Best-Response Dynamics for Large-Scale Integer Programming Games with Applications to Aquatic Invasive Species Prevention

Hyunwoo Lee[1], Robert Hildebrand *[1], Wenbo Cai[2], and İ. Esra Büyüktahtakın[1]

[1]Grado Department of Industrial and Systems Engineering, Virginia Tech
[2]Department of Mechanical and Industrial Engineering, New Jersey Institute of Technology

Abstract

This paper presents a scalable algorithm for computing the best pure Nash equilibrium (PNE) in large-scale integer programming games (IPGs). While recent advances in IPG algorithms are extensive, existing methods are limited to a small number of players, typically $n = 2, 3$. Motivated by a county-level aquatic invasive species (AIS) prevention problem involving 84 players, we develop efficient and scalable algorithms that significantly extend the applicability of IPGs. Specifically, we propose the best-response dynamics incorporated zero-regret (BZR) algorithm, which leverages best-response dynamics (BRD) for rapid PNE identification and integrates BRD as a primal heuristic within the zero-regret (ZR) framework. This approach dramatically improves the scalability of IPG algorithms, allowing us to solve IPGs with up to 30 players in random datasets and the 84-player AIS prevention problem with Minnesota data. To model the AIS prevention problem, we introduce the edge-weighted budgeted maximum coverage (EBMC) game, a new class of IPG that has not been previously studied. We establish theoretical conditions for the existence of a PNE under both selfish and locally altruistic utility functions. Experimental results in EBMC games and knapsack problem games demonstrate that BZR significantly enhances ZR in both finding a PNE and identifying the best PNE, particularly in games with many players.

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

1 Introduction

Integer programming games (IPGs) stand at the crossroads of integer programming and non-cooperative game theory, focusing on games where players solve optimization models involving discrete decisions that influence each player's utility. IPGs are non-cooperative, complete-information games, meaning players are self-interested and have complete information about each other's objectives and strategies (Carvalho et al., 2022). A pure Nash Equilibrium (PNE), as defined by Nash Jr (1950), is a state in which no player can improve their utility by unilaterally changing their strategy. The social welfare is the total of all players' utilities. In this paper, we develop efficient algorithms for large-scale IPGs to: (i) compute multiple PNEs, and (ii) identify the best PNE in terms of social welfare.

*Corresponding author: rhil@vt.edu

Enumerating PNEs is an important task in non-cooperative games, as some equilibria may be more desirable than others. In particular, socially optimal PNE—the one maximizing total social welfare—is preferred in settings involving mediators or centralized decision-makers. Recent work in IPGs has introduced two key algorithms to compute socially optimal Nash equilibrium (NE) and enumerate Nash equilibria (NEs): one for PNEs (Dragotto and Scatamacchia, 2023) and another for mixed Nash equilibria (MNEs) (Crönert and Minner, 2024), though both are 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 ($n = 2, 3$) (Dragotto and Scatamacchia, 2023) or with smaller strategy spaces (Sagratella, 2016; Schwarze and Stein, 2023). However, many real-world decision-making problems involve many more players, often representing individual self-interested stakeholders in complex systems. Given the maturity and diversity of the existing IPG literature, it is timely to address large-scale instances that reflect such practical complexities. 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 demands scalable and efficient algorithmic approaches.

To address this challenge, we focus on best-response dynamics (BRD), a widely used technique to compute a PNE in normal-form games (Matsui, 1992; Heinrich et al., 2023). While many recent studies have focused on simultaneous-play IPGs, where all players select strategies simultaneously (Carvalho et al., 2023), it is important to recognize that a PNE obtained through sequential play, such as BRD, remains a valid PNE in the simultaneous-play setting. BRD can efficiently identify a PNE, but their outcomes depend on the initial strategy profile and terminate once a PNE is found. In contrast, the zero-regret (ZR) algorithm proposed by Dragotto and Scatamacchia (2023) offers an exact method to enumerate and optimize over PNEs but struggles to find any PNE in large-scale settings due to computational intractability. To overcome these shortcomings, we propose the best-response dynamics incorporated zero-regret (BZR) algorithm, which integrates BRD as a primal heuristic within the ZR framework. This hybrid approach significantly improves the scalability of the PNE computation, increasing the number of solvable players tenfold, from small-scale cases ($n = 2, 3$) with 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 prior optimization-based approaches to invasive species control (Büyüktahtakın and Haight, 2018; Chen et al., 2023; Kıbış et al., 2021) by introducing the EBMC games. The EBMC framework naturally emerges in decentralized, county-level AIS prevention efforts across US lakes. Previous studies have applied the EBMC formulation (Caskurlu et al., 2014) to model AIS inspection at the county (Haight et al., 2021) and state levels (Haight et al., 2023), while others have examined AIS spread across network graphs (Escobar et al., 2018). In EBMC games, players are represented by subsets of vertices on a network graph, and each player maximizes their edge coverage under budget constraints. We define two variations of 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 paper, offer a powerful and scalable framework for modeling and solving large-scale multi-agent problems.

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 (Crönert and Minner, 2024), fixed-charge transportation problems (Sagratella et al., 2020), critical node games (Dragotto and Scatamacchia, 2023; Dragotto et al., 2024), kidney exchange games (Carvalho et al., 2017), quadratic IPGs (qIPGs) (Sagratella, 2016; Schwarze and Stein, 2023), integer convex quadratic games, and knapsack problem games (KPGs) (Carvalho et al., 2021; Dragotto and Scatamacchia, 2023). These models typically represent player interactions using reciprocally bilinear terms, with either linear objectives (e.g., KPG) or quadratic objectives (e.g., qIPG).

Carvalho et al. (2023) provide a comprehensive overview of the IPG solution methods, categorizing them into six approaches. Among these, SGM (Carvalho et al., 2022), eSGM (Crönert and Minner, 2024), and CnP (Carvalho et al., 2021) approximate MNEs, while BM (Sagratella, 2016), ZR (Dragotto and Scatamacchia, 2023), and BnP (Schwarze and Stein, 2023) focus on PNEs. Notably, Carvalho et al. (2022) and Crönert and Minner (2024) 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 (Dragotto and Scatamacchia, 2023) is state-of-the-art for computing socially optimal PNEs via a cutting-plane approach, but it has been applied only to small-scale cases with two or three players.

Best-response dynamics (BRD) (Matsui, 1992) is a foundational and commonly-used method for computing PNE in normal-form games, where players take turns solving their best-response problems while keeping the strategies of the others fixed. Since the playing sequence can influence the outcome, the BRD may cycle if a fixed sequence is used. To mitigate this, randomized strategies have been explored in the algorithmic game theory (AGT) community (Amiet et al., 2021; Wiese and Heinrich, 2022). Recent results from Heinrich et al. (2023) show that BRD reliably find a PNE in potential games and, with sufficient randomization, can do so in non-potential games if a PNE exists. In the IPG context, Sankaranarayanan (2024) applies BRD to compute MNEs in two-player pure integer convex quadratic games, leveraging the occurrence of cycles during BRD.

In normal-form games with strategy sets that are considerably smaller than those in IPGs, several algorithms have been proposed to compute all PNEs (Wu et al., 2014; Corley, 2020), while metaheuristic methods (Elgers et al., 2019) have demonstrated scalability for moderately larger instances.

1.2 Computing NEs in IPGs and the Scalability Challenge

The IPG literature has grown rapidly, with an increasing emphasis on exact methods for equilibrium computation. However, most existing approaches remain limited in scalability. We review the leading algorithms for computing equilibria in IPGs and summarize their capabilities and the largest instances tested in Table 1. For a comprehensive overview, see the tutorial by Carvalho et al. (2023). Two key observations emerge from

Table 1. First, existing methods are restricted by either the number of players (n) or the number of integer variables per player (m), with $n \cdot m \leq 300$ in all reported cases. For example, the ZR algorithm fails to compute a PNE for KPG instances with $n = 3$ and $m = 100$. Second, BRD have not been used to compute pure strategy equilibria in IPGs, despite their success in computing mixed equilibria.

In contrast, many real-world IPG applications involve dozens of players and tens or hundreds of decision variables per player, rendering current methods impractical. To address this challenge of scalability, we propose algorithms based on BRD, which avoid exponential construction of joint strategy sets and scale more favorably with the size of the problem. We introduce two variants: clockwork-random BRD (CR-BRD) for quickly computing a PNE, and BZR for enumerating and optimizing over multiple PNEs.

The tutorial paper by Carvalho et al. (2023) organizes IPG algorithms using three key building blocks: 'approximate', 'play', and 'improve' phases. In the 'approximate' phase, an approximated strategy space $\tilde{\mathcal{X}}_c$ for each player c is selected. If $\tilde{\mathcal{X}}_c \subseteq \mathcal{X}_c$ (or $\tilde{\mathcal{X}}_c \supseteq \mathcal{X}_c$), i.e., $\tilde{\mathcal{X}}_c$ contains the original strategy space $\mathcal{X}_c$ (or possibly removes some strategies in $\mathcal{X}_c$), it is classified as an inner (or outer) approximation, respectively. The 'play' phase computes a tentative NE within this approximated space, and the 'improve' phase verifies whether it is a NE or refines the approximation. We analyze CR-BRD and BZR through the lens of these algorithmic building blocks in Section 2.3. Computational implications related to interaction terms and dimensionality are further discussed in Section 5.5.

Table 1: Summary of Algorithms for Computing NE in IPGs (Adapted From Table 1 in Carvalho et al. (2023)).

Method	NE Type	Approx	Play	Opt.	Enum.	Largest IPG Instances Tested	Reference
SGM	Mixed	Inner	FP	X	X	KPG: $n \leq 3, m \leq 100$	(Carvalho et al., 2022)
eSGM	Mixed	Inner	MIP	✓	✓	KPG: $n \leq 3, m \leq 60$	(Crönert and Minner, 2024)
CnP	Mixed	Outer	CP	X	X	KPG: $n \leq 3, m \leq 100$	(Carvalho et al., 2021)
BM	Pure	Outer	CP	X	✓	qIPG: $n \leq 6, m \leq 3$	(Sagratella, 2016)
ZR	Pure	Outer	MIP	✓	✓	KPG: $n \leq 3, m \leq 100$	(Dragotto and Scatamacchia, 2023)
BnP	Pure	Outer	MIP	X	✓	qIPG: $n \leq 3, m \leq 5$	(Schwarze and Stein, 2023)
CR-BRD	**Pure**	**Inner**	**BRD**	X	X	**KPG: $n \leq 30, m \leq 100$**	**Proposed in this paper**
BZR		**Both**	**MIP, BRD**	✓	✓	**EBMC†: $n \leq 30, m \leq 50$**	
BRM	Mixed	Inner	BRD	X	X	ICQG: $n \leq 5, m \leq 25$	(Sankaranarayanan, 2024)

Note. 'Approx' = 'approximate' phase, 'Play' = 'play' phase, 'Opt.' = ability to optimize over NEa, 'Enum.' = can enumerate NEs. FP = Feasibility Problem, MIP = Mixed-Integer Problem, CP = Complementarity Probability, KPG = Knapsack Problem Games, qIPG = Quadratic IPG, ICQG = Integer Convex Quadratic Games. n: the number of players, $m = m_c, c \in N$: the number of integer variables for each player c.

† For EBMC game with real-world data (Section 5.2), $n \leq 84$; m_c varies for each player c and $m_c \leq 100$.

1.3 Contribution

Our detailed contributions are summarized as follows:

- We introduce best-response dynamics (BRD) into the IPG literature for the first time and propose the *clockwork-random BRD* (CR-BRD) algorithm to efficiently compute a (single) PNE. We develop the *best-response dynamics incorporated zero-regret* (BZR) algorithm, which leverages CR-BRD's rapid PNE identification while optimizing over PNEs, by incorporating it as a primal heuristic within the zero-regret (ZR) framework to guide solution construction. We further introduce *PNE inequalities*—a subclass of equilibrium inequalities that are binding at PNEs—to tighten the dual bound. This

integrated approach significantly improves scalability, enabling the computation of a PNE, multiple PNEs, the best PNE (within a time limit), and tighter dual bounds compared to ZR. In sum, our BZR algorithm greatly improves ZR by combining a primal-search method (CR-BRD), dynamic generation of initial solutions via callback functions, and additional useful cuts (PNE inequalities).

- We introduce EBMC games, a novel class of non-cooperative integer programming games where players maximize edge coverage under budget constraints. We examine two variations of the players' objective functions: locally altruistic and selfish. We prove the existence of a PNE in locally altruistic games using potential game theory and establish conditions for the existence of a PNE in selfish games.
- We conduct extensive computational experiments to evaluate the scalability and effectiveness of the proposed CR-BRD and BZR algorithms. Both algorithms efficiently compute PNEs in synthetic and real-world EBMC game instances with up to 84 players, and in KPGs with up to 30 players—a broad class of resource-constrained IPGs. Across 135 total test cases (27 per each of the two EBMC variants and three KPG types) the CR-BRD computes a PNE in 126 out of 135 instances, typically within 10 seconds for small instances and under 2 minutes for larger ones. Regarding BZR vs. ZR, BZR finds at least one PNE in 126 instances, far outperforming the ZR algorithm, which succeeds in only 30. In total, BZR identifies 179 PNEs for EBMC games (vs. 37 for ZR) and 3,350 for KPGs (vs. 36). BZR also outperforms ZR in identifying the best PNE in 97 out of 135 cases, with 29 ties and no instance where ZR is superior. While BZR yields tighter dual bounds in games with many PNEs, ZR performs better when equilibria are few.
- We demonstrate the use of our CR-BRD and BZR approaches to a real-world EBMC problem involving 84 county planners within Minnesota's AIS inspection programs, providing managerial insights to both state and county decision-makers. Notably, all but one county (83 out of 84) achieve unintended mutual benefits from playing a non-cooperative game with complete information. For the state, we confirm the existence of a socially optimal PNE, which offers a stable and desirable solution for resource allocation.

This paper is organized as follows. Section 2.1 provides notation and definitions relevant to IPGs and PNEs. Section 2 presents the CR-BRD and BZR algorithms, detailing their functionality and their role in identifying and improving PNEs. Section 3 formally defines EBMC games as an example IPG, highlighting variations in utility functions among players. Section 4 establishes conditions for the existence of PNE within EBMC games. Section 5 presents computational experiments that demonstrate the effectiveness and generalizability of the BZR algorithm across both random and real-world datasets, as well as random knapsack game datasets representing broader classes of resource-constrained IPGs. Finally, Section 6 offers concluding remarks.

2 Best-response Dynamics Incorporated Zero-regret Algorithm

In Section 2.1, we introduce the fundamental definitions and concepts related to IPGs and PNEs. Section 2.2 provides a detailed description of our proposed algorithms. Section 2.3 characterizes these algorithms through the lens of the algorithmic building blocks introduced by Carvalho et al. (2023).

2.1 Integer Programming Games and Pure Nash Equilibrium

We introduce the fundamental definitions and concepts related to IPGs and PNEs. Following Köppe et al. (2011), an IPG is defined as a tuple $G = (N, (\mathcal{X}_c)_{c\in N}, (u_c)_{c\in N})$, where $N := \{1, \dots, n\}$ denotes the set of players, $\mathcal{X}_c = \{\mathbf{x}^c \in \mathbb{Z}^{m_c} : A^c\mathbf{x}^c \leq \mathbf{b}^c\}$ represents the pure-integer strategy set for player c with rational matrices A^c, vectors $\mathbf{b}^c$, and dimension m_c, and $u_c(\mathbf{x}^c, \mathbf{x}^{-c}) : \mathcal{X} \to \mathbb{R}$ is the utility function for player c, parametrized by the strategies $\mathbf{x}^{-c}$ of all other players. Here, $\mathcal{X} := \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$ denotes the joint strategy space. The utility function is often expressed as

$$u_c(\mathbf{x}^c, \mathbf{x}^{-c}) = \frac{1}{2}(\mathbf{x}^c)^\top Q^c\mathbf{x}^c + (D^c\mathbf{x}^{-c})^\top\mathbf{x}^c + \mathbf{d}^c\mathbf{x}^c, \tag{1}$$

where Q^c is a symmetric matrix (not necessarily positive semi-definite), $\mathbf{d}^c$ is a linear coefficient vector, and $(D^c\mathbf{x}^{-c})^\top\mathbf{x}^c$ captures inter-player interactions through the *reciprocally bilinear terms*. These terms can be linearized by introducing auxiliary variables, resulting in a lifted space as described in Dragotto and Scatamacchia (2023). This transformation is particularly useful when constructing the joint strategy set $\mathbf{x} \in \mathcal{X}$, which is crucial when analyzing socially optimal PNEs. Throughout the paper, we assume that appropriate auxiliary variables can be introduced for each IPG—for example, $\mathbf{z}$-variables in KPGs and $\mathbf{y}$-variables in EBMC games. In an IPG, players act non-cooperatively, each solving an individual integer programming problem to maximize their utility. The best-response problem for each player c is defined as:

$$\max_{\mathbf{x}^c}\{u_c(\mathbf{x}^c, \mathbf{x}^{-c}) : \mathbf{x}^c \in \mathcal{X}_c\}. \tag{2}$$

The standard assumption in IPGs is that the best-response problems (2) are tractable and solved to optimality. In our study, we focus on two IPG classes, EBMC games, which exemplify dense D_c and KPGs, which represent the sparse case of D_c.

