Gabriele Dragotto and Rosario Scatamacchia. The Zero Regrets Algorithm: Optimizing over Pure Nash Equilibria via Integer Programming. *INFORMS Journal on Computing*, 2023.
- [59] David F. Drake and Stefan Spinler. OM Forum—Sustainable operations management: An enduring stream or a passing fancy? *Manufacturing & Service Operations Management*, 15(4):689–700, 2013.
- [60] Moshe Dror. Cost allocation: The traveling salesman, binpacking, and the knapsack. *Appl. Math. Comput.*, 35(2):191–207, 1990.
- [61] AloANGs Duguet, Tobias Harks, Martin Schmidt, and Julian Schwarz. Branch-and-cut for computing approximate equilibria of mixed-integer generalized nash games. *arXiv preprint arXiv:2511.03340*, 2025.
- [62] AloANGSs Duguet, Tobias Harks, Martin Schmidt, and Julian Schwarz. Branch-and-cut for mixed-integer generalized nash equilibrium problems. *arXiv preprint arXiv:2506.02520*, 2025.
- [63] Aloïs Duguet, Margarida Carvalho, Gabriele Dragotto, and Sandra Ulrich Ngueveu. Computing Approximate Nash Equilibria for Integer Programming Games. *Optimization Letters*, pages 1–25, 2025.
- [64] Ivan Eryganov, Radovan Šomplák, Vlastimír Nevrlý, Veronika Smejkalová, Dušan Hrabec, and Kjetil K Haugen. Application of cooperative game theory in waste management. *Chemical Engineering Transactions*, 2020.

- [65] Luis E Escobar, Sophie Mallez, Michael McCartney, Christine Lee, Daniel P Zielinski, Ratna Ghosal, Przemyslaw G Bajer, Carli Wagner, Becca Nash, Megan Tomamichel, et al. Aquatic Invasive Species in the Great Lakes Region: an Overview. *Reviews in Fisheries Science & Aquaculture*, 26(1):121–138, 2018.
- [66] Filippo Fabiani and Sergio Grammatico. Multi-vehicle automated driving as a generalized mixed-integer potential game. *IEEE Transactions on Intelligent Transportation Systems*, 21(3):1064–1073, 2020. doi: 10.1109/TITS.2019.2901505.
- [67] Alex Fabrikant and Christos H Papadimitriou. The Complexity of Game Dynamics: BGP Oscillations, Sink Equilibria, and Beyond. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 844–853, 2008.
- [68] Michal Feldman and Tom Ron. Capacitated network design games. In Maria Serna, editor, *Algorithmic Game Theory: 5th International Symposium, SAGT 2012, Barcelona, Spain, October 22–23, 2012, Proceedings*, volume 7615 of *Lecture Notes in Computer Science*, pages 132–143, Berlin, Heidelberg, 2012. Springer. ISBN 978-3-642-33996-7.
- [69] Aris Filos-Ratsikas, Yiannis Giannakopoulos, and Philip Lazos. The Pareto frontier of inefficiency in mechanism design. *Mathematics of Operations Research*, 47(2):923–944, 2022.
- [70] Dimitris Fotakis, Spyros Kontogiannis, and Paul Spirakis. Selfish unsplittable flows. *Theoretical Computer Science*, 348(2-3):226–239, 2005.
- [71] Antonio Frangioni and Claudio Gentile. Perspective cuts for a class of convex 0–1 mixed integer programs. *Math. Programming*, 106(2):225–236, 2006.
- [72] Margaretha Gansterer and Richard F Hartl. Collaborative vehicle routing: A survey. *Eur. J. Oper. Res.*, 268(1):1–12, 2018.

- [73] Michel Goemans, Vahab Mirrokni, and Adrian Vetta. Sink Equilibria and Convergence. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pages 142–151. IEEE, 2005.
- [74] Maud Göthe-Lundgren, Kurt Jörnsten, and Peter Värbrand. On the nucleolus of the basic vehicle routing game. *Math. Programming*, 72(1):83–100, 1996.
- [75] Daniel Granot. A generalized linear production model: A unifying model. *Mathematical Programming*, 34(2):212–222, 1986.
- [76] Zonghao Gu, George L Nemhauser, and Martin WP Savelsbergh. Lifted cover inequalities for 0-1 integer programs: Computation. *INFORMS J. Comput.*, 10(4):427–437, 1998.
- [77] Mario Guajardo and Mikael Rönnqvist. Operations research models for coalition structure in collaborative logistics. *Eur. J. Oper. Res.*, 240(1):147–159, 2015.
- [78] Mario Guajardo and Mikael Rönnqvist. A review on cost allocation methods in collaborative transportation. *Internat. Trans. Oper. Res.*, 23(3):371–392, 2016.
- [79] Oktay Günlük and Jeff Linderoth. Perspective reformulations of mixed integer nonlinear programs with indicator variables. *Math. Programming*, 124(1–2):183–205, 2010.
- [80] Gurobi Optimization. Gurobi Optimizer Reference Manual, 2024. URL <https://docs.gurobi.com/projects/optimizer/en/current/>.
- [81] Robert G Haight, Amy C Kinsley, Szu-Yu Kao, Denys Yemshanov, and Nicholas BD Phelps. Optimizing the Location of Watercraft Inspection Stations to Slow the Spread of Aquatic Invasive Species. *Biological Invasions*, 23(12):3907–3919, 2021.