A strategy profile, $\mathbf{x} = (\mathbf{x}^1, \dots, \mathbf{x}^n)$, represents a collection of all players' strategies. A strategy profile $\hat{\mathbf{x}}$ is said to be a PNE if, for each player $c \in N$, the strategy $\hat{\mathbf{x}}^c$ is a best response strategy to the other players' strategies $\hat{\mathbf{x}}^{-c}$, i.e., $u_c(\hat{\mathbf{x}}^c, \hat{\mathbf{x}}^{-c}) \geq u_c(\mathbf{x}^c, \hat{\mathbf{x}}^{-c})$, $\forall \mathbf{x}^c \in \mathcal{X}_c$. Carvalho et al. (2018) have proved that deciding if a general IPG has a PNE is $\sum_2^p$-complete problem. They have also showed that PNEs may not exist, even in simple two-player IPGs. We define the social welfare function as $\phi(\mathbf{x}) = \sum_{c\in N} u_c(\mathbf{x}^c, \mathbf{x}^{-c})$, and refer to the socially optimal strategy profile, or the optimal social welfare (OSW) solution, as $\bar{\mathbf{x}}^*_{\text{sw}} \in \arg\max_{\mathbf{x}\in\mathcal{X}}\{\phi(\mathbf{x}) : \mathbf{x} \in \mathcal{X}\}$. Let $\mathcal{S}_{\text{pne}}$ denote the set of all PNEs. Then, the *best PNE* is defined as $\hat{\mathbf{x}}^*_{\text{pne}} \in \arg\max_{\mathbf{x}\in\mathcal{X}}\{\phi(\mathbf{x}) : \mathbf{x} \in \mathcal{X} \cap \mathcal{S}_{\text{pne}}\}$. The *price of stability* (POS) (Roughgarden, 2010), a common measure of inefficiency in game theory, is defined as the ratio $\frac{\phi(\bar{\mathbf{x}}^*_{\text{sw}})}{\phi(\hat{\mathbf{x}}^*_{\text{pne}})}$. A larger POS indicates a larger gap in social welfare between the best PNE and the socially optimal solution. These notations are later recalled in Table 4.

2.2 Algorithm Details

Clockwork-random BRD (CR-BRD) algorithm We present a variant of the best-response dynamics (BRD), which we call the clockwork-random BRD algorithm (CR-BRD). While BRD operates in an intuitive manner and is therefore omitted from the main text, we provide two illustrative examples demonstrating its

mechanics in Appendix 1 for reference. This algorithm initially follows a clockwork playing sequence (Matsui, 1992), in which players sequentially update their strategies by solving their individual best-response problems (2) while assuming fixed strategies for other players. This deterministic sequence continues while tracking the sequence of strategy profiles. Upon detection of a cycle, the algorithm switches to randomized playing sequences to escape such cycles. The formal steps of the CR-BRD are presented in Algorithm 1.

Algorithm 1 CR-BRD Algorithm

Require: Player utilities u_c and Strategy set $\mathcal{X}_c$ for all $c \in N$, initial strategy profile $\bar{\mathbf{x}}_{\text{init}}$, maximum iterations T, maximum restarts L

1: Initialize $\bar{\mathbf{x}} \leftarrow \bar{\mathbf{x}}_{\text{init}}$, history $\mathcal{H} \leftarrow \emptyset$, update order N (clockwork), and `Random` $\leftarrow$ `False`
2: **for** $i = 1$ to L **do**
3: **for** $t = 1$ to T **do**
4: **for** each player c in N **do**
5: Compute best response: $\hat{\mathbf{x}}^c \in \arg\max_{\mathbf{x}^c} \{u_c(\mathbf{x}^c, \bar{\mathbf{x}}^{-c})\}$
6: Update $\bar{\mathbf{x}} \leftarrow (\hat{\mathbf{x}}^c, \bar{\mathbf{x}}^{-c})$
7: Append $\bar{\mathbf{x}}$ to $\mathcal{H}$
8: **if** consecutive profiles are identical **then return** $\bar{\mathbf{x}}$ ▷ PNE found
9: **if** `Random` is `True` **then** update $N \leftarrow$ *Randomized*(N)
10: **else if** any profile in $\mathcal{H}$ repeats (cycle detected) **then** set `Random` $\leftarrow$ `True`
11: $\bar{\mathbf{x}} \leftarrow$ *Random_Generation*$(\mathbf{x})$
12: **return** `None`

Remark 1. Assuming T and L are infinite—implying that CR-BRD operates as a fully randomized BRD in both initial strategy profiles and playing sequences—the algorithm converges to a PNE under two conditions: (i) when the IPG is a finite potential game, and (ii) when the IPG admits at least one PNE. For (i), the classical result of Monderer and Shapley (1996), which guarantees BRD convergence in potential games, directly applies to IPGs, since IPGs with pure-integer variables are representable in normal form (Carvalho et al., 2022). For (ii), even if the game is not a potential game, a fully randomized BRD effectively acts as an enumeration scheme, making convergence still valid. For example, CR-BRD could be extended to exhaustively generate all possible strategy profiles, sampling from those not yet visited in Line 11. However, this approach is computationally impractical for large-scale IPGs due to exponential memory requirements. Thus, our focus is on scalable and efficient algorithms for rapid PNE identification.

Randomized strategies in BRD have been extensively studied in algorithmic game theory (AGT) (Amiet et al., 2021; Wiese and Heinrich, 2022; Heinrich et al., 2023). Heinrich et al. (2023) show that BRD reliably finds PNEs in potential games regardless of the playing sequence, and with sufficient randomization, can also identify PNEs in non-potential games when they exist. Note that the result from Heinrich et al. (2023) cannot be directly applied in the IPGs because they consider a normal form game with random utilities, different from the IPGs where the utility comes from the utility function. While prior AGT work often centers on theoretical analysis with random utilities, our approach prioritizes a structured sequential BRD and selectively introduces randomized update orders only when cycles are detected, enhancing practical implementation without compromising interpretability.

Zero-regret (ZR) algorithm The zero-regret (ZR) algorithm computes the socially optimal PNE by dynamically adding *equilibrium inequalities*, which are valid for all PNEs. Given a solution $\bar{\mathbf{x}} \in \operatorname{argmax}\{\phi(\mathbf{x}) : \mathbf{x} \in \mathcal{X}\}$, a separation oracle verifies its equilibrium status by solving the best-response problem $\hat{\mathbf{x}}^c \in \operatorname{argmax}_{\mathbf{x}^c}\{u_c(\mathbf{x}^c, \bar{\mathbf{x}}^{-c}) : \mathbf{x}^c \in \mathcal{X}_c\}$ for each player c. If a deviation $\hat{\mathbf{x}}^c$ yields a higher utility, i.e., $u_c(\hat{\mathbf{x}}^c, \bar{\mathbf{x}}^{-c}) > u_c(\bar{\mathbf{x}}^c, \bar{\mathbf{x}}^{-c})$, the following equilibrium inequality is generated:

$$u_c(\hat{\mathbf{x}}^c, \mathbf{x}^{-c}) \leq u_c(\mathbf{x}^c, \mathbf{x}^{-c}), \tag{3}$$

where $\hat{\mathbf{x}}^c$ represents the best response with $\bar{\mathbf{x}}^{-c}$. Let Ω denote the container set for equilibrium inequalities, initially set to $\emptyset$. The inequality (3) is then added to the constraint set Ω, refining the feasible set to $\mathcal{X} \cap \Omega$. The process repeats until a socially optimal PNE is found. Dragotto and Scatamacchia (2023) show that the convex hull of PNEs is characterized by equilibrium inequalities (see Theorem 1 in (Dragotto and Scatamacchia, 2023)). Although effective for small-scale games (up to three players), the ZR algorithm struggles with large-scale IPGs, where solving the optimization becomes intractable within reasonable time limits. This limitation motivates the development of our proposed BZR algorithm.

Best-response Dynamics Incorporated Zero-Regret (BZR) algorithm We incorporate CR-BRD into the ZR algorithm as a primal heuristic to enhance the search for PNEs. While the original ZR framework adds cuts only after solving $\mathcal{Q} := \max\{\phi(\mathbf{x}) : \mathbf{x} \in \mathcal{X} \cap \Omega\}$, we implement a callback-based version of ZR, in which integer-feasible solutions are identified during the branch-and-bound process. Subsequently, from the output of a CR-BRD, if a PNE is found, we apply the following PNE inequalities for each player c:

$$u_c(\hat{\mathbf{x}}^c, \mathbf{x}^{-c}) \leq u_c(\mathbf{x}^c, \mathbf{x}^{-c}) \text{ where } \hat{\mathbf{x}} \in \mathcal{S}_{\text{pne}}. \tag{4}$$

The PNE inequalities (4) represent a subset of the equilibrium inequalities (3) that are satisfied—and thus binding—at any PNE. In comparison, a significant portion of the equilibrium inequalities (3) are not binding to any PNE, and hence less effective, and furthermore, a large number of these are created in the ZR algorithm, introducing computational burden. However, PNE inequalities may not be sufficient to fully describe the convex hull of PNEs. Consequently, additional equilibrium inequalities — those not dominated by PNE inequalities — may be necessary to eliminate non-equilibrium solutions with higher social welfare.

The key steps of BZR algorithm (Algorithm 2) are as follows:

- **Obtain an integer feasible solution and equilibrium check:** Integer-feasible solutions denoted as $\bar{\mathbf{x}}$ (Line 3), generated by the callback function, are passed to the separation oracle (Lines 4–7). The oracle checks if $\bar{\mathbf{x}}$ is a PNE. If not, equilibrium inequalities are added to remove $\bar{\mathbf{x}}$ from further consideration.
- **CR-BRD as primal heuristic:** If $\bar{\mathbf{x}}$ is not a PNE, it serves as an initial strategy profile for CR-BRD (Line 8). Since the callback function generates diverse solutions, this introduces randomness and variety in initial strategy selection, improving the discovery of multiple PNEs. If CR-BRD finds a new PNE, $\bar{\mathbf{x}}$, then PNE inequalities are added (Line 12). If a PNE provides an improved primal bound, we update the lower bound and set $\bar{\mathbf{x}}$ as $\hat{\mathbf{x}}^*_{\text{pne}}$ (Line 13).

Algorithm 2 Best-response Dynamics Incorporated Zero-regret (BZR) Algorithm

Require: Joint strategy set $\mathcal{X}$, Social welfare function ϕ, Player utilities u_c for all $c \in N$, Parameters T, L, `tol`
Ensure: Best pure Nash Equilibrium $\hat{\mathbf{x}}^*_{\text{pne}}$ or `None`

1: Initialize: $LB \leftarrow -\infty$, $UB \leftarrow \max\{\phi(\mathbf{x}) \mid \mathbf{x} \in \mathcal{X}_{\text{LP}}\}$, $\Omega \leftarrow \emptyset$, `PNE-TRUE` $\leftarrow$ `True`
2: **while** $UB - LB >$ `tol` and time limit not exceeded **do**
3: $\quad \bar{\mathbf{x}} \leftarrow$ incumbent solution from approximate solve of $\max\{\phi(\mathbf{x}) \mid \mathbf{x} \in \mathcal{X} \cap \Omega\}$ ▷ Time-limited IP solve
4: $\quad$ **for** $c \in N$ **do**
5: $\quad\quad$ Compute best response: $\hat{\mathbf{x}}^c \in \arg\max_{\mathbf{x}^c \in \mathcal{X}_c} \{u_c(\mathbf{x}^c, \bar{\mathbf{x}}^{-c})\}$
6: $\quad\quad$ **if** $u_c(\hat{\mathbf{x}}^c, \bar{\mathbf{x}}^{-c}) > u_c(\bar{\mathbf{x}}^c, \bar{\mathbf{x}}^{-c})$ **then**
7: $\quad\quad\quad$ Add $u_c(\hat{\mathbf{x}}^c, \mathbf{x}^{-c}) \leq u_c(\mathbf{x}^c, \mathbf{x}^{-c})$ to Ω and set `PNE-TRUE` $\leftarrow$ `False` ▷ Add an equilibrium inequality
8: $\quad$ **if not** `PNE-TRUE` **then** $\bar{\mathbf{x}} \leftarrow$ `CR-BRD`$(\bar{\mathbf{x}}, T, L)$ ▷ Apply Clockwork-random best-response dynamics
9: $\quad$ **if** `PNE-TRUE` **OR** $\bar{\mathbf{x}}$ is a pure Nash Equilibrium **then**
10: $\quad\quad$ **for** $c \in N$ **do**
11: $\quad\quad\quad$ **if** the PNE inequality for $\bar{\mathbf{x}}^c$ has not been added **then**
12: $\quad\quad\quad\quad$ Add $u_c(\bar{\mathbf{x}}^c, \mathbf{x}^{-c}) \leq u_c(\mathbf{x}^c, \mathbf{x}^{-c})$ to Ω ▷ Add a PNE inequality
13: $\quad\quad$ **if** $\phi(\bar{\mathbf{x}}) \geq LB$ **then** $\hat{\mathbf{x}}^*_{\text{pne}} \leftarrow \bar{\mathbf{x}}$ and $LB \leftarrow \phi(\bar{\mathbf{x}})$
14: $\quad UB \leftarrow \max\{\phi(\mathbf{x}) \mid \mathbf{x} \in \mathcal{X}_{\text{LP}} \cap \Omega\}$
15: **if** $\hat{\mathbf{x}}^*_{\text{pne}} \neq$ `None` **then return** $\hat{\mathbf{x}}^*_{\text{pne}}$ **else return** `None` ▷ Return best PNE found

- **Termination:** The upper bound is updated by solving $\max\{\phi(\mathbf{x}) \mid \mathbf{x} \in \mathcal{X}_{\text{LP}} \cap \Omega\}$, where $\mathcal{X}_{\text{LP}}$ is a linear programming relaxation of $\mathcal{X}$ (Line 14). The BZR algorithm terminates once the upper and lower bounds coincide, ensuring that either the best PNE is found or no PNE exists (Line 15).

We remark that computationally, in line 3 of the algorithm, we terminate within a given time limit since verifying optimality via dual bounds can be computationally expensive in large-scale IPGs. As a nice extension to the algorithm, it retains ZR's ability to optimize over and enumerate PNEs using Hamming distance cuts (Dragotto and Scatamacchia, 2023, Section 4.1).

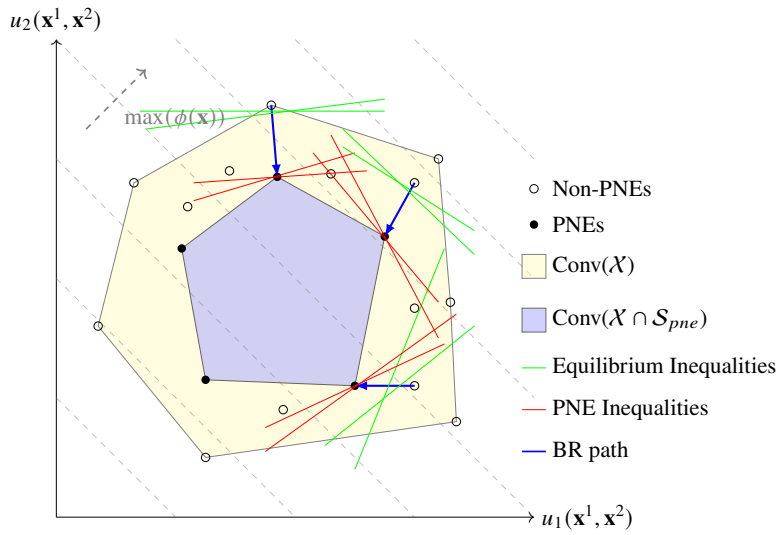


Figure 1: Algorithmic Behavior of BZR.

To illustrate the BZR algorithm's behavior, Figure 1 depicts the joint strategy space $\mathcal{X}$ for two players and the convex hull of PNEs. When a non-PNE is obtained through the callback function, equilibrium inequalities (green) are added to refine the feasible region. Following this, CR-BRD quickly identifies a PNE (blue), prompting the addition of PNE inequalities (red) to further tighten the solution space.

Our computational experiments show that CR-BRD significantly enhances performance in large-scale

IPGs. While equilibrium inequalities alone often fail to identify even a single PNE, the BZR algorithm—by incorporating CR-BRD—effectively discovers multiple PNEs and enriches the model with corresponding sets of PNE inequalities, as illustrated in Figure 1. The strength of these inequalities depends on both the number and quality of discovered PNEs: when many high-welfare PNEs exist, BZR refines the strategy space near the best one, accelerating dual bound improvement. However, if a unique PNE is far from the OSW, PNE inequalities offer limited benefit, and other mechanisms such as general-purpose cuts or non-dominated equilibrium inequalities must be used to improve the dual bound.

Example 1. To highlight the key differences between BZR and ZR, we present a simple two-player example. It illustrates how BZR, by focusing on verified best responses and strategy profiles that are already PNEs, efficiently identifies the best PNE with fewer equilibrium inequalities. In contrast, ZR explores a broader space and generates more cuts. The example also shows that the convex hull of PNEs can often be captured using only a small number of targeted inequalities.

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

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

Here, x_j^c denotes the binary decision variable of player $c \in \{1, 2\}$, corresponding to variable index $j \in \{1, 2, 3\}$. The bilinear terms $x_j^1 x_j^2$ can be linearized by introducing auxiliary variables z_j and the following constraints: $z_j \geq x_j^1 + x_j^2 - 1, z_j \leq x_j^1, z_j \leq x_j^2, \forall j \in \{1, 2, 3\}$ (see, e.g., Dragotto and Scatamacchia (2023)). Each player has a set of seven possible strategies: $\{(0,0,0) \vee (1,0,0) \vee (0,1,0) \vee (0,0,1) \vee (1,1,0) \vee (1,0,1) \vee (0,1,1)\}$, resulting in 49 possible strategy profiles. For example, the profile where both players select $(0, 0, 0)$ is denoted as $\mathbf{0}^N = \{(0,0,0), (0,0,0)\}$. This game has a unique PNE: $\hat{\mathbf{x}}^*_{\text{pne}} = \{(0,0,1), (1,1,0)\}$ with a social welfare of $\phi(\hat{\mathbf{x}}^*_{\text{pne}}) = 5$. We explain the steps of CR-BRD, BZR, and ZR on this example.

CR-BRD: Starting from $\mathbf{0}^N$ and following the playing sequence (1,2), CR-BRD converges to the PNE via the following best-response (BR) path: $\{(0,0,0), (0,0,0)\} \xrightarrow{p1} \{(1,1,0), (0,0,0)\} \xrightarrow{p2} \{(1,1,0), (1,1,0)\} \xrightarrow{p1} \{(0,0,1), (1,1,0)\}$ where 'p1' and 'p2' over the arrows indicate the player performing the BR update. One more iteration confirms this as the final strategy profile $\hat{\mathbf{x}}_{\text{pne}}$ for both players.

Continuing with BZR: The PNE inequalities are generated based on these best responses. Player 1's BR $(0, 0, 1)$ leads to the PNE inequality: $2 - 3x_3^2 \leq 3x_1^1 + 2x_2^1 + 2x_3^1 - 4z_1 - 5z_2 - 3z_3$. Player 2's BR $(1, 1, 0)$ leads to: $3 + 5x_1^1 + 2x_2^1 \leq 2x_1^2 + x_2^2 + 0x_3^2 + 5z_1 + 2z_2 + z_3$. After adding two PNE inequalities, the BZR algorithm terminates upon confirming via the separation oracle that the current solution $\{(0,0,1), (1,1,0)\}$ — which maximizes social welfare — is indeed the best PNE.

ZR: On the other hand, the ZR algorithm explores additional points. It first checks $\{(1,0,1), (1,1,0)\}$, which has a social welfare of 9, generating a violated equilibrium inequality from BR $(0, 0, 1)$ for player 1: $2 - 3x_3^2 \leq 3x_1^1 + 2x_2^1 + 2x_3^1 - 4z_1 - 5z_2 - 3z_3$. Next, it evaluates $\{(1,1,0), (1,0,1)\}$, with a social welfare of 8, generating two violated equilibrium inequalities from BR $(0, 1, 0)$ for player 1 and BR $(1, 1, 0)$ for player 2: $2 - 5x_2^2 \leq 3x_1^1 + 2x_2^1 + 2x_3^1 - 4z_1 - 5z_2 - 3z_3$, and $3 + 5x_1^1 + 2x_2^1 \leq 2x_1^2 + x_2^2 + 0x_3^2 + 5z_1 + 2z_2 + z_3$, respectively. Finally, the strategy profile $\{(0,0,1), (1,1,0)\}$ is tested and confirmed as the best PNE.

In comparison, the BZR algorithm requires one CR-BRD, resulting in two PNE inequalities and one separation, while the ZR algorithm requires three separation oracles with three equilibrium inequalities. The BZR algorithm is efficient because CR-BRD is computationally fast and PNE inequalities are generally useful to identify the best PNE. The performance gap between BZR and ZR increases significantly for large-scale IPGs, where solving Q becomes expensive and numerous separations may be required.

This example also illustrates that the convex hull of PNEs can be characterized by PNE inequalities combined with a minimal set of non-redundant equilibrium inequalities. Table 2 lists all strategy profiles and their corresponding BRs. Player 1 and player 2 have seven and two distinct BRs, respectively. However, not all corresponding equilibrium inequalities are needed to describe the convex hull of PNEs. As shown, the two PNE inequalities eliminate most non-equilibrium strategy profiles. The two remaining strategy profiles in the last row are eliminated by the equilibrium inequality with BR $(0, 1, 0)$ for player 1. To explicitly describe the convex hull, we first define $\mathcal{K} = \{\mathbf{x}^1 \in \{0,1\}^3,\ \mathbf{x}^2 \in \{0,1\}^3,\ \mathbf{z} \in \{0,1\}^3 : x_1^1 + x_2^1 + x_3^1 \leq 2,\ x_1^2 + x_2^2 + x_3^2 \leq 2,\ z_j \leq x_j^1,\ z_j \leq x_j^2,\ z_j \geq x_j^1 + x_j^2 - 1,\ \forall j \in \{1,2,3\}\}$. The convex hull of PNEs is then: $\text{conv}(\{(\mathbf{x}, \mathbf{z}) \in \mathcal{K} : 2 - 3x_3^2 \leq 3x_1^1 + 2x_2^1 + 2x_3^1 - 4z_1 - 5z_2 - 3z_3,\ 2 - 5x_2^2 \leq 3x_1^1 + 2x_2^1 + 2x_3^1 - 4z_1 - 5z_2 - 3z_3,\ 3 + 5x_1^1 + 2x_2^1 \leq 2x_1^2 + x_2^2 + 0x_3^2 + 5z_1 + 2z_2 + z_3\})$. This convex hull reduces to the unique point: $\{(\mathbf{x}, \mathbf{z}) \in \mathcal{K} : x_1^1 = 0,\ x_2^1 = 0,\ x_3^1 = 1,\ x_1^2 = 1,\ x_2^2 = 1,\ x_3^2 = 0,\ z_1 = 0,\ z_2 = 0,\ z_3 = 0\}$. After projecting out the $\mathbf{z}$-variables, the convex hull of PNEs corresponds exactly to this unique PNE.

Table 2: Best-responses for All Strategy Profiles.

$\mathbf{x}^1$	$\mathbf{x}^2$	BR for player 1	BR for player 2	Cut by or PNE
$(0,0,0) \vee (1,0,0) \vee (0,1,0) \vee (1,1,0) \vee (0,1,1)$	$(0,0,0)$	$(1,1,0) \vee (1,0,1)$	$(1,1,0)$	$\hat{\mathbf{x}}^2 = (1,1,0)$
$(0,0,0) \vee (1,0,0) \vee (0,1,0) \vee (1,1,0) \vee (0,1,1)$	$(1,0,0)$	$(0,1,1)$	$(1,1,0)$	$\hat{\mathbf{x}}^2 = (1,1,0)$
$(0,0,0) \vee (1,0,0) \vee (0,1,0) \vee (1,1,0) \vee (0,1,1)$	$(0,1,0)$	$(1,0,1)$	$(1,1,0)$	$\hat{\mathbf{x}}^2 = (1,1,0)$
$(0,0,0) \vee (1,0,0) \vee (0,1,0) \vee (1,1,0) \vee (0,1,1)$	$(0,0,1)$	$(1,1,0)$	$(1,1,0)$	$\hat{\mathbf{x}}^2 = (1,1,0)$
$(0,0,0) \vee (1,0,0) \vee (0,1,0) \vee (1,1,0) \vee (0,1,1)$	$(1,1,0)$	$(0,0,1)$	$(1,1,0)$	$\hat{\mathbf{x}}^1 = (0,0,1)$
$(0,0,0) \vee (1,0,0) \vee (0,1,0) \vee (1,1,0) \vee (0,1,1)$	$(1,0,1)$	$(0,1,0)$	$(1,1,0)$	$\hat{\mathbf{x}}^2 = (1,1,0)$
$(0,0,0) \vee (1,0,0) \vee (0,1,0) \vee (1,1,0) \vee (0,1,1)$	$(0,1,1)$	$(1,0,0)$	$(1,1,0)$	$\hat{\mathbf{x}}^2 = (1,1,0)$
$(1,0,1) \vee (0,0,1)$	$(0,0,0)$	$(1,1,0) \vee (1,0,1)$	$(1,1,0) \vee (1,0,1)$	$\hat{\mathbf{x}}^2 = (1,1,0)$
$(1,0,1) \vee (0,0,1)$	$(1,0,0)$	$(0,1,1)$	$(1,1,0) \vee (1,0,1)$	$\hat{\mathbf{x}}^2 = (1,1,0)$
$(1,0,1) \vee (0,0,1)$	$(0,1,0)$	$(1,0,1)$	$(1,1,0) \vee (1,0,1)$	$\hat{\mathbf{x}}^2 = (1,1,0)$
$(1,0,1) \vee (0,0,1)$	$(0,0,1)$	$(1,1,0)$	$(1,1,0) \vee (1,0,1)$	$\hat{\mathbf{x}}^2 = (1,1,0)$
$(1,0,1) \vee (0,0,1)$	$(0,1,1)$	$(1,0,0)$	$(1,1,0) \vee (1,0,1)$	$\hat{\mathbf{x}}^2 = (1,1,0)$
$(0,0,1)$	$(1,1,0)$	$(0,0,1)$	$(1,1,0) \vee (1,0,1)$	PNE
$(1,0,1) \vee (0,0,1)$	$(1,0,1)$	$(0,1,0)$	$(1,1,0) \vee (1,0,1)$	$\hat{\mathbf{x}}^1 = (0,1,0)$

Note. Each row represents a set of strategy profiles composed of $\mathbf{x}^1$ and $\mathbf{x}^2$. For example, the last row with $\mathbf{x}^1 = \{(1,0,1) \vee (0,0,1)\}$ and $\mathbf{x}^2 = (1,0,1)$ includes two profiles: $\mathbf{x} = \{(1,0,1), (1,0,1)\}$ and $\mathbf{x} = \{(0,0,1), (1,0,1)\}$. The 'BR for player 1 (or 2)' column lists the best responses of player 1 (or 2) corresponding to the strategy profile $\mathbf{x}^2$ (or $\mathbf{x}^1$) in that row. The 'Cut by or PNE' column shows if a profile is a valid PNE, or if it is eliminated by either a PNE or an equilibrium inequality (with the associated best response).

2.3 CR-BRD and BZR via Algorithmic Building Blocks

We analyze CR-BRD and BZR through the lens of the three algorithmic building blocks introduced in Section 1.2. The CR-BRD algorithm selects any initial strategy profile, i.e., only a point within the original

strategy space $\mathcal{X}_c$, making it functionally equivalent to an inner approximation. It integrates the 'play' and 'improve' phases through the BRD process. A single iteration of solving the best response problems for all players, using a clockwork sequence, serves as the 'play' phase. After each iteration, the algorithm checks whether the resulting strategy profile is a PNE. If not, it continues the BRD process, which serves as the 'improve' phase. When a cycle is detected with a clockwork sequence, the CR-BRD algorithm randomizes strategies, a mechanism that also aligns with the 'improve' phase. The BZR algorithm inherits the features of the three algorithmic building blocks—'approximate', 'play', and 'improve'—from both CR-BRD and ZR, combining their strengths into a unified framework.

In our study, we focus on two classes of IPGs: EBMC games, which exemplify dense interaction matrices D_c and KPGs, which illustrate sparse D_c, where D_c represents inter-player interaction terms as introduced in the player's utility function (1). Although KPGs have been widely studied in the prior literature, EBMC games are introduced for the first time in this study. The following two sections are dedicated to formally defining EBMC games and establishing the existence of PNEs.

3 Edge-weighted Budgeted Maximum Coverage (EBMC) Games

This section presents the edge-weighted budgeted maximum coverage (EBMC) game as a motivating example of large-scale IPGs. In Section 3.1, we introduce the EBMC problem. We then develop two EBMC games with inter-player dynamics in Section 3.2. Finally, we discuss how the AIS prevention resource allocation problem can be modeled as an EBMC game in Section 3.3.

3.1 The Edge-weighted Budgeted Maximum Coverage (EBMC) Problem

The budgeted maximum coverage (BMC) problem (Chekuri and Kumar, 2004) 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 (Chekuri and Kumar, 2004), and Caskurlu et al. (2014) further demonstrate that EBMC remains NP-hard even when restricted to bipartite graphs. In this study, we consider EBMC on both bipartite and non-bipartite graphs and formulate corresponding EBMC games for both cases.

Consider a graph $(I, \mathcal{A})$, where each player $c \in N$ controls a subset of vertices denoted by I_c. The collection I_c, $c \in N$ forms a partition of I, i.e., $\bigsqcup_{c \in N} I_c = I$. Player c seeks to maximize edge coverage by selecting a subset of vertices $I'_c \subseteq I_c$. Edges $(i, j) \in \mathcal{A}$ represent tasks that may be completed if both endpoints are selected. Table 3 summarizes the notation used in the EBMC problem. The parameter k indicates the type of task and determines the structure of the graph, specifically whether it is bipartite. We assume that edges only exist between vertices of different types. When $k = 1$, vertices are classified into two groups: those with type k and those without, forming a bipartite graph. In this case, an edge (i, j) exists only if $f_{ik} = 1$ and $f_{jk} = 0$. When $k \geq 2$, the graph is non-bipartite, as vertices cannot be divided into two disjoint groups. Furthermore, directed edges in both directions—(i, j) and (j, i)—may exist between vertices of different types k_1 and k_2.

In a centralized planning scenario, the objective is to maximize edge coverage throughout the graph $\mathcal{A}$, as described by the objective function (5a). Constraint (5b) ensures that an edge (i, j) is covered if at least one of its adjacent vertices, i or j, is selected. The Social Welfare (SW) model ((5a) –(5c)) incorporates

Table 3: Notation

Set	Description
N	Set of players, $N := \{1, ..., n\}$.
I (resp. I_c)	Set of vertices in the entire graph (resp. subset of vertices controlled by player $c \in N$).
$\mathcal{A}$ (resp. $\mathcal{A}_c$)	Set of arcs in the entire graph (resp. arcs between vertices in I_c).
$\delta_+(I_c)$ (resp. $\delta_-(I_c)$)	Set of arcs with tail (resp. head) in I_c and head (resp. tail) in $I \setminus I_c$.
Param.	**Description**
k	Type index characterizing vertices.
n_{ij}	Edge weight from vertex i to vertex j.
f_{ik}	Indicator: 1 if vertex i has type k, 0 otherwise.
w_{ij}	Aggregated edge weight of arc (i, j) across all types: $w_{ij} = \sum_k f_{ik}(1 - f_{jk})n_{ij}$.
$\mathcal{B}_c$	Budget allocated to player c.
Variable	**Description**
x_i	1 if vertex i is selected, 0 otherwise.
y_{ij}	1 if arc (i, j) is covered (i.e., $x_i = 1$ or $x_j = 1$), 0 otherwise.
Acronym	**Description**
$\mathcal{G}^{\text{Alt}}_{k=1}$ (or $\mathcal{G}^{\text{Alt}}_{k\geq 2}$)	Locally altruistic game with a single type (or multiple types)
$\mathcal{G}^{\text{Self}}_{k=1}$ (or $\mathcal{G}^{\text{Self}}_{k\geq 2}$)	Selfish game with a single type (or multiple types)

individual budget constraints (5c) to ensure adherence to these limits.

The Social Welfare (SW) Model

$$\max_{\substack{x_i \in \{0,1\} \quad \forall i \in I \\ y_{ij} \in \{0,1\} \quad \forall (i,j) \in \mathcal{A}}} \quad \sum_{(i,j)\in\mathcal{A}} w_{ij} y_{ij} \tag{5a}$$

$$\text{s.t.} \quad y_{ij} \leq x_i + x_j \qquad (i, j) \in \mathcal{A} \tag{5b}$$

$$\sum_{i \in I_c} x_i \leq \mathcal{B}_c \qquad c \in N \tag{5c}$$

The optimal solution to the SW model is the OSW solution ($\bar{\mathbf{x}}^*_{\text{sw}}$) as introduced in Section 2.1.

3.2 A Non-cooperative EBMC Game Among N Players

Although the SW model in (5) 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 function to model player interactions: a locally altruistic function and a selfish function, based on the different sets of arcs.

We use I_{-c} to denote the set of vertices not controlled by the player c, and let $\bar{\mathbf{x}}^{-c} \in \{0, 1\}^{I_{-c}}$ be a vector that represents decisions for all players except player c. The optimization model for a single player, described in formulation (6), incorporates $\bar{\mathbf{x}}^{-c}$.

The Single-Player (SP) Model

$$
\begin{aligned}
\max \quad & \tilde{u}_c(\mathbf{y}) && && (6a)\\
\text{s.t.} \quad & y_{ij} \le x_i + x_j && \forall\, (i,j) \in \mathcal{A} && (6b)\\
& \sum_{i \in I_c} x_i \le \mathcal{B}_c && && (6c)\\
& x_i = \bar{x}_i^{-c} && \forall\, i \in I_{-c} && (6d)\\
& x_i \in \{0,1\} && \forall\, i \in I_c && (6e)\\
& y_{ij} \in \{0,1\} && \forall\, (i,j) \in \mathcal{A}. && (6f)
\end{aligned}
$$

The SP model (6), defined by (6a)–(6f), for player c is parameterized by the decisions $\bar{\mathbf{x}}^{-c}$ of other players. When an edge connects player c's vertex $i \in I_c$ to another player's vertex $j \in I_{-c}$, a positive decision $x_j = 1$ implies that edge (i, j) can be covered even if $x_i = 0$. Thus, the actions of other players may positively affect player c's objective. The objective function $\tilde{u}_c(\mathbf{y})$ in (6a) is general and can be tailored to reflect the specific interests of player c. Since each y_{ij} can be written as $y_{ij} = \max(x_i, x_j) = x_i + x_j - x_i x_j$, the objective can equivalently be expressed as a function of $\mathbf{x}$, denoted by $u_c(\mathbf{x})$. This equivalence enables interchangeable use of $\tilde{u}_c(\mathbf{y})$ and $u_c(\mathbf{x})$ under the condition that $y_{ij} = \max(x_i, x_j)$. 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 game-theoretic formulations.

Let $\hat{\mathbf{x}}^c \in \operatorname{argmax}_{\mathbf{x}_c \in \{0,1\}^{I_c}} u_c(\mathbf{x}^c, \bar{\mathbf{x}}^{-c})$ denote the optimal choices of x_i for $i \in I_c$ according to the SP model (6). Note that the optimal solution may not be unique; in such cases, any optimal solution may be selected. Although all optimal solutions yield the same utility value for player c, some may lead to more favorable outcomes at the system level—particularly in terms of total edge coverage across the graph, as defined in (5a). Such global effects are taken into account in the algorithmic designs proposed in this paper. Including all edges in Constraints (6b) and (6f) is computationally inefficient, especially for large graphs. Instead, arc sets can be restricted to relevant edges using a refined definition introduced below.