- [82] Robert G Haight, Denys Yemshanov, Szu-Yu Kao, Nicholas BD Phelps, and Amy C Kinsley. A Bi-Level Model for State and County Aquatic Invasive Species Prevention Decisions. *Journal of Environmental Management*, 327:116855, 2023.
- [83] Rashida Hakim, Jason Milionis, Christos Papadimitriou, and Georgios Piliouras. Swim Till You Sink: Computing the Limit of a Game. In *International Symposium on Algorithmic Game Theory*, pages 205–222. Springer, 2024.
- [84] Bainian Hao and Carla Michini. Integer-splittable bin packing games. In *Proceedings of the 21st Conference on Web and Internet Economics (WINE)*, 2025.
- [85] Tobias Harks and Julian Schwarz. Generalized Nash equilibrium problems with mixed-integer variables. *Mathematical Programming*, 209:231–277, 2025.
- [86] John C Harsanyi. A new theory of equilibrium selection for games with complete information. *Games and Economic Behavior*, 8(1):91–122, 1995.
- [87] Joseph C. Hartman, İ. Esra Büyüктаhtakın, and J. Cole Smith. Dynamic-programming-based inequalities for the capacitated lot-sizing problem. *IIE Transactions*, 42(12):915–930, 2010. doi: 10.1080/0740817X.2010.504683.
- [88] Torsten Heinrich, Yoojin Jang, Luca Mungo, Marco Pangallo, Alex Scott, Bassel Tarbush, and Samuel Wiese. Best-Response Dynamics, Playing Sequences, and Convergence to Equilibrium in Random Games. *International Journal of Game Theory*, 52(3):703–735, 2023.
- [89] Matthew O Jackson. Mechanism theory. *Available at SSRN 2542983*, 2014.
- [90] Szu-Yu Zoe Kao, Eva A Enns, Megan Tomamichel, Adam Doll, Luis E Escobar, Huijie Qiao, Meggan E Craft, and Nicholas BD Phelps. Network Connectivity of Minnesota

- Waterbodies and Implications for Aquatic Invasive Species Prevention. *Biological Invasions*, 23(10):3231–3242, 2021.
- [91] Eyyüb Y Kılış and İ Esra Büyüктаhtakın. Optimizing Invasive Species Management: A Mixed-integer Linear Programming Approach. *European Journal of Operational Research*, 259(1):308–321, 2017.
- [92] Eyyüb Y Kılış, İ Esra Büyüктаhtakın, Robert G Haight, Najmaddin Akhundov, Kathleen Knight, and Charles E Flower. A Multistage Stochastic Programming Approach to the Optimal Surveillance and Control of the Emerald Ash Borer in Cities. *INFORMS Journal on Computing*, 33(2):808–834, 2021.
- [93] Amy C Kinsley, Robert G Haight, Nels Snellgrove, Petra Muellner, Ulrich Muellner, Meredith Duhr, and Nicholas B D Phelps. AIS Explorer: Prioritization for watercraft inspections—a decision-support tool for aquatic invasive species management. *Journal of Environmental Management*, 314:115037, 2022.
- [94] Amy C Kinsley, Alex W Bajcz, Robert G Haight, and Nicholas BD Phelps. Facilitating effective collaboration to prevent aquatic invasive species spread. *Biological Conservation*, 290:110449, 2024.
- [95] László Á. Kóczy. *Partition Function Form Games: Coalitional Games with Externalities*, volume 48 of *Theory and Decision Library C*. Springer, 2018.
- [96] Matthias Köppe, Christopher Thomas Ryan, and Maurice Queyranne. Rational Generating Functions and Integer Programming Games. *Operations Research*, 59(6):1445–1460, 2011.
- [97] Vijay Krishna. *Auction theory*. Academic press, 2009.