Definition 1 (Induced Arc Sets)**.** *Let $\mathcal{D} = (I, \mathcal{A})$ be a directed graph and let the subset of vertices $I_c, c \in N$ be a partition of I, i.e., $I := \bigsqcup_{c \in N} I_c$. The induced neighborhood arc set $\mathcal{A}[I_c]$ consists of all arcs that have at least one endpoint in the set I_c. The induced inbound arc set $\mathcal{A}^-[I_c]$ consists of all arcs that have their terminal points in the set I_c. Formally, $\mathcal{A}[I_c] := \{(i,j) \in \mathcal{A}_c \cup \delta_-(I_c) \cup \delta_+(I_c)\}$ and $\mathcal{A}^-[I_c] := \{(i,j) \in \mathcal{A}_c \cup \delta_-(I_c)\}$.*

Regardless of the choice of induced arc sets in Definition 1, the joint constraints of the SP model (6)—interpreted by treating the parametrized variables $\bar{\mathbf{x}}^{-c}$ as decision variables in Constraint (6d)—correspond to those of the SW model (5b)–(5c). Based on these arc set definitions, we introduce two variants of the EBMC game, each defined by the corresponding objective and constraint structures.

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

Locally Altruistic Single-Player (LASP) Model

$$\max \quad \tilde{u}_c^{\text{Alt}}(\mathbf{y}) := \sum_{(i,j)\in\mathcal{A}[I_c]} w_{ij}y_{ij} \tag{7a}$$

$$\text{s.t.} \quad y_{ij} \le x_i + x_j \qquad \forall (i,j) \in \mathcal{A}[I_c] \tag{7b}$$

$$(6c) - (6d) \tag{7c}$$

$$x_i \in \{0,1\} \qquad \forall i \in I_c \tag{7d}$$

$$y_{ij} \in \{0,1\} \qquad \forall (i,j) \in \mathcal{A}[I_c]. \tag{7e}$$

Note that the arc sets $\mathcal{A}$ in Constraints (6b) and (6f) are replaced by $\mathcal{A}[I_c]$. The locally altruistic game, denoted by $\mathcal{G}_k^{\text{Alt}}$, is an EBMC game with N players, where each player solves their individual LASP model (7), defined by (7a)–(7e). We denote the game with a single type as $\mathcal{G}_{k=1}^{\text{Alt}}$ and multiple types as $\mathcal{G}_{k\ge2}^{\text{Alt}}$.

The locally altruistic game still exhibits a degree of selfishness by focusing on arcs controlled by player c and the incoming arcs. However, the term "locally altruistic" arises from the inclusion of outgoing arcs in the utility function, implying that each player not only seeks its own benefit, but also considers the welfare of those players directly impacted by their decisions of player c. Although this game is not the main focus of our study, it plays an important conceptual role: it illustrates that when players account for both self-interest and localized externalities, the OSW solution also satisfies the PNE conditions of the locally altruistic game. These implications will be discussed in detail in Section 4, along with the concept of a potential game.

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

Selfish Single-Player (SSP) Model

$$\max \quad \tilde{u}_c^{\text{Self}}(\mathbf{y}) := \sum_{(i,j)\in\mathcal{A}^-[I_c]} w_{ij}y_{ij} \tag{8a}$$

$$\text{s.t.} \quad y_{ij} \le x_i + x_j \qquad \forall\, (i,j) \in \mathcal{A}^-[I_c] \tag{8b}$$

$$(6c) - (6e) \tag{8c}$$

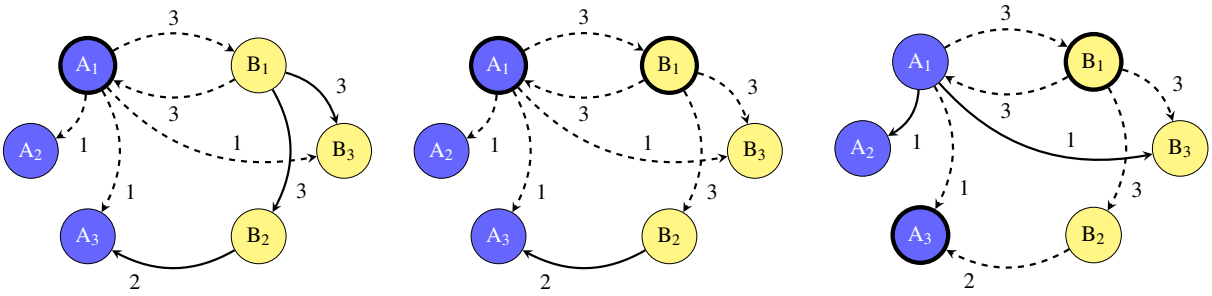
$$x_i \in \{0,1\} \qquad \forall i \in I_c \tag{8d}$$

$$y_{ij} \in \{0,1\} \qquad \forall\, (i,j) \in \mathcal{A}^-[I_c]. \tag{8e}$$

Note that the arc sets $\mathcal{A}$ in Constraints (6b) and (6f) are replaced by $\mathcal{A}^-[I_c]$. The selfish game, denoted by $\mathcal{G}_k^{\text{Self}}$, is an EBMC game with N players, where each player solves their SSP model (8), defined by (8a)–(8e). As in the locally altruistic game, we denote games with a single type as $\mathcal{G}_{k=1}^{\text{Self}}$ and multiple types as $\mathcal{G}_{k\ge2}^{\text{Self}}$.

Figure 2 illustrates the interplay between two selfish players, each controlling three vertices and selecting one under a budget constraint of one. Assume that the vertices differ in status between two task types. Specifically, vertices A_1 and B_2 are of type-1, while vertex B_1 is of type-2. This setup enables interactions across edges (A_1, B_1) and (B_1, A_1), highlighting complexities that extend beyond the bipartite structure of single-type scenarios. 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).



(a) Player A first chooses vertex A_1 **(b)** Player B first chooses vertex B_1 **(c)** Player A now can choose vertex A_3

Figure 2: The interactions in $\mathcal{G}^{\text{Self}}_{k=2}$ for two players A and B. The bold circles and the dashed lines represent the selected vertices and the edges covered by the choices of players, respectively. For example: In Fig. (2a), edges from B_1 to A_1 are covered at vertex A_1, while those from B_1 to B_2 are not covered.

Explicit Reformulation to Standard IPGs The objective functions of the locally altruistic and selfish game can be reformulated into the standard IPG form by substituting **y**-variables with **x**-variables. These are detailed in Appendix 2.

Selfish Non-Game In this scenario, each player c solves the SSP model (8) independently, without engaging in any strategic interaction—that is, each player assumes that no other players will select any vertices. Consequently, the player c determines their strategy by solving $\bar{\mathbf{x}}^c_{\text{ng}} \in \text{argmax}_{\mathbf{x}_c \in \{0,1\}^{I_c}} u_c(\mathbf{x}^c, \mathbf{0}^{-c})$, where $\mathbf{0}^{-c}$ denotes the zero vector. The combination of these individually optimal strategies forms the selfish non-game strategy profile, denoted by $\bar{\mathbf{x}}_{\text{ng}}$.

Equilibrium Inequality for EBMC Games We will show in Section 4.1 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:

$$\sum_{(i,j)\in\mathcal{A}_c} w_{ij} \max(\hat{x}_i, \hat{x}_j) + \sum_{(i,j)\in\delta_-(\mathcal{I}_c)} w_{ij}(x_i + \hat{x}_j - x_i\hat{x}_j) \leq \sum_{(i,j)\in\mathcal{A}_c\cup\delta_-(\mathcal{I}_c)} w_{ij} y_{ij},$$

for all $x_i \in \{0,1\}$, $y_{ij} \in \{0,1\}$ satisfying Constraints (8b)–(8d).

3.3 A Resource Allocation Game among Minnesota Counties

The EBMC games formulated in Section 3.2 are motivated by prevention of AIS in Minnesota (MN). Currently, 84 county planners across Minnesota manage Aquatic Invasive Species (AIS) inspection programs independently, supported by direct state funding (MDR, 2025), which gives them full control over their individual budgets. Faced with limited resources, county planners often implement inspection programs to reduce AIS spread, recognizing recreational boat movements as a primary driver. Although the Minnesota Department of Natural Resources (MN DNR) supports knowledge sharing and limited collaboration through workshops, it does not use a formal mechanism to coordinate the allocation of inspection stations at each lake to maximize the coverage of risky boats moving between two lakes. However, counties report inspection data, such as total inspection hours, to MN DNR, which can be used to estimate inspection stations deployed by each county. Moreover, the growing adoption of AIS Explorer, a web-based planning tool for county planner (MAISRC, 2025), can facilitate the direct sharing of resource allocation decisions in the future. Despite the inherent difficulties in implementing inter-county resource sharing, and the absence of mandated inter-county collaboration by the MN DNR, the positive externalities of one county's inspection program on other counties should be considered for optimal resource utilization. As a result, the strategic interactions among county planners are modeled as a non-cooperative game under complete information.

Although EBMC games are introduced in general terms in Section 3.2, here we demonstrate their specific application to the problem of AIS prevention resource allocation. Each county planner, denoted by player $c \in N$, manages a set of lakes, represented as vertices $i \in I_c$. Each lake i can be infested with AIS of type-k, where k represents one of several aquatic invasive species. A directed edge (i, j) represents a risky boat movement from infested lake i to uninfested lake j, where the edge weight w_{ij} denotes the volume of traffic from lake i to lake j. Each player c has a budget $\mathcal{B}_c$, which represents the maximum number of inspection stations they can deploy, with the assumption that one station covers one lake. The objective of player c is to allocate these stations among their lakes ($i \in I_c$) to maximize the number of inspected risky boat movements while considering the other counties' station allocations. The presence of a single AIS type corresponds to a EBMC game with $k = 1$, while multiple AIS types correspond to a game with $k \geq 2$. The selfish game in this context models county planners whose allocation decisions are based solely on risky boats traveling within their counties and those entering from other counties. In contrast, the locally altruistic game represents county planners who broaden their perspective to also include risky boats leaving their counties. To align with the MN DNR's interest in state-wide AIS prevention effectiveness, it is important to identify a PNE that maximizes the overall social welfare, measured by the total number of inspected risky boat movements. Having defined these game structures, we will establish the theoretical foundation for the existence of PNE for EBMC games in the next section.

4 Pure Nash Equilibrium for EBMC games

As discussed in Section 2.1, the existence of a PNE is not guaranteed for general IPGs. Specifically, when the interaction matrix D^c in (1) is 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 qualifies as a potential game (Monderer and Shapley, 1996), the existence of a PNE can

be ensured. Building upon this, we show that locally altruistic EBMC games have a PNE. Furthermore, we 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 (5a) as the following global function: $\tilde{\phi}(\mathbf{y}) = \sum_{(i,j)\in\mathcal{A}} w_{ij}y_{ij} = \sum_{(i,j)\in\mathcal{A}} w_{ij}\max(x_i, x_j) = \sum_{(i,j)\in\mathcal{A}} w_{ij}(x_i + x_j - x_i x_j) = \phi(\mathbf{x})$. To show that $\phi(\mathbf{x})$ is truly the social welfare function, i.e., the sum of selfish utilities $u_c^{\text{Self}}(\mathbf{x})$ as defined in (8a) (where $\tilde{u}_c^{\text{Self}}(\mathbf{y})$ is rewritten as $u_c^{\text{Self}}(\mathbf{x})$), we present the following lemma and corollary.

Lemma 1. *Let $\mathcal{D} = (I, \mathcal{A})$ be a directed graph and let the subset of vertices $I_c, c \in N$ be a partition of I, i.e., $I := \bigsqcup_{c\in N} I_c$. Then $\mathcal{A} = \bigsqcup_{c\in N} \mathcal{A}^-[\mathcal{I}_c]$.*

Proof. Proof By Definition 1, $\mathcal{A}^-[I_c] := \delta_-(I_c) \cup \mathcal{A}_c$. For any edge $(i, j) \in \mathcal{A}$, there are two possibilities for its endpoints: (i) both endpoints are in the same partition, or (ii) the endpoints are in different partitions. In case (i), $(i, j) \in \mathcal{A}_c$ for any player c. In case (ii), $(i, j) \in \delta_-(I_c)$ for any player c. Considering all players $c \in N$ ensures that every edge (i, j) must be in $\bigcup_{c=1}^{N} \mathcal{A}^-[I_c]$. Furthermore, since each edge (i, j) is unique, it must exclusively belong to one $\mathcal{A}^-[I_c]$. Therefore, $\mathcal{A} = \bigsqcup_{c\in N} \mathcal{A}^-[I_c]$. □

Corollary 1. *The objective of SW model* (5) *is the social welfare function with utilities $u_c^{Self}(\mathbf{x})$, i.e., it is the sum of the selfish objectives given as $\phi(\mathbf{x}) = \sum_{c\in N} u_c^{Self}(\mathbf{x})$.*

Proof. Proof The result follows directly from Lemma 1 by expressing the functions as follows: $\phi(\mathbf{x}) = \sum_{c\in N} u_c^{\text{Self}}(\mathbf{x}) = \sum_{c\in N} \tilde{u}_c^{\text{Self}}(\mathbf{y}) = \sum_{c\in N} \sum_{(i,j)\in\mathcal{A}^-[\mathcal{I}_c]} w_{ij}y_{ij} = \sum_{(i,j)\in\mathcal{A}} w_{ij}y_{ij}$. □

Note that ϕ is not the sum of individual players' objectives in the locally altruistic games.

4.1 Existence of a PNE in the Locally Altruistic Games

First, we show that locally altruistic EBMC games are exact potential games. A game is an *exact potential game* if there exists an exact potential function $\psi : \mathcal{X} \to \mathbb{R}$ such that for all $c \in N$, any strategies $\bar{\mathbf{x}}^{-c} \in \mathcal{X}_{-c}$ and $\bar{\mathbf{x}}^c, \hat{\mathbf{x}}^c \in \mathcal{X}_c$, the change in the potential function equals the change in player c's utility: $\psi(\hat{\mathbf{x}}^c, \bar{\mathbf{x}}^{-c}) - \psi(\bar{\mathbf{x}}^c, \bar{\mathbf{x}}^{-c}) = u_c(\hat{\mathbf{x}}^c, \bar{\mathbf{x}}^{-c}) - u_c(\bar{\mathbf{x}}^c, \bar{\mathbf{x}}^{-c})$. A theorem by Monderer and Shapley (1996) states, "Every potential game possesses at least one PNE, specifically the strategy $\mathbf{x}$ that maximizes $\psi(\mathbf{x})$."

Theorem 1. *The social welfare function ϕ is the exact potential function for the locally altruistic games.*

Proof. Proof For any player c, the edges of the entire graph can be partitioned into two sets: $(i, j) \in \mathcal{A}[\mathcal{I}_c] = \mathcal{A}_c \cup \delta_+(\mathcal{I}_c) \cup \delta_-(\mathcal{I}_c)$ and $(i, j) \in \mathcal{A}_{-c}$. Hence, the potential function can be expressed as:

$$\begin{aligned}\phi(\mathbf{x}) &= \sum_{i,j\in\mathcal{I}_c} w_{ij}\cdot\max(x_i, x_j) + \sum_{i\in\mathcal{I}_c, j\in\mathcal{I}_{-c}} w_{ij}\cdot\max(x_i, \bar{x}_j) + \sum_{i\in\mathcal{I}_{-c}, j\in\mathcal{I}_c} w_{ij}\cdot\max(\bar{x}_i, x_j) + \sum_{i,j\in\mathcal{I}_{-c}} w_{ij}\cdot\max(\bar{x}_i, \bar{x}_j)\\ &= \sum_{(i,j)\in\mathcal{A}[\mathcal{I}_c]} u_c^{\text{Alt}}(\mathbf{x}^c, \bar{\mathbf{x}}^{-c}) + \sum_{i,j\in\mathcal{I}_{-c}} w_{ij}(\bar{\mathbf{x}}^{-c}).\end{aligned}$$

Since player c determines its strategy profile $\mathbf{x}^c$ while $\bar{\mathbf{x}}^{-c}$ remains fixed, any change in $u_c^{\text{Alt}}(\mathbf{x}^c, \bar{\mathbf{x}}^{-c})$ (7a) (where $\tilde{u}_c^{\text{Alt}}(\mathbf{y})$ is rewritten as $u_c^{\text{Alt}}(\mathbf{x})$) corresponds exactly to the change in the potential function $\phi(x)$. □

This result follows from the partitioned structure of the graph, where the locally altruistic utility function captures both outgoing and incoming arcs. While the inclusion of externalities shares conceptual similarities with congestion games — originally introduced in Rosenthal (1973) as a class of games that always admit a PNE — the EBMC game is not a congestion game. In congestion games, a player's payoff depends on the number of players selecting the same choice, whereas in EBMC games, edge coverage is determined by at most two players, and utilities are not based on shared congestion but on bilateral edge coverage.

Recall that $\bar{\mathbf{x}}^*_{\text{sw}}$ is the strategy profile from solving the SW model (5). The following corollary then holds.

Corollary 2. *The strategy profile $\bar{\mathbf{x}}^*_{sw}$ is a PNE for locally altruistic EBMC games.*

Proof. Proof By Theorem 1, the locally altruistic EBMC games are the exact potential games with the potential function $\phi(\mathbf{x})$ and $\bar{\mathbf{x}}^*_{\text{sw}}$ is an optimal solution that maximizes $\phi(\mathbf{x})$ over $\mathbf{x} \in \mathcal{X}$. By Lemma 2.1 from (Monderer and Shapley, 1996), which establishes that the maximum of a potential function in a game constitutes a PNE, $\bar{\mathbf{x}}^*_{\text{sw}}$ is a PNE for locally altruistic EBMC games. □

This corollary allows central planners to recommend an optimal strategy profile $\bar{\mathbf{x}}^*_{\text{sw}}$ 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 (5) and implement the solution.

4.2 Existence of a PNE in the Selfish Games

Next, we establish sufficient conditions for the existence of a PNE in selfish EBMC games, especially when externalities from other players $\bar{\mathbf{x}}^{-c}$ outweigh internal utility contributions from a player's own decisions $\mathbf{x}^c$. Let $I = I^{=k} \cup I^{\neq k}$, where $I^{=k}$ and $I^{\neq k}$ represent the sets of vertices with and without type k, respectively.

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

Proof. Proof Suppose $I^{\text{ng}} \subseteq I^{\neq k}$. For each player $c \in N$, since no edges are covered by the choices of other players $-c$, $u^{\text{self}}_c(\mathbf{x}^c, \bar{\mathbf{x}}^{-c}_{\text{ng}}) = u^{\text{self}}_c(\mathbf{x}^c, \mathbf{0}^{-c})$. Thus, $\bar{\mathbf{x}}^c_{\text{ng}} \in \arg\max_{\mathbf{x}^c \in \mathcal{X}_c} u^{\text{self}}_c(\mathbf{x}^c, \mathbf{0}^{-c}) = \arg\max_{\mathbf{x}^c \in \mathcal{X}_c} u^{\text{self}}_c(\mathbf{x}^c, \bar{\mathbf{x}}^{-c}_{\text{ng}})$. This holds for all $c \in N$, confirming that $\bar{\mathbf{x}}_{\text{ng}}$ is indeed a PNE. On the other hand, suppose $I^{\text{ng}} \subseteq I^{=k}$. Then for each $c \in N$, $u^{\text{self}}_c(\mathbf{x}^c, \bar{\mathbf{x}}^{-c}_{\text{ng}}) \leq u^{\text{self}}_c(\mathbf{x}^c, \mathbf{0}^{-c}) + u^{\text{self}}_c(\mathbf{0}^c, \bar{\mathbf{x}}^{-c}_{\text{ng}}) \leq u^{\text{self}}_c(\bar{\mathbf{x}}^c_{\text{ng}}, \mathbf{0}^{-c}) + u^{\text{self}}_c(\mathbf{0}^c, \bar{\mathbf{x}}^{-c}_{\text{ng}}) = u^{\text{self}}_c(\bar{\mathbf{x}}^c_{\text{ng}}, \bar{\mathbf{x}}^{-c}_{\text{ng}})$. The first inequality is because of potential double counting of edges (i, j), where $i \in I^{\text{ng}}_{-c}$ and $j \in I_c$, covered by both $\mathbf{x}^c$ and $\bar{\mathbf{x}}^{-c}_{\text{ng}}$. The second inequality is achieved by the definition of the non-game solution. The last equality is due to the bipartiteness of set I. To be specific, since $I^{\text{ng}}_c, I^{\text{ng}}_{-c} \in I^{=k}$ and the edges connect exactly one endpoint in $I^{=k}$ to one in $I^{\neq k}$, the summation of the two objectives equals a single objective. □

We now examine the conditions aligned with Theorem 2. Let $w(i), i \in I_c$ denote the sum of the weights of the edges that have an end point at vertex i within $\mathcal{A}^-[\mathcal{I}_c]$. In the random graphs we generated, the condition $I^{\text{ng}} \subseteq I^{\neq k}$ for $\mathcal{G}^{\text{Self}}_{k=1}$ frequently holds. This arises from the structure of the induced inbound arc set, $\mathcal{A}^-[\mathcal{I}_c]$, where $w(j)$ for vertices $j \in I^{\neq k}_c$ frequently exceeds $w(i)$ for vertices $i \in I^{=k}_c$. As the number of players increases, the aggregated weight from incoming arcs generally outnumbers the aggregated weight from internal arcs, leading to a non-game strategy profile $\bar{\mathbf{x}}^c_{\text{ng}}$ that favors the choices of vertices in $I^{\neq k}$.

Even when condition $I^{\text{ng}} \subseteq I^{\neq k}$ is not strictly satisfied, we observe computationally in randomly generated instances that $\mathcal{G}^{\text{Self}}_{k=1}$ almost always admits a PNE. This is because large, though not strictly dominant, externalities can guide the BRD toward a PNE in just a few iterations. Typically, I^{ng} favors the vertices in $I^{\neq k}$ due to the high volume of risky incoming arcs. If BRD starts from a conservative profile like I^{ng}, players iteratively benefit: selecting vertices in $I^{=k}$ triggers other players to select vertices in $I^{\neq k}$ not yet covered. Importantly, selecting vertices in $I^{\neq k}_c$ benefits only player c, whereas selecting vertices in $I^{=k}_c$ benefits others. Given the finite number of vertices in $I^{=k}$, this iterative improvement eventually ends, leading to a PNE. In Section 5.2, we confirm that all instances of the selfish game $\mathcal{G}^{\text{Self}}_{k=1}$ result in a PNE.

However, this dynamic pattern of the players does not guarantee the existence of a PNE in all scenarios. In Appendix 3, we present counterexamples for $\mathcal{G}^{\text{Self}}_{k=1}$ and $\mathcal{G}^{\text{Self}}_{k\geq 2}$.

5 Computational Results

This section evaluates the effectiveness of the CR-BRD algorithm (Algorithm 1), the BZR algorithm (Algorithm 2), and the ZR algorithm proposed by Dragotto and Scatamacchia (2023) in their ability to compute a PNE and identify the best PNE across EBMC games and KPGs. Summarized results for CR-BRD with various initial strategy profiles, along with full results of EBMC games and partial results of KPGs using BZR and ZR, are presented in the main manuscript, while full results are provided in Appendices 4–6.

We begin by describing the computational settings and metrics used throughout the experiments in Section 5.1. Section 5.2 then reports on the results of the EBMC games using randomly generated datasets with varying numbers of players ($n = 2, 3, 5, 8, 10, 15, 20, 25, 30$). In Section 5.3, we present results for the Minnesota dataset, which includes 84 county-level players, providing managerial insights. Finally, Section 5.4 reports results for KPGs, using the instance generation scheme from Dragotto and Scatamacchia (2023) and scaling from small ($n = 2, 3$) to large-scale IPGs ($n = 10, 15, 20, 25, 30$).

5.1 Computational Setup and Metrics

All experiments were conducted on a machine with an Intel(R) Core(TM) i9-13900F CPU @ 3.10GHz (24 cores) and 64GB RAM. All algorithms were implemented in Python and solved using Gurobi Optimizer (v11.0.3) with 16 threads. Equilibrium inequalities were handled using Gurobi's callback and lazy constraint features. The code, instances and results are provided at `https://github.com/HyunwooLee0429/Best-response-dynamics-IPG`. We note that model generation time is non-negligible, especially for large-scale IPGs, where solving best-response problems iteratively for multiple players is required. To optimize performance, we constructed the model once and updated only the relevant changes when solving each player's best-response problem across all algorithms.

Table 4 summarizes the acronyms used throughout the paper and those introduced for the computational study. While OSW and POS are defined in Section 2.1, we report their time-limited approximations—denoted as $\widetilde{\text{OSW}}$ and $\widetilde{\text{POS}}$—due to the large-scale nature of the integer programs. For both BZR and ZR, the time limit was set to 1800 seconds. The SW model (5) was solved with a 1800-second limit to compute the approximate optimal social welfare $\phi(\tilde{\mathbf{x}}^{1800}_{\text{sw}})$. For CR-BRD, two initial strategy profiles were used: $\mathbf{0}^N$ (all-zero vector) and $\tilde{\mathbf{x}}^{300}_{\text{sw}}$ (solution from the SW model solved with a 300-second limit). For standalone CR-BRD, the number of initial solutions (L) and iterations per solution (T) were set to 10 and 20, respectively

(maximum of 200 iterations). When used as a subroutine within BZR, these parameters were reduced to $L = 5$ and $T = 20$ (maximum of 100 iterations).

Table 4: Acronyms, Abbreviations, and Shorthands

Symbol / Abbreviation	Full Term / Meaning
CR-BRD	Clockwork-random best-response dynamics (Algorithm 1)
BZR	Best-response dynamics incorporated zero-regret (Algorithm 2)
ZR	Zero-regret Algorithm (from Dragotto and Scatamacchia (2023))
EBMC	Edge-weighted budgeted maximum coverage (Game)
KPG	Knapsack problem game
SW model	Social welfare maximization model ($\max_{\mathbf{x}\in\mathcal{X}}\{\phi(\mathbf{x}) : \mathbf{x} \in \mathcal{X}\}$)
$\mathbf{0}^N$	Initial strategy profile: all-zero vector
$\tilde{\mathbf{x}}_{\text{sw}}^{\texttt{time}}$	Initial strategy profile obtained by solving the SW model with `time` sec limit
$\bar{\mathbf{x}}_{\text{ng}}$	Non-game strategy profile
$\hat{\mathbf{x}}_{\text{pne}}^{1}$, $\hat{\mathbf{x}}_{\text{pne}}^{*}$	First identified PNE; best identified PNE (within time limit)
$\phi(\cdot)$	Social welfare function
$\widetilde{\text{OSW}}$	Approximate optimal social welfare ($\phi(\tilde{\mathbf{x}}_{\text{sw}}^{\texttt{time}})$)
$\widetilde{\text{POS}}$	Approximated price of stability given by $\phi(\hat{\mathbf{x}}_{\text{pne}}^{*})/\phi(\tilde{\mathbf{x}}_{\text{sw}}^{\texttt{time}})$
BG	Budget ratio
L	Number of initial solutions for CR-BRD
T	Number of iterations per solution in CR-BRD
Iteration	Number of iterations taken by the algorithm
Cycle Detected	Boolean variable for whether a cycle was encountered during CR-BRD
$\#_{\text{cuts(BZR)}}$	Number of equilibrium and PNE inequalities added by BZR
$\#_{\text{pne(BZR)}}$	Number of distinct PNEs found by BZR
$\text{T(BZR}^{\text{1st}}\text{)}$	Time to find the first PNE using BZR
T(BZR)	Total runtime of the BZR algorithm
n	Number of players (or counties) in the game
k	Number of AIS (Aquatic Invasive Species) types considered
$\mathcal{G}_k^{\text{Self}}$	Selfish variant of the EBMC game with k AIS types
$\mathcal{G}_k^{\text{Alt}}$	Altruistic variant of the EBMC game with k AIS types

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

- **CR-BRD Metrics:** For each instance, using both initial strategy profiles ($\mathbf{0}^N$ and $\tilde{\mathbf{x}}_{\text{sw}}^{\texttt{300}}$), we report the identified PNE (in terms of ϕ), runtime (e.g., Time (secs)), number of iterations (e.g., Iteration), whether a cycle was detected (e.g., Cycle Detected), $\widetilde{\text{OSW}}$, and $\widetilde{\text{POS}}$ (computed using the PNE with higher social welfare). Additionally, we include reference values from the BZR results, such as the best PNE found and the best dual bound (e.g., References). Note that the time to obtain $\tilde{\mathbf{x}}_{\text{sw}}^{\texttt{300}}$ is included in the runtime.

- **ZR and BZR Metrics:** For each instance, we report the number of distinct PNEs identified ($\#_{\text{pne(BZR)}}$), the best PNE found (in terms of ϕ), the dual bound (in terms of ϕ), the time to find the first PNE (e.g., $\text{T(BZR}^{\text{1st}}\text{)}$), total runtime (e.g., T(BZR)), $\widetilde{\text{OSW}}$, $\widetilde{\text{POS}}$ (based on the PNE with the highest social welfare), and the number of equilibrium and PNE inequalities added ($\#_{\text{cuts(BZR)}}$).

5.2 EBMC Games with Random Dataset

Table 5 summarizes the data generation scheme for EBMC games based on the random dataset. For multiple AIS cases, infestation probabilities were adjusted to ensure that most lakes remained protected from at least

one AIS type. To enhance realism and avoid a fully connected network, 20% of the edges were randomly removed in the single AIS case and 50% in the multiple AIS cases—the higher deletion rate in the latter helping to manage dataset size. In total, 27 data sets were generated for each case (single and multiple AIS) corresponding to 9 counties × 1 lakes per county × 3 infestation probabilities.

Table 5: Random Dataset Generation Schema with Parameters and Their Description.

Parameter	**Description**
Number of Counties	Chosen once from $\{2, 3, 5, 8, 10, 15, 20, 25, 30\}$.
Number of Lakes per County	Set as 50.
Infestation Probability ($k = 1$)	Randomly chosen from $\{0.2, 0.4, 0.6, 0.8, 1.0\}$.
Infestation Probability ($k \geq 2$)	Randomly chosen for each AIS type. First type: $\{0.2, 0.4, 0.6, 0.8, 1.0\}$, Second type: $\{0.2, 0.4, 0.6, 0.8\}$, Third type: $\{0.2, 0.4, 0.6\}$, Fourth type: $\{0.2, 0.4\}$.
Budget Ratio	Chosen once from $\{0.3, 0.5, 0.8\}$.
County Budgets	Calculated as the number of infested lakes multiplied by the budget ratio.
Edge Weights	Uniformly distributed between 10 and 20.
Graph Connectivity	($k = 1$): 20% of edges are randomly deleted over a completely connected graph. ($k \geq 2$): 50% of edges are removed.

Table 6 summarizes the CR-BRD results across 27 instances using two initial strategy profiles: $\mathbf{0}^N$ and $\tilde{\mathbf{x}}_{\text{sw}}^{300}$. Complete results are provided in Appendix 4. Compared to the single-AIS case $\mathcal{G}_{k=1}^{\text{Self}}$, the multiple-AIS setting $\mathcal{G}_{k=4}^{\text{Self}}$ exhibits a higher $\widetilde{\text{POS}}$, more frequent cycles, and a significantly higher number of iterations. These patterns suggest that the strategic interactions among players become considerably more complex when multiple AIS types are present.

Table 6: Summarized CR-BRD Results with Different Initial Strategy Profiles for $\mathcal{G}_{k=1}^{\text{Self}}$ and $\mathcal{G}_{k=4}^{\text{Self}}$.

type k	Avg† PNE (ϕ)		Avg Price of Stability		Avg References		Avg Time (secs)		Avg Iteration		Cycle Detected	
	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:$\tilde{\mathbf{x}}_{\text{sw}}^{300}$	init:$\tilde{\mathbf{x}}_{\text{sw}}^{300}$	init:$\tilde{\mathbf{x}}_{\text{sw}}^{300}$
$k = 1$	786.88	1006.8	1055.5	1.048	1023.8	1123.2	6.7	208.1	2.0	1.5	0 out of 27	0 out of 27
$k = 4$	3658.0	3658.0	4125.8	1.128	3658.1	4619.5	68.7	349.4	28.6	33.0	5 out of 27	4 out of 27

Note. All objective values (ϕ) are reported in thousands and rounded to the nearest tenth.
†: Averages of ϕ, OSW, and POS are calculated from instances where a PNE is found by both initial strategy profiles.

Tables 7 and 8 present BZR and ZR computational results for $\mathcal{G}_{k=1}^{\text{Self}}$ and $\mathcal{G}_{k=4}^{\text{Self}}$, respectively. Table 7 demonstrates that the BZR algorithm consistently outperforms the ZR algorithm on several capabilities: speed in finding a PNE, finding multiple PNEs, identifying the best PNE, and achieving tighter dual bounds. Specifically, BZR identifies a PNE within a few seconds for instances with $n \leq 10$ and within 90 seconds for $15 \leq n \leq 30$, illustrating strong scalability across problem sizes. In contrast, ZR only returns a PNE in 11 out of 27 instances within the time limit. Moreover, when the instance admits a richer set of equilibria, BZR typically achieves tighter dual bounds and identifies multiple distinct PNEs, as confirming the intuition discussed in Section 2.

Moreover, BZR typically achieves tighter dual bounds and identifies multiple distinct PNEs in many cases, benefiting from the effectiveness of PNE inequalities in instances that admit a richer set of equilibria, as discussed in Section 2. These trends are consistent with the structure of the single-AIS setting $\mathcal{G}_{k=1}^{\text{Self}}$, which often permits a diverse equilibrium landscape. These results underscore the superiority of BZR over ZR in finding a PNE in more complex games.

Table 7: Comparison of BZR and ZR for Selfish EBMC Games $\mathcal{G}^{\text{Self}}_{k=1}$.