- [98] Hyunwoo Lee, Robert Hildebrand, Wenbo Cai, and İ Esra Büyüктаhtakın. Random-restart best-response dynamics for large-scale integer programming games and their applications. *arXiv preprint arXiv:2409.04078*, 2024. URL <https://arxiv.org/abs/2409.04078>. Invited for resubmission at *Management Science*.
- [99] MAISRC. AIS Explorer: Data-Driven Approach to Inform Decision Making, 2025. <https://www.aisexplorer.umn.edu/>, last accessed August 2025.
- [100] Michael Maschler, Bezalel Peleg, and Lloyd S Shapley. Geometric properties of the kernel, nucleolus, and related solution concepts. *Mathematics of Operations Research*, 4(4):303–338, 1979.
- [101] Akihiko Matsui. Best Response Dynamics and Socially Stable Strategies. *Journal of Economic Theory*, 57(2):343–362, 1992.
- [102] MDR. Minnesota Department of Revenue 2017, Aug 2017. URL https://www.revenue.state.mn.us/sites/default/files/2017-08/aispa_2018certification.pdf. Last accessed August 26, 2024.
- [103] Igal Milchtaich. Congestion Games with Player-specific Payoff Functions. *Games and Economic Behavior*, 13(1):111–124, 1996.
- [104] Vahab S Mirrokni and Alexander Skopalik. On the Complexity of Nash Dynamics and Sink Equilibria. In *Proceedings of the 10th ACM conference on Electronic Commerce*, pages 1–10, 2009.
- [105] Dov Monderer and Lloyd S Shapley. Potential Games. *Games and Economic Behavior*, 14(1):124–143, 1996.
- [106] Liying Mu, Bin Hu, A. Amarender Reddy, and Srinagesh Gavirneni. Negotiating

- government-to-government food importing contracts: A Nash bargaining framework. *Manufacturing & Service Operations Management*, 24(3):1681–1697, 2022.
- [107] John F. Nash. The bargaining problem. *Econometrica*, 18(2):155–162, 1950.
- [108] John F Nash. Equilibrium Points in N-person Games. *Proceedings of the National Academy of Sciences*, 36(1):48–49, 1950.
- [109] George L. Nemhauser and Laurence A. Wolsey. *Integer and Combinatorial Optimization*. Wiley, New York, 1988. doi: 10.1002/9781118627372.
- [110] Noam Nisan and Amir Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35(1–2):166–196, 2001.
- [111] Sevilay Onal, Najmaddin Akhundov, İ. Esra Büyüктаhtakın, Jennifer Smith, and Gregory R. Houseman. An integrated simulation-optimization framework to optimize search and treatment path for controlling a biological invader. *International Journal of Production Economics*, 222:107507, 2020. doi: 10.1016/j.ijpe.2019.09.028.
- [112] Sevilay Onal, Sabah Bushaj, Jennifer Smith, Gregory R. Houseman, and İ. Esra Büyüктаhtakın. Managing large-scale invasions: Simulation-optimization with gaussian dispersal kernels and stochastic seed establishment. *Journal of the Operational Research Society*, pages 1–25, 2026. doi: 10.1080/01605682.2026.2641676. Published online March 23, 2026.
- [113] Guillermo Owen. On the core of linear production games. *Mathematical Programming*, 9(1):358–370, 1975.
- [114] Okan Örsan Özener, Özlem Ergun, and Martin Savelsbergh. Allocating cost of service to customers in inventory routing. *Oper. Res.*, 61(1):112–125, 2013.

- [115] Christos Papadimitriou and Georgios Piliouras. Game Dynamics as the Meaning of a Game. *ACM SIGecom Exchanges*, 16(2):53–63, 2019.
- [116] Christos H. Papadimitriou and Tim Roughgarden. Computing correlated equilibria in multi-player games. *Journal of the ACM*, 55(3):14:1–14:29, 2008.
- [117] Bezalel Peleg and Peter Sudhölter. *Introduction to the theory of cooperative games*. Springer, 2007.
- [118] Georgia Perakis. The “Price of Anarchy” under nonlinear and asymmetric costs. *Mathematics of Operations Research*, 32(3):614–628, 2007.
- [119] Georgia Perakis and Guillaume Roels. The price of anarchy in supply chains: Quantifying the efficiency of price-only contracts. *Management Science*, 53(8):1249–1268, 2007.
- [120] Erica L. Plambeck and Terry A. Taylor. Implications of breach remedy and renegotiation design for innovation and capacity. *Management Science*, 53(12):1859–1871, 2007.
- [121] Ariel D. Procaccia and Moshe Tennenholtz. Approximate mechanism design without money. *ACM Transactions on Economics and Computation*, 1(4):18:1–18:26, 2013.
- [122] Talal Rahwan, Sarvapali D Ramchurn, Nicholas R Jennings, and Andrea Giovannucci. An anytime algorithm for optimal coalition structure generation. *Journal of artificial intelligence research*, 34:521–567, 2009.
- [123] Talal Rahwan, Tomasz P Michalak, Michael Wooldridge, and Nicholas R Jennings. Coalition structure generation: A survey. *Artificial Intelligence*, 229:139–174, 2015.
- [124] R. Tyrrell Rockafellar. *Convex Analysis*, volume 28 of *Princeton Mathematical Series*. Princeton University Press, Princeton, NJ, 1970.