		# PNEs Found		Best PNE (ϕ)		Dual Bounds		Price of Stability		Time (secs)				# EI Cuts (PNE Cuts)	
n	BG	$\#_{\text{pne(ZR)}}$	$\#_{\text{pne(BZR)}}$	ZR	BZR	ZR	BZR	$\widetilde{\text{OSW}}$	$\widetilde{\text{POS}}$	T(ZR$^{\text{1st}}$)	T(ZR)	T(BZR$^{\text{1st}}$)	T(BZR)	$\#_{\text{cuts(ZR)}}$	$\#_{\text{cuts(BZR)}}$
2	0.3	1	1	19.2	19.2	19.2	19.2	19.2	1.000	0.2	0.2	0.1	0.2	8	10(2)
	0.5	6	**7**	22.3	22.3	22.3	22.3	22.3	1.000	0.8	1.4	0.2	1.7	11	26(14)
	0.8	1	**5**	23.3	23.3	23.3	23.3	24.9	1.066	1.2	1.2	0.2	1.4	12	20(8)
3	0.3	1	**5**	27.6	27.6	27.6	27.6	27.6	1.000	1.9	1.9	0.4	2.6	9	19(10)
	0.5	**5**	4	42.1	42.1	42.1	42.1	42.1	1.000	7.5	10.0	0.5	9.6	15	25(10)
	0.8	1	1	53.1	53.1	53.1	53.1	53.1	1.000	0.6	0.6	0.4	0.6	12	15(3)
5	0.3	1	**6**	90.4	90.4	90.4	90.4	90.4	1.000	11.4	11.6	1.1	9.5	17	36(18)
	0.5	3	**4**	100.3	**100.6**	101.3	**101.1**	100.6	1.000	1307.4	1800.1	1.5	1800.1	95	39(16)
	0.8	10	**25**	156.7	156.7	157.9	**157.3**	156.7	1.000	983.9	1801.6	1.4	1800.2	182	77(52)
8	0.3	3	**5**	134.1	134.1	134.2	134.2	149.9	1.118	31.9	58.4	4.4	58.0	41	57(16)
	0.5	0	**3**	-	**252.2**	260.1	**259.9**	252.2	1.000	-	1800.4	4.0	1800.3	28	52(24)
	0.8	0	**10**	-	**349.2**	379.6	**353.4**	381.3	1.092	-	1800.2	4.1	1800.2	28	80(40)
10	0.3	2	**6**	317.2	317.2	317.3	**317.2**	317.2	1.000	468.2	1244.1	6.5	794.2	54	78(30)
	0.5	0	**28**	-	**361.7**	372.5	**372.4**	361.7	1.000	-	1802.2	4.1	1801.8	108	175(141)
	0.8	0	**4**	-	**538.1**	558.8	**549.3**	539.8	1.003	-	1800.2	5.4	1800.2	31	57(24)
15	0.3	0	**4**	-	**725.4**	**742.6**	744.7	725.4	1.000	-	1800.0	13.2	1800.0	51	91(40)
	0.5	0	**3**	-	**708.5**	**728.8**	729.9	857.5	1.210	-	1800.1	18.7	1800.0	75	120(38)
	0.8	0	**2**	-	**1216.5**	1328.9	**1247.0**	1219.7	1.003	-	1800.0	14.8	1800.1	49	79(30)
20	0.3	0	**3**	-	**904.5**	937.0	**921.6**	904.5	1.000	-	1800.2	33.6	1800.1	79	130(51)
	0.5	0	**3**	-	**1282.1**	**1468.4**	1469.9	1529.1	1.193	-	1800.3	33.0	1800.1	88	130(42)
	0.8	0	**2**	-	**2024.4**	2435.7	**2181.3**	2415.5	1.193	-	1800.2	32.6	1800.3	78	118(40)
25	0.3	0	**4**	-	**1765.0**	1819.1	**1817.1**	1765.0	1.000	-	1800.3	50.4	1800.1	93	162(69)
	0.5	0	**4**	-	**2597.4**	2768.5	**2767.9**	2597.7	1.000	-	1800.3	51.6	1800.1	100	172(72)
	0.8	0	**2**	-	**3307.4**	3883.9	**3511.5**	3312.8	1.002	-	1800.3	52.6	1800.3	86	135(49)
30	0.3	0	**3**	-	**2448.1**	2568.4	**2552.6**	2448.2	1.000	-	1800.3	76.7	1800.1	98	178(80)
	0.5	0	**3**	-	**3263.9**	3509.6	**3470.5**	3263.8	1.000	-	1800.3	85.1	1800.2	130	202(72)
	0.8	0	**3**	-	**4910.4**	5574.9	**5399.1**	4919.7	1.002	-	1800.4	75.7	1800.4	103	166(63)

Note. All objective values (ϕ) are reported in thousands and rounded to the nearest tenth.
Boldfaced values indicate better solutions found within the time limit, tighter dual bounds, or more PNEs.

Table 8 reinforces the performance gains of the BZR algorithm relative to ZR, particularly in its ability to identify PNEs efficiently and consistently across instances. BZR succeeds in finding a PNE in 24 out of 27 instances, whereas ZR identifies a PNE in only 3 instances within the time limit. While BZR consistently identifies high-quality PNEs, it does not yield tighter dual bounds in most cases. This is due to the fact that the selfish game, $\mathcal{G}^{\text{Self}}_{k=4}$, typically admits a unique or very limited number of PNEs, which leads the BRD process—despite different initializations—to converge to the same equilibrium. As a result, the number of distinct PNE inequalities that can be generated is limited, reducing the potential for additional dual-bound improvements. The contrast reflects structural complexity: single-AIS games, such as $\mathcal{G}^{\text{Self}}_{k=1}$ often admit multiple equilibria, while the added complexity in multiple-AIS games, such as $\mathcal{G}^{\text{Self}}_{k=4}$, have fewer equilibria.

5.3 EBMC Games with Minnesota Dataset

Tables 9 and 10 report the results of the CR-BRD algorithm and the comparison between the BZR and ZR algorithms, respectively, for the selfish EBMC game, $\mathcal{G}^{\text{Self}}_{k=4}$, applied to the Minnesota dataset. This dataset, provided by the Minnesota Aquatic Invasive Species Research Center (MAISRC), contains boat movement data between 9,182 lakes across the state and is based on 2018 infestation records, following the data generation scheme described in Kao et al. (2021). Counties with no lakes or lakes that do not have any incoming or outgoing risky boat movements are excluded from the analysis.

To reflect real-world resource allocation for AIS prevention, we model financial support corresponding to 436 inspection centers across the state. Minnesota allocates approximately $10 million annually (MDR, 2017) to county-level boat inspection programs, employing 800 to 1000 inspectors. This level of staffing

Table 8: Comparison of BZR and ZR for Selfish EBMC Games $\mathcal{G}^{\text{Self}}_{k=4}$.

		# PNEs Found		Best PNE (ϕ)		Dual Bounds		Price of Stability		Time (secs)				# EI Cuts (PNE Cuts)	
n	BG	$\#_{\text{pne(ZR)}}$	$\#_{\text{pne(BZR)}}$	ZR	BZR	ZR	BZR	$\widetilde{\text{OSW}}$	$\widetilde{\text{POS}}$	T(ZR$^{\text{1st}}$)	T(ZR)	T(BZR$^{\text{1st}}$)	T(BZR)	$\#_{\text{cuts(ZR)}}$	$\#_{\text{cuts(BZR)}}$
2	0.3	1	1	51.4	51.4	51.4	51.4	56.3	1.096	8.0	8.1	0.9	8.9	12	13(2)
	0.5	0	**1**	-	**83.8**	**85.5**	86.5	86.6	1.033	-	1800.3	1.0	1800.2	11	13(2)
	0.8	0	**5**	-	**92.8**	**94.0**	94.4	95.2	1.027	-	1800.1	1.8	1800.2	14	19(7)
3	0.3	1	1	115.0	115.0	115.0	115.0	139.7	1.215	1188.2	1188.5	1.5	956.6	18	20(3)
	0.5	0	**1**	-	**178.6**	197.9	**197.4**	200.3	1.122	-	1800.4	1.5	1800.3	15	18(3)
	0.8	0	**1**	-	**252.2**	**257.7**	258.0	253.7	1.006	-	1800.2	1.4	1800.2	15	19(3)
5	0.3	1	1	380.0	380.0	380.0	380.0	399.8	1.052	393.6	534.5	4.1	630.6	28	32(5)
	0.5	0	**1**	-	**557.6**	626.7	**614.6**	577.3	1.035	-	1800.3	4.1	1800.0	25	30(5)
	0.8	0	**1**	-	**650.1**	**671.1**	671.5	661.9	1.018	-	1800.4	4.0	1800.3	25	32(5)
8	0.3	0	**1**	-	**862.1**	**971.5**	971.8	995.8	1.155	-	1800.1	12.0	1800.0	40	48(8)
	0.5	0	**1**	-	**1214.5**	**1522.6**	1527.0	1387.9	1.143	-	1800.1	12.5	1800.0	40	49(8)
	0.8	0	**1**	-	**1661.5**	**1769.7**	1773.7	1721.0	1.036	-	1800.0	11.7	1800.9	40	50(8)
10	0.3	0	**1**	-	**1265.2**	**1507.5**	1508.8	1521.6	1.203	-	1800.1	20.6	1800.1	50	60(10)
	0.5	0	**2**	-	**2084.2**	**2614.2**	2615.5	2253.9	1.081	-	1800.1	19.3	1800.1	50	62(12)
	0.8	0	**1**	-	**2612.3**	**2755.7**	2756.1	2685.2	1.028	-	1800.4	18.5	1800.3	50	60(10)
15	0.3	0	**1**	-	**2916.4**	**3501.5**	3501.6	3440.2	1.180	-	1800.3	45.6	1800.1	75	90(15)
	0.5	0	**1**	-	**4476.8**	**5724.9**	5726.9	5030.1	1.124	-	1800.3	45.9	1800.2	75	90(15)
	0.8	0	0	-	-	**6421.9**	6423.7	6206.3	-	-	1800.3	-	1800.3	75	75(0)
20	0.3	0	**1**	-	**5373.6**	**6476.3**	6476.4	6284.2	1.169	-	1800.4	90.1	1800.2	100	120(20)
	0.5	0	0	-	-	10639.6	10639.6	9019.4	-	-	1800.4	-	1800.4	100	100(0)
	0.8	0	**1**	-	**10149.7**	**10853.9**	10855.5	10510.8	1.036	-	1800.4	81.2	1800.2	99	119(20)
25	0.3	0	**1**	-	**8023.9**	9787.5	9787.5	9502.9	1.184	-	1800.5	144.0	1812.0	125	150(25)
	0.5	0	**1**	-	**12640.7**	17424.1	17424.1	13936.0	1.102	-	1800.5	148.1	1800.5	125	150(25)
	0.8	0	**1**	-	**16255.3**	**17510.3**	17520.3	16806.1	1.034	-	1800.5	179.4	1800.3	125	150(25)
30	0.3	0	**1**	-	**11151.4**	13751.2	13751.2	13520.2	1.212	-	1800.5	229.0	1818.4	150	180(30)
	0.5	0	**1**	-	**17342.4**	25103.4	25103.4	19631.8	1.132	-	1800.6	222.9	1800.6	150	180(30)
	0.8	0	0	-	-	**25019.7**	25028.7	24180.9	-	-	1800.7	-	1912.7	150	120(0)

Note. All objective values (ϕ) are reported in thousands and rounded to the nearest tenth.
Boldfaced values indicate better solutions found within the time limit, tighter dual bounds, or more PNEs.

translates to coverage of approximately 400 to 500 lakes, assuming two inspectors per lake. In our analysis, the number of inspectors assigned to a county is proportional to their specific budgets (MDR, 2017). Four AIS species, Zebra Mussels, Starry Stonewort, Eurasiatic Watermilfoil, and Spiny Water Fleas, are modeled as infestation types in the selfish game $\mathcal{G}^{\text{Self}}_{k=4}$.

Due to the large-scale nature of this 84-player game, we set a time limit of 3600 seconds for the ZR, BZR, and SW models, and a limit of 600 seconds to compute the initial strategy profile $\tilde{\mathbf{x}}^{600}_{\text{SW}}$. As shown in Table 9, CR-BRD takes more than 400 seconds to run from either initial strategy profile.

Table 10 shows that the BZR algorithm successfully identifies a PNE in a reasonable amount of time, while the ZR algorithm does not find any PNE. This result is consistent with previous observations that the ZR algorithm struggles to find PNEs in $\mathcal{G}^{\text{Self}}_{k=4}$, even for much smaller n. The large problem size and relatively high POS suggest that a significant number of cutting planes would be needed, making the ZR algorithm computationally intractable. Although the BZR algorithm does not yield a tighter dual bound, due to the higher complexity of the underlying integer program with multiple AIS types, it proves far more effective in practice by delivering high-quality primal solutions.

Table 9: CR-BRD Results for Selfish EBMC Games $k = 4$ in Minnesota Dataset.

	CR-BRD (ϕ)		Price of Stability		References		Time (secs)		Iteration		Cycle Detected	
n	init:$\mathbf{0}^N$	init:$\tilde{\mathbf{x}}^{600}_{\text{SW}}$	$\widetilde{\text{OSW}}$	$\widetilde{\text{POS}}$	BZR(ϕ)	BZR(Bound)	init:$\mathbf{0}^N$	init:$\tilde{\mathbf{x}}^{600}_{\text{SW}}$	init:$\mathbf{0}^N$	init:$\tilde{\mathbf{x}}^{600}_{\text{SW}}$	init:$\mathbf{0}^N$	init:$\tilde{\mathbf{x}}^{600}_{\text{SW}}$
84	758.9	761.4	887.7	1.166	761.4	1004.7	458.3	587.9	5	4	FALSE	FALSE

Note. Objective values are presented in thousands and rounded to the nearest tenth.

Table 10: Comparison of BZR and ZR for Selfish EBMC Games $k = 4$ in Minnesota Dataset.

	# PNEs Found		Best PNE (ϕ)		Dual Bounds		Price of Stability		Time (secs)				# EI Cuts (PNE Cuts)	
n	$\#_{\text{pne(ZR)}}$	$\#_{\text{pne(BZR)}}$	ZR	BZR	ZR	BZR	$\widetilde{\text{OSW}}$	$\widetilde{\text{POS}}$	$\text{T(ZR}^{\text{1st}}\text{)}$	T(ZR)	$\text{T(BZR}^{\text{1st}}\text{)}$	T(BZR)	$\#_{\text{cuts(ZR)}}$	$\#_{\text{cuts(BZR)}}$
84	0	**2**	-	**761.4**	**861.7**	1004.7	887.7	1.166	-	3601.1	1231.6	3669.6	363	453(90)

Note. Objective values are presented in thousands and rounded to the nearest tenth.

Table 11 presents the county-level utility values for various strategy profiles using the same Minnesota dataset. Here, we only show partial results that consist of 20 counties, and the complete results for all 84 counties are available in Appendix 5. The profiles compared include the non-game strategy profile, $\bar{\mathbf{x}}_{\text{ng}}$, (from Section 3.2), the first identified PNE, $\hat{\mathbf{x}}^1_{\text{pne}}$, and the best PNE, $\hat{\mathbf{x}}^*_{\text{pne}}$, calculated by the BZR algorithm, and the approximated OSW strategy profile $\tilde{\mathbf{x}}^{3600}_{\text{sw}}$. To analyze the results, we compare utility values column-wise, checking whether the following inequalities hold for every county c: $u^{\text{Self}}_c(\bar{\mathbf{x}}_{\text{ng}}) \leq u^{\text{Self}}_c(\hat{\mathbf{x}}^1_{\text{pne}})$, $u^{\text{Self}}_c(\hat{\mathbf{x}}^1_{\text{pne}}) \leq u^{\text{Self}}_c(\hat{\mathbf{x}}^*_{\text{pne}})$, and $u^{\text{Self}}_c(\hat{\mathbf{x}}^*_{\text{pne}}) \leq u^{\text{Self}}_c(\tilde{\mathbf{x}}^{3600}_{\text{sw}})$. Counties that do not meet any of these conditions are highlighted in bold. These patterns indicate that, for nearly all counties: (i) participating in the game yields better outcomes than not participating, (ii) among PNEs, the best one provides higher utility, and (iii) the approximated OSW strategy gives the highest utility. However, this pattern does not hold for 'Sibley' and 'Winona'. In 'Sibley', playing the game does not improve utility compared to the non-game strategy. In 'Winona', the approximated OSW solution does not provide a higher utility than the best (or first) PNE.

The intuition behind these findings is that even counties with selfish objectives benefit from participating in the game due to mutual, unintended positive effects. The identified PNE offers a stable state in which no county has an incentive to deviate unilaterally, making it a useful benchmark. If a socially optimal PNE - defined as the equilibrium with the highest total utility across all counties - does not significantly disadvantage most players, it becomes a strong candidate for implementation. However, this PNE may still diverge notably from the approximated OSW solution, as reflected by a POS of 1.17. Designing locally altruistic games in which the OSW solution also qualifies as a PNE can enable its implementation, although doing so requires consensus among counties. While these findings do not prescribe specific utility functions or endorse any strategy profile as inherently superior, they provide valuable guidance for decision-makers.

Table 11: County-level Utility Values Using Different Strategy Profiles.

County	$u^{\text{Self}}_c(\bar{\mathbf{x}}_{\text{ng}})$	$u^{\text{Self}}_c(\hat{\mathbf{x}}^1_{\text{pne}})$	$u^{\text{Self}}_c(\hat{\mathbf{x}}^*_{\text{pne}})$	$u^{\text{Self}}_c(\tilde{\mathbf{x}}^{3600}_{\text{sw}})$	County	$u^{\text{Self}}_c(\bar{\mathbf{x}}_{\text{ng}})$	$u^{\text{Self}}_c(\hat{\mathbf{x}}^1_{\text{pne}})$	$u^{\text{Self}}_c(\hat{\mathbf{x}}^*_{\text{pne}})$	$u^{\text{Self}}_c(\tilde{\mathbf{x}}^{3600}_{\text{sw}})$
Aitkin	23746	24921	24982	29013	Martin	24	24	24	24
Anoka	11292	12407	12495	15723	Mcleod	7194	7488	7503	8933
Becker	24219	26424	26497	31712	Meeker	8866	9401	9428	9964
Hennepin	21136	22474	22474	27024	Sherburne	5077	5538	5564	6698
Hubbard	22780	24383	24444	29543	**Sibley**	**2390**	**2388**	**2388**	**2429**
Isanti	3635	3876	3884	4227	Stearns	20200	21482	21554	26076
Le sueur	6804	7143	7157	7929	Watonwan	946	955	955	1114
Lincoln	4221	4292	4292	4501	Wilkin	94	102	102	128
Lyon	2752	2934	2939	3314	**Winona**	**1561**	**1579**	**1579**	**1444**
Mahnomen	4177	4404	4428	6012	Wright	22047	23687	23805	29464

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

5.4 Large-scale Knapsack Problem Games Datasets and Results

To further assess the effectiveness of the CR-BRD and BZR algorithms, we present extensive computational results for large-scale knapsack problem games (KPGs)—a widely used benchmark in the IPG literature (Carvalho et al., 2021, 2022; Dragotto and Scatamacchia, 2023; Crönert and Minner, 2024). KPGs are IPGs where n players interact over m items through reciprocally bilinear terms. Each player c solves the following binary knapsack problem: $\max_{x^c} \left\{ \sum_{j=1}^{m} p_j^c x_j^c + \sum_{k=1,k\neq c}^{n} \sum_{j=1}^{m} f_{k,j}^c x_j^c x_j^k : \sum_{j=1}^{m} w_j^c x_j^c \leq b^c, x^c \in \{0,1\}^m \right\}$, where x_j^c is a binary decision variable equal to 1 if player c selects item j, and 0 otherwise. The parameters p_j^c, w_j^c, and b^c denote the unit profit of the item j and the weight of the item, and the total budget of the player c, respectively. The interaction between players is captured by the coefficient of the bilinear term $f_{k,j}^c$, which contributes to the objective of the player c when both players c and k select the item j. To linearize the bilinear terms $x_j^c x_j^k$, we introduce auxiliary variables $z_j^{k,c}$ and impose the following constraints: $z_j^{k,c} \geq x_j^c + x_j^k - 1$, $z_j^{k,c} \leq x_j^c$, $z_j^{k,c} \leq x_j^k$, $\forall j \in \{1,\ldots,m\}$, $\forall c \in N$, $\forall k \in N \setminus \{c\}$.

To solve KPGs, Dragotto and Scatamacchia (2023) have introduced two types of problem-specific equilibrium inequalities: strategic dominance and strategic payoff inequalities. We incorporate these inequalities in our computations along with the standard equilibrium inequalities. However, for larger instances ($n \geq 15$), identifying the dominance relationships and minimal interaction sets required for strategic payoff inequalities is computationally expensive, so these inequalities are omitted.

We adopt the data generation scheme from Dragotto and Scatamacchia (2023) for the KPG interaction types (A) and (B), with a slight modification to the type (C), and substantially scale the size of the problem. Types (A) and (B) draw $f_{k,j}^c$ from a constant value uniformly set in $[1, 100]$ and from independent values uniformly sampled from $[1, 100]$, respectively. Type (C) draws $f_{k,j}^c$ from a uniform distribution over $[-100, 100]$. However, we observe that such large negative interactions make the game overly adversarial, often preventing the identification of a PNE, even for small instances ($n = 2, 3$), as reported in Dragotto and Scatamacchia (2023). This issue becomes even more pronounced in large-scale settings. To mitigate this, we modify type (C) to sample $f_{k,j}^c$ from $[-20, 0]$, reducing adversarial intensity while retaining negative interactions. The player budgets b^c are set as a proportion of the total weights of the elements: $0.2 \sum_{j=1}^{m} w_j^c$, $0.5 \sum_{j=1}^{m} w_j^c$, and $0.8 \sum_{j=1}^{m} w_j^c$. We begin with the largest dataset considered in Dragotto and Scatamacchia (2023), which includes instances with $n = 2, 3$ and $m = 100$, and extend this to significantly larger cases: $n = 2, 3, 5, 8, 10, 15, 20, 25, 30$ with $m = 100$, resulting in 27 generated instances.

Table 12 presents the summarized CR-BRD results with different initial strategy profiles $\mathbf{0}^N$ and $\tilde{\mathbf{x}}_{\text{sw}}^{300}$. A complete set of results are provided in Tables 19, 20, and 21 of Appendix 4). There are several key distinctions across the different interaction types: Type (C) exhibits higher $\widetilde{\text{POS}}$, a greater number of instances with cycles, and requires substantially more iterations to converge to a PNE. These findings indicate that type (C) interactions produce more complex inter-player dynamics than type (A) and (B).

We present a summarized comparison of BZR and ZR across three KPG datasets in Table 13. For each interaction type (A, B, and C), we report the number of instances (out of 27) in which a PNE was found, multiple PNEs were identified, the best PNE was obtained, and tighter dual bounds were achieved. Ties indicate the number of instances for which both algorithms either identified the same best PNE or achieved the same dual bound. We also report the average number of equilibrium inequalities (including PNE inequalities)

Table 12: Summarized CR-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 Iteration		Cycle Detected	
	init:$\mathbf{0}^N$	init:$\bar{\mathbf{x}}_{\text{sw}}^{300}$	$\widetilde{\text{OSW}}$	$\widetilde{\text{POS}}$	BZR(ϕ)	BZR(Bound)	init:$\mathbf{0}^N$	init:$\bar{\mathbf{x}}_{\text{sw}}^{300}$	init:$\mathbf{0}^N$	init:$\bar{\mathbf{x}}_{\text{sw}}^{300}$	init:$\mathbf{0}^N$	init:$\bar{\mathbf{x}}_{\text{sw}}^{300}$
Type A	700.5	710.8	714.2	1.005	715.6	715.8	5.9	189.1	7.1	4.0	0 out of 27	0 out of 27
Type B	705.7	714.7	717.1	1.003	715.6	719.1	9.7	207.3	8.2	4.0	1 out of 27	0 out of 27
Type C	13.2	13.2	15.3	1.161	13.4	14.7	19.6	142.2	64.6	68.8	10 out of 27	11 out of 27

Note. The objective values are presented with 1000s and the value is rounded to the nearest tenth. †: The averages of ϕ, OSW, and POS are computed only from instances where a PNE is identified under both initial strategy profiles: $\mathbf{0}^N$ and $\bar{\mathbf{x}}_{\text{sw}}^{300}$.

added and the total number of PNEs identified. Equilibrium inequalities include both standard and strategic types and are applied only when $n \leq 15$. Following Dragotto and Scatamacchia (2023), strategic payoff inequalities are used only for KPG type (C). The results for type (B) are presented in Table 14, while those for types (A) and (C) are provided in Appendix 6. All metrics are defined in Section 5.1.

Table 13: 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	# EI Cuts (PNE Cuts)	
	$\#_{\text{pne(ZR)}}$	$\#_{\text{pne(BZR)}}$	ZR	BZR	Ties	ZR	BZR	Ties	ZR	BZR	$\widetilde{\text{POS}}$	$\#_{\text{cuts(ZR)}}$	$\#_{\text{cuts(BZR)}}$
Type A	13	751	5	27	5	0	22	9	0	18	1.004	1189	1000(199)
Type B	11	1932	4	27	4	0	23	7	0	20	1.002	892	935(330)
Type C	12	667	7	21	7	0	14	11	5	11	1.147	2180	2353(96)

Table 13 demonstrates that the BZR algorithm consistently outperforms the ZR algorithm in all types of interaction to find a PNE, identify the best PNE, and tighten the dual bounds. The advantage is particularly pronounced for types (A) and (B), where BZR finds a PNE in all 27 instances, while ZR succeeds in only 5 and 4 cases, respectively. Type (C) remains the most challenging due to its adversarial structure, as reflected in higher $\widetilde{\text{POS}}$ values and fewer PNEs found overall, despite the addition of more equilibrium inequalities. In Table 23 (in Appendix 6) and Table 14, the BZR algorithm finds a PNE within a few seconds for small and medium instances and within 60 seconds even for the largest ones. For the dual bound, BZR achieves tighter results in most instances of types (A) and (B), enabled by the presence of multiple high–social-welfare PNEs, which strengthen the impact of PNE inequalities.

For KPG type (C) (shown in Table 24 in Appendix 6), the BZR algorithm again outperforms the ZR algorithm in identifying PNEs, finding multiple PNEs and locating the best PNE. It succeeds in 21 out of 27 cases, compared to 7 out of 27 for the ZR algorithm. BZR also produces tighter dual bounds for $n \leq 15$, while ZR performs better for $n \geq 25$, reflecting the diminishing effectiveness of PNE inequalities as PNEs become scarce. When $n \leq 15$, multiple PNEs are typically found, but for $n \geq 25$, they are rare. As shown in Table 12, type (C) exhibits complex inter-player dynamics driven by adversarial interactions, which ntensify with larger n, creating a more chaotic equilibrium landscape and hindering PNE identification.

5.5 Computational Remarks

The CR-BRD and BZR algorithms offer complementary benefits, allowing users to choose based on their objectives. CR-BRD is well-suited for quickly identifying a single PNE, while BZR is preferable when the goal is to compute multiple PNEs or the socially optimal PNE. This distinction is important in practice,

Table 14: Comparison of BZR and ZR for Type B KPG.

		# PNEs Found		Best PNE (ϕ)		Dual Bounds		Price of Stability		Time (secs)				# EI Cuts (PNE Cuts)	
n	BG	$\#_{\text{pne(ZR)}}$	$\#_{\text{pne(BZR)}}$	ZR	BZR	ZR	BZR	$\widetilde{\text{OSW}}$	$\widetilde{\text{POS}}$	T(ZR$^{\text{1st}}$)	T(ZR)	T(BZR$^{\text{1st}}$)	T(BZR)	$\#_{\text{cuts(ZR)}}$	$\#_{\text{cuts(BZR)}}$
2	0.2	3	**9**	8.2	8.2	8.2	8.2	8.3	1.018	16.4	24.3	0.4	6.5	1944	1905(18)
	0.5	4	**124**	14.2	14.2	14.2	14.2	14.4	1.010	103.1	181.8	0.2	45.0	2281	2025(247)
	0.8	1	**8**	19.1	19.1	19.1	19.1	19.1	1.002	1.1	1.3	0.2	1.7	1224	1233(16)
3	0.2	0	**235**	-	**15.6**	15.7	**15.6**	15.9	1.025	-	1800.1	0.6	1800.4	2516	2084(547)
	0.5	0	**435**	-	**29.0**	29.2	**29.1**	29.4	1.015	-	1800.3	0.5	1800.2	2396	3318(1219)
	0.8	3	**350**	40.0	40.0	40.1	40.1	40.3	1.007	785.7	1800.2	0.5	1800.1	2185	2452(947)
5	0.2	0	**86**	-	**40.5**	40.8	40.8	40.9	1.012	-	1800.4	1.1	1800.1	645	732(304)
	0.5	0	**193**	-	**78.0**	78.8	**78.7**	78.8	1.011	-	1800.7	1.5	1800.2	1417	1862(872)
	0.8	0	**89**	-	**108.3**	108.7	**108.6**	108.7	1.003	-	1801.4	2.1	1800.3	1752	787(423)
8	0.2	0	**66**	-	**91.6**	92.7	92.7	92.6	1.011	-	1800.7	3.7	1800.6	666	962(499)
	0.5	0	**77**	-	**188.6**	190.0	**189.9**	189.9	1.007	-	1800.5	4.0	1800.6	828	1098(573)
	0.8	0	**8**	-	**272.9**	273.4	**273.1**	273.1	1.001	-	1800.7	3.1	1800.3	795	137(64)
10	0.2	0	**48**	-	**135.9**	137.1	**137.0**	136.9	1.008	-	1800.6	3.1	1800.6	523	780(415)
	0.5	0	**31**	-	**287.8**	289.5	**289.4**	289.2	1.005	-	1800.6	5.2	1800.5	306	514(260)
	0.8	0	**35**	-	**426.6**	428.4	**428.2**	427.7	1.003	-	1801.8	5.0	1800.3	667	551(322)
15	0.2	0	**40**	-	**287.6**	290.6	**290.5**	289.2	1.006	-	1800.9	13.4	1801.1	369	895(477)
	0.5	0	**13**	-	**642.1**	645.7	**645.4**	644.0	1.003	-	1801.2	11.4	1800.7	252	337(161)
	0.8	0	**7**	-	**963.8**	967.1	**966.5**	965.3	1.002	-	1800.8	11.4	1800.5	258	217(93)
20	0.2	0	**15**	-	**489.2**	493.2	**492.9**	491.1	1.004	-	1800.4	6.3	1800.6	353	455(229)
	0.5	0	**9**	-	**1123.7**	1129.2	**1129.0**	1125.9	1.002	-	1800.4	6.1	1800.5	325	326(144)
	0.8	0	**11**	-	**1671.6**	1677.5	**1677.3**	1672.9	1.001	-	1800.6	8.3	1800.7	264	346(163)
25	0.2	0	**12**	-	**762.0**	767.6	**767.5**	763.5	1.002	-	1800.7	7.9	1800.6	315	525(256)
	0.5	0	**7**	-	**1719.0**	1728.7	**1728.5**	1723.7	1.003	-	1800.8	9.6	1800.5	338	321(127)
	0.8	0	**6**	-	**2610.9**	2623.1	**2622.8**	2614.3	1.001	-	1800.6	12.6	1800.4	371	314(124)
30	0.2	0	**7**	-	**1085.4**	1094.7	**1094.6**	1089.0	1.003	-	1803.6	13.1	1800.7	422	362(148)
	0.5	0	**6**	-	**2460.2**	2473.3	2473.3	2467.2	1.003	-	1800.6	16.8	1800.6	237	334(124)
	0.8	0	**5**	-	**3749.0**	3764.1	**3763.9**	3750.8	1.000	-	1800.9	11.6	1800.6	425	386(150)

Note. The objective values are presented with 1000s and the value is rounded to the nearest tenth.
Boldfaced values indicate better solutions found within the time limit, tighter dual bounds, or a greater number of PNEs.

as the number and quality of PNEs in an IPG are typically unknown. Estimating the number of PNEs, their social welfare range, and identifying the best PNE can offer valuable guidance for decision-makers. The efficiency of both algorithms stems from CR-BRD's ability to operate on individual strategy sets $\mathcal{X}_c$, avoiding the construction of full joint strategy sets ($\prod_{c\in N}\tilde{\mathcal{X}}_c$ or $\prod_{c\in N}\mathcal{X}_c$), and thereby significantly reducing computational overhead.

These properties enable CR-BRD and BZR to significantly improve the scalability of existing IPG algorithms. In a KPG with $m_c = m$ for all $c \in N$, each player contributes $(n-1)m$ interaction terms, for a total of $n(n-1)m$ across the game. In fully interactive games such as qIPGs, where D_c is element-wise nonzero, the interaction count rises to $n(n-1)m^2$, since each player has $(n-1)m^2$ terms. Our KPG instances reach up to 87,000 interactions—more than 100× the 600 handled by the ZR algorithm (Dragotto and Scatamacchia, 2023). In EBMC games, with dense (but not full) interaction matrices, the largest datasets exceed 1 million interactions, surpassing the largest instances tackled by BM (Sagratella, 2016) and BnP (Schwarze and Stein, 2023) by at least two orders of magnitude.

6 Conclusion

Solving IPGs remains a significant challenge, as current algorithms are often limited to small-scale instances that involve only a few players. The need for efficient and scalable solution methods becomes even more pressing in large-scale settings with many stakeholders, such as the EBMC games introduced in this paper. To address this challenge, we introduce the best-response dynamics for the first time in the IPG literature, and present clockwork-random BRD (CR-BRD) algorithm, which bypasses the need to enumerate the full

joint strategy space of the IPG. We further incorporate CR-BRD as a primal heuristic into the ZR algorithm, resulting in the best-response dynamics incorporated zero-regret (BZR) algorithm. Upon identifying a PNE, we introduce a set of PNE inequalities — binding inequalities valid at that PNE — to tighten the dual bound, especially when the IPG admits multiple PNEs near the socially optimal solution.

Our computational experiments on EBMC games and KPGs show that the BZR algorithm substantially outperforms the state-of-the-art ZR algorithm in identifying PNEs, enumerating multiple equilibria, and finding near-optimal PNEs. This leads to at least an order-of-magnitude improvement in the scalability of solution methods for IPGs.

We also study two variants of the EBMC games that we introduce, characterized by locally altruistic and selfish utility functions, and establish sufficient conditions for the existence of a PNE in each. We validate our framework using real-world data, providing managerial insights for policymakers, particularly in the allocation of limited resources for AIS prevention at both state and county levels.

Acknowledgement: H. Lee and R. Hildebrand were partially funded by AFOSR grant FA9550-21-1-0107. H. Lee and İ. E. Büyüktahtakın have also been funded by the Grado Department of ISE at VT. S. Cai was partially funded by MAISRC Subaward H009064601. 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.

References

Amiet B, Collevecchio A, Scarsini M, Zhong Z (2021) Pure Nash Equilibria and Best-response Dynamics in Random Games. Mathematics of Operations Research 46(4):1552–1572.

Büyüktahtakın IE, Haight RG (2018) 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.

Carvalho M, Dragotto G, Lodi A, Sankaranarayanan S (2021) The Cut-and-Play Algorithm: Computing Nash Equilibria via Outer Approximations. arXiv preprint arXiv:2111.05726 .

Carvalho M, Dragotto G, Lodi A, Sankaranarayanan S (2023) Integer Programming Games: A Gentle Computational Overview. Tutorials in Operations Research: Advancing the Frontiers of OR/MS: From Methodologies to Applications, 31–51 (INFORMS).

Carvalho M, Lodi A, Pedroso JP (2018) Existence of Nash Equilibria on Integer Programming Games. Vaz AIF, Almeida JP, Oliveira JF, Pinto AA, eds., Operational Research: IO2017, Valença, Portugal, June 28-30 XVIII, 11–23 (Springer).

Carvalho M, Lodi A, Pedroso JP (2022) Computing Equilibria for Integer Programming Games. European Journal of Operational Research 303(3):1057–1070.

Carvalho M, Lodi A, Pedroso JP, Viana A (2017) Nash Equilibria in the Two-player Kidney Exchange Game. Mathematical Programming 161:389–417.

Caskurlu B, Mkrtchyan V, Parekh O, Subramani K (2014) On Partial Vertex Cover and Budgeted Maximum Coverage

Problems in Bipartite Graphs. Diaz J, Lanese I, Sangiorgi D, eds., IFIP International Conference on Theoretical Computer Science, 13–26 (Springer).

Chekuri C, Kumar A (2004) Maximum Coverage Problem with Group Budget Constraints and Applications. International Workshop on Randomization and Approximation Techniques in Computer Science, 72–83 (Springer).

Chen C, Cai W, Büyüktahtakın İE, Haight RG (2023) A Gme-theoretic Approach to Incentivize Landowners to Mitigate an Emerald Ash Borer Outbreak. IISE Transactions 1–15.

Corley H (2020) A Regret-based Algorithm for Computing All Pure Nash Equilibria for Noncooperative Games in Normal Form. Theoretical Economics Letters 10(6):1253–1259.

Crönert T, Minner S (2024) Equilibrium Identification and Selection in Finite Games. Operations Research 72(2):816–831.

Dragotto G, Boukhtouta A, Lodi A, Taobane M (2024) The Critical Node Game. Journal of Combinatorial Optimization 47(5):74.

Dragotto G, Scatamacchia R (2023) The Zero Regrets Algorithm: Optimizing over Pure Nash Equilibria via Integer Programming. INFORMS Journal on Computing .

Elgers N, Dang N, De Causmaecker P (2019) A Metaheuristic Approach to Compute Pure Nash Equilibria. Bioinspired Heuristics for Optimization 221–233.