- [125] J Ben Rosen. Existence and uniqueness of equilibrium points for concave n-person games. *Econometrica: Journal of the Econometric Society*, pages 520–534, 1965.
- [126] Robert W Rosenthal. A Class of Games Possessing Pure-strategy Nash Equilibria. *International Journal of Game Theory*, 2:65–67, 1973.
- [127] Tim Roughgarden. Algorithmic Game Theory. *Communications of the ACM*, 53(7):78–86, 2010.
- [128] Simone Sagratella. Computing All Solutions of Nash Equilibrium Problems with Discrete Strategy Sets. *SIAM Journal on Optimization*, 26(4):2190–2218, 2016.
- [129] Simone Sagratella. Algorithms for generalized potential games with mixed-integer variables. *Computational Optimization and Applications*, 68(3):689–717, 2017. doi: 10.1007/s10589-017-9927-4.
- [130] Simone Sagratella. On generalized Nash equilibrium problems with linear coupling constraints and mixed-integer variables. *Optimization*, 68(1):197–226, 2019. doi: 10.1080/02331934.2018.1545125.
- [131] Simone Sagratella, Marcel Schmidt, and Nathan Sudermann-Merx. The Noncooperative Fixed Charge Transportation Problem. *European Journal of Operational Research*, 284(1):373–382, 2020.
- [132] Tuomas Sandholm, Kate Larson, Martin Andersson, Onn Shehory, and Fernando Tohmé. Coalition structure generation with worst case guarantees. *Artificial intelligence*, 111(1-2):209–238, 1999.
- [133] Sriram Sankaranarayanan. Best-Response Algorithms for Lattice Convex Quadratic Simultaneous Games. *arXiv preprint arXiv:2405.07119*, 2024.

- [134] David Schmeidler. The nucleolus of a characteristic function game. *SIAM Journal on Applied Mathematics*, 17(6):1163–1170, 1969.
- [135] Stefan Schwarze and Oliver Stein. A Branch-and-Prune Algorithm for Discrete Nash Equilibrium Problems. *Computational Optimization and Applications*, 86(2):491–519, 2023.
- [136] Lloyd S Shapley. A value for n -person games. In Harold W Kuhn and Albert W Tucker, editors, *Contributions to the Theory of Games II*, pages 307–317. Princeton University Press, 1953.
- [137] Georgios C Stamtsis and István Erlich. Use of cooperative game theory in power system fixed-cost allocation. *IEE Proceedings-Generation, Transmission and Distribution*, 151(3):401–406, 2004.
- [138] L. Beril Toktay, Lawrence M. Wein, and Stefanos A. Zenios. Inventory management of remanufacturable products. *Management Science*, 46(11):1412–1426, 2000.
- [139] Adrian Vetta. Nash equilibria in competitive societies, with applications to facility location, traffic routing and auctions. In *Proceedings of the 43rd IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 416–425. IEEE, 2002.
- [140] LZ Wang, Liping Fang, and Keith W Hipel. Water resources allocation: a cooperative game theoretic approach. *Journal of Environmental Informatics*, 2(2):11–22, 2003.
- [141] Qin Wang, Jincan Zeng, Beibei Cheng, Minwei Liu, Guori Huang, Xi Liu, Gengsheng He, Shangheng Yao, Peng Wang, and Longxi Li. A cooperative game approach for optimal design of shared energy storage system. *Sustainability*, 16(17):7255, 2024.
- [142] Laurence A. Wolsey. *Integer Programming*. Wiley, New York, 1998.

- [143] Jian Yang. The weak core, partition-based universal stability, and their risk associations through a partial order. *Naval Research Logistics*, 2025. Forthcoming.
- [144] Xuecheng Yin and İ. Esra Büyüктаhtakın. A multi-stage stochastic programming approach to epidemic resource allocation with equity considerations. *Health Care Management Science*, 24(3):597–622, 2021. doi: 10.1007/s10729-021-09559-z.
- [145] Xuecheng Yin and İ. Esra Büyüктаhtakın. Risk-averse multi-stage stochastic programming to optimizing vaccine allocation and treatment logistics for effective epidemic response. *IIE Transactions on Healthcare Systems Engineering*, 12(1):52–74, 2022. doi: 10.1080/24725579.2021.1938298.
- [146] Chengcheng Zhai, Rodney Parker, Kurt Michael Bretthauer, Jorge Mejia, and Alfonso Pedraza-Martinez. Keep the water flowing: The hidden crisis of rural water management. *Manufacturing & Service Operations Management*, pages 1–17, 2026. Articles in Advance.