Escobar LE, Mallez S, McCartney M, Lee C, Zielinski DP, Ghosal R, Bajer PG, Wagner C, Nash B, Tomamichel M, et al. (2018) Aquatic Invasive Species in the Great Lakes Region: an Overview. Reviews in Fisheries Science & Aquaculture 26(1):121–138.

Haight RG, Kinsley AC, Kao SY, Yemshanov D, Phelps NB (2021) Optimizing the Location of Watercraft Inspection Stations to Slow the Spread of Aquatic Invasive Species. Biological Invasions 23(12):3907–3919.

Haight RG, Yemshanov D, Kao SY, Phelps NB, Kinsley AC (2023) A bi-level model for state and county aquatic invasive species prevention decisions. Journal of Environmental Management 327:116855.

Heinrich T, Jang Y, Mungo L, Pangallo M, Scott A, Tarbush B, Wiese S (2023) Best-response Dynamics, Playing Sequences, and Convergence to Equilibrium in Random Games. International Journal of Game Theory 52(3):703–735.

Kao SYZ, Enns EA, Tomamichel M, Doll A, Escobar LE, Qiao H, Craft ME, Phelps NB (2021) Network Connectivity of Minnesota Waterbodies and Implications for Aquatic Invasive Species Prevention. Biological Invasions 23:3231–3242.

Kıbış EY, Büyüktahtakın İE, Haight RG, Akhundov N, Knight K, Flower CE (2021) 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.

Köppe M, Ryan CT, Queyranne M (2011) Rational Generating Functions and Integer Programming Games. Operations research 59(6):1445–1460.

MAISRC (2025) AIS Explorer: Data-driven approach to inform decision making. Https://www.aisexplorer.umn.edu/.

Matsui A (1992) Best Response Dynamics and Socially Stable Strategies. Journal of Economic Theory 57(2):343–362.

MDR (2017) Minnesota Department of Revenue 2017. URL `https://www.revenue.state.mn.us/sites/default/files/2017-08/aispa_2018certification.pdf`, Last accessed August 26, 2024.

MDR (2025) Local Aquatic Invasive Species Prevention Aid. `https://www.dnr.state.mn.us/invasives/ais/prevention/index.html`.

Monderer D, Shapley LS (1996) Potential Games. Games and Economic Behavior 14(1):124–143.

Nash Jr JF (1950) Equilibrium Points in N-person Games. Proceedings of the National Academy of Cciences 36(1):48–49.

Rosenthal RW (1973) A Class of Games Possessing Pure-strategy Nash Equilibria. International Journal of Game Theory 2:65–67.

Roughgarden T (2010) Algorithmic Game Theory. Communications of the ACM 53(7):78–86.

Sagratella S (2016) Computing All Solutions of Nash Equilibrium Problems with Discrete Strategy Sets. SIAM Journal on Optimization 26(4):2190–2218.

Sagratella S, Schmidt M, Sudermann-Merx N (2020) The Noncooperative Fixed Charge Transportation Problem. European Journal of Operational Research 284(1):373–382.

Sankaranarayanan S (2024) Best-response Algorithms for Integer Convex Quadratic Simultaneous Games. arXiv preprint arXiv:2405.07119 .

Schwarze S, Stein O (2023) A Branch-and-Prune Algorithm for Discrete Nash Equilibrium Problems. Computational Optimization and Applications 86(2):491–519.

Wiese SC, Heinrich T (2022) The Frequency of Convergent Games Under Best-response Dynamics. Dynamic Games and Applications 1–12.

Wu Z, Dang C, Karimi HR, Zhu C, Gao Q (2014) A Mixed 0-1 Linear Programming Approach to the Computation of All Pure-strategy Nash Equilibria of a Finite n-person Game in Normal Form. Mathematical Problems in Engineering 2014(1):640960.

Appendix 1: Best-response Dynamics in Integer Programming Games

Best-response dynamics (BRD) is an iterative procedure in which players solve their best-response (BR) problems (2) sequentially, based on a predefined playing sequence and an initial strategy profile. To highlight the role of BRD in the IPG setting and the motivation for our study, we present two examples.

Example 2 (BRD Requires Different Initial Strategy Profiles for Multiple PNEs). Consider the following IPG, where two players optimize their respective utility functions subject to individual constraints:

$$\textbf{Player 1:} \quad \max_{x_1^1, x_2^1, x_3^1 \in \{0,1\}} \quad x_1^1 + x_1^1 x_1^2 + 3x_2^1 x_2^2 + 100 x_3^1 x_3^2 \quad \text{s.t.} \quad x_1^1 + x_2^1 + x_3^1 \leq 1.$$

$$\textbf{Player 2:} \quad \max_{x_1^2, x_2^2, x_3^2 \in \{0,1\}} \quad x_1^2 + x_1^2 x_1^1 + 3x_2^2 x_2^1 + 100 x_3^2 x_3^1 \quad \text{s.t.} \quad x_1^2 + x_2^2 + x_3^2 \leq 1.$$

Each player has a strategy set $(0,0,0), (1,0,0), (0,1,0), (0,0,1)$, resulting in 16 possible strategy profiles. For example, one possible profile is $\mathbf{0}^N = \{(0,0,0), (0,0,0)\}$, which we denote as $\{(0,0,0), (0,0,0)\}$. This game has three PNEs: $\hat{\mathbf{x}}^1_{\text{pne}} = \{(1,0,0), (1,0,0)\}, \hat{\mathbf{x}}^2_{\text{pne}} = \{(0,1,0), (0,1,0)\}, \hat{\mathbf{x}}^3_{\text{pne}} = \{(0,0,1), (0,0,1)\}$. Furthermore, defining social welfare as $\phi(\mathbf{x}) = u_1(\mathbf{x}^1, \mathbf{x}^2) + u_2(\mathbf{x}^2, \mathbf{x}^1)$, we obtain $\phi(\hat{\mathbf{x}}^3_{\text{pne}}) = 200 \geq \phi(\hat{\mathbf{x}}^2_{\text{pne}}) = 6 \geq \phi(\hat{\mathbf{x}}^1_{\text{pne}}) = 4$.

The best-response (BR) path refers to the sequence of strategy profiles starting from an initial strategy profile $\bar{\mathbf{x}}_{\text{init}}$ and following a predefined playing sequence. For simplicity, we consider the playing sequence (1,2). Figure 3 illustrates that different PNEs can be reached depending on the chosen initial strategy profile. Figure 3a shows that starting from $\bar{\mathbf{x}}_{\text{init}} = \mathbf{0}^N$, BRD converges to $\hat{\mathbf{x}}^1_{\text{pne}}$, with the following BR path $\{(0,0,0), (0,0,0)\} \xrightarrow{p1} \{(1,0,0), (0,0,0)\} \xrightarrow{p2} \{(1,0,0), (1,0,0)\}$. In fact, any initial strategy profile of the form $\{\mathbf{x}^1, (1,0,0)\}$, where $\mathbf{x}^1 \in \mathcal{X}_1$ will result in the same PNE. If the objective is to find a socially optimal PNE, a different initial strategy profile is needed. Figure 3b demonstrates that the second-best PNE is obtained with the BR path $\{(0,0,0), (0,1,0)\} \xrightarrow{p1} \{(0,1,0), (0,1,0)\} \xrightarrow{p2} \{(0,1,0), (0,1,0)\}$. Any initial strategy profile of the form $\{\mathbf{x}^1, (0,1,0)\}$, where $\mathbf{x}^1 \in \mathcal{X}_1$, will lead to this second-best PNE. Figure 3c shows that the socially optimal PNE is reached with the BR path $\{(0,0,0), (0,0,1)\} \xrightarrow{p1} \{(0,0,1), (0,0,1)\} \xrightarrow{p2} \{(0,0,1), (0,0,1)\}$. Any initial strategy profile of the form $\{\mathbf{x}^1, (0,0,1)\}$, where $\mathbf{x}^1 \in \mathcal{X}_1$ will result in the same PNE.

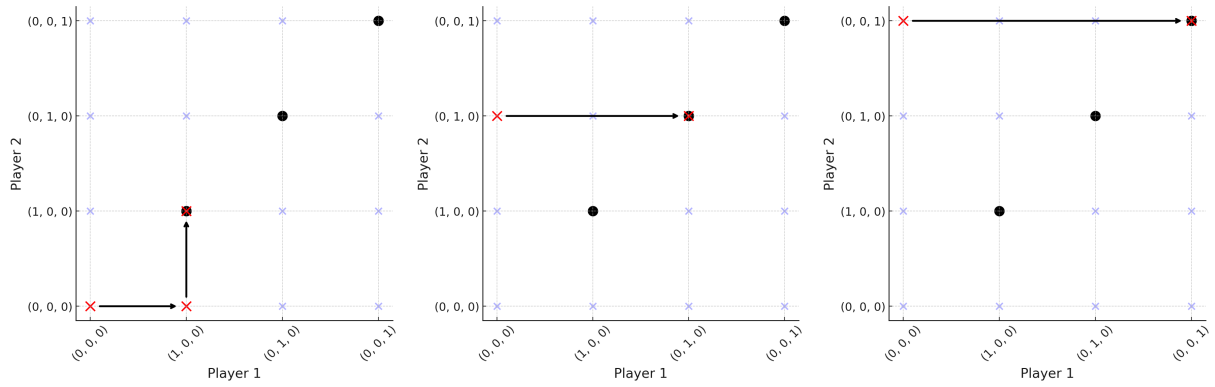
Example 3 (Escaping a Cycle in BRD via Randomized Playing Sequence). Consider the following IPG, where three players optimize their utility functions subject to individual constraints:

$$\textbf{Player 1:} \quad \max_{x_1^1, x_2^1 \in \{0,1\}} \quad x_1^2 + x_2^2 - 6x_1^2 x_1^1 + 6x_1^2 x_1^3 - 2x_2^2 x_2^1 - x_2^2 x_2^3 \quad \text{s.t.} \quad x_1^1 + x_2^1 \leq 1.$$

$$\textbf{Player 2:} \quad \max_{x_1^2, x_2^2 \in \{0,1\}} \quad x_1^1 + x_2^1 + 0x_1^1 x_1^2 + 4x_1^1 x_1^3 + 8x_2^1 x_2^2 - 9x_2^1 x_2^3 \quad \text{s.t.} \quad x_1^2 + x_2^2 \leq 1.$$

$$\textbf{Player 3:} \quad \max_{x_1^3, x_2^3 \in \{0,1\}} \quad x_1^3 + x_2^3 - 6x_1^3 x_1^1 - 5x_1^3 x_1^2 - 7x_2^3 x_1^3 - 6x_2^3 x_2^2 \quad \text{s.t.} \quad x_1^3 + x_2^3 \leq 1.$$

Each player has a strategy set $(0,0), (0,1), (1,0)$, resulting in nine possible strategy profiles. For example, one of such profiles is $\mathbf{0}^N = \{(0,0), (0,0), (0,0)\}$, which means $\mathbf{x}^1 = (0,0), \mathbf{x}^2 = (0,0)$, and $\mathbf{x}^3 = (0,0)$.



(a) BR path 1 **(b)** BR path 2 **(c)** BR path 3

Figure 3: Various Best-response path for two-player game

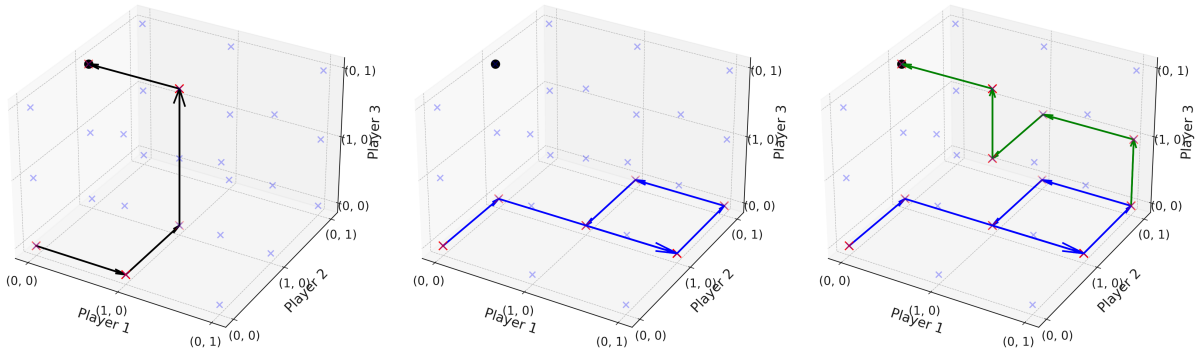
This game has a unique PNE at the strategy profile $\hat{\mathbf{x}}_{\text{pne}} = \{(0,0),(1,0),(0,1)\}$. This example demonstrates that the BR path can vary depending on the playing sequence. In Figure 4, we illustrate several BR paths that highlight these variations. In Figure 4a, the BR path successfully converges to a PNE with a BR path $\{(0,0),(0,0),(0,0)\} \xrightarrow{p1} \{(1,0),(0,0),(0,0)\} \xrightarrow{p2} \{(1,0),(1,0),(0,0)\} \xrightarrow{p3} \{(1,0),(1,0),(0,1)\} \xrightarrow{p1} \{(0,0),(1,0),(0,1)\} \xrightarrow{p2} \{(0,0),(1,0),(0,1)\} \xrightarrow{p3} \{(0,0),(1,0),(0,1)\}$ where 'p1' and 'p2' on the arrows indicate that the player performs the BR update.

However, Figure 4b demonstrates that when using a different playing sequence (2,1,3) with the same initial strategy profile, the BR path falls into a cycle, failing to find a PNE. The path of BR with cycle is: $\{(0,0),(0,0),(0,0)\} \xrightarrow{p2} \{(0,0),(1,0),(0,0)\} \xrightarrow{p1} \{(0,1),(1,0),(0,0)\} \xrightarrow{p3} \{(0,1),(1,0),(0,0)\} \xrightarrow{p2} \{(0,1),(0,1),(0,0)\} \xrightarrow{p1} \{(1,0),(0,1),(0,0)\} \xrightarrow{p3} \{(1,0),(0,1),(0,0)\} \xrightarrow{p2} \{(1,0),(1,0),(0,0)\} \xrightarrow{p1} \{(0,1),(1,0),(0,0)\} \xrightarrow{p3} \{(0,1),(1,0),(0,0)\}$. Here, the last two iterations (six BR problems) repeat indefinitely, preventing convergence.

Figure 4c shows that modifying the play sequence from (2,1,3) to (3,1,2) allows the BRD to escape the cycle. Continuing from the strategy profile $\{(0,1),(1,0),(0,0)\}$ with a new playing sequence (1,2,3), the BR path successfully reaches the PNE: $\{(0,1),(1,0),(0,0)\} \xrightarrow{p1} \{(0,1),(1,0),(0,0)\} \xrightarrow{p2} \{(0,1),(0,1),(0,0)\} \xrightarrow{p3} \{(0,1),(0,1),(1,0)\} \xrightarrow{p1} \{(1,0),(0,1),(1,0)\} \xrightarrow{p2} \{(1,0),(1,0),(1,0)\} \xrightarrow{p3} \{(1,0),(1,0),(1,0)\} \xrightarrow{p1} \{(0,0),(1,0),(1,0)\} \xrightarrow{p2} \{(0,0),(1,0),(1,0)\} \xrightarrow{p3} \{(0,0),(1,0),(0,1)\}$.

In these two examples, we illustrate the application of BRD to IPG settings. A key advantage of BRD is that it follows a BR path without requiring an explicit description of the joint strategy set $\mathcal{X}$; instead, it only involves solving sequential BR problems. Although Figures 3 and 4 depict the BR paths within the joint strategy space (that is, the variable space), the operation of BRD does not require explicit construction of $\mathcal{X}$. This is because, in each BR problem, the strategies of the other players, $\bar{\mathbf{x}}^{-c}$, remain fixed and are treated as parameters. This property significantly enhances computational scalability, making BRD particularly well-suited for large-scale IPGs.

However, some modifications are necessary to improve the performance of BRD. First, introducing randomness and diversity in initial strategy profiles is essential. Second, a randomized playing sequence



(a) BR path 1 **(b)** BR path 2 **(c)** BR path 3

Figure 4: Various Best-response path for three-player game

should be employed upon detecting a cycle to escape local non-convergence. To achieve these improvements, we introduce clockwork-random BRD (CR-BRD) and integrate CR-BRD within a callback version of the ZR algorithm, allowing any integer-feasible solution within the joint strategy space to be explored. Additionally, we implement a randomized version of BRD, where both the initial strategy profiles and playing sequence are randomized upon detecting a cycle.

Appendix 2: Explicit Reformulation to Standard IPGs

The objective functions of the locally altruistic and the selfish game can be reformulated into the standard IPG form by substituting $\mathbf{y}$-variables with $\mathbf{x}$-variables:

$$u_c(\mathbf{x}^c, \mathbf{x}^{-c}) = \frac{1}{2}(\mathbf{x}^c)^\top Q^c \mathbf{x}^c + (D^c \mathbf{x}^{-c})^\top \mathbf{x}^c + \mathbf{d}^c \mathbf{x}^c + \mathbf{d}^{-c} \mathbf{x}^{-c}.$$

This reformulation can be derived by expressing the three core utility components — $\sum_{(i,j)\in\mathcal{A}_c} w_{ij} y_{ij}$, $\sum_{(i,j)\in\delta_+(I_c)} w_{ij} y_{ij}$, and $\sum_{(i,j)\in\delta_-(I_c)} w_{ij} y_{ij}$ — in terms of $\mathbf{x}$-variables. First, for $\sum_{(i,j)\in\mathcal{A}_c} w_{ij}(x_i + x_j - x_{ij})$, the coefficients w_{ij} are added to the i-th and j-th entries of the vector $\mathbf{d}^c$, and $-w_{ij}$ is added to the (i,j)-th entry of the matrix Q^c. Second, for $\sum_{(i,j)\in\delta_+(I_c)} w_{ij}(x_i + x_j - x_{ij})$, w_{ij} is added to the (i,j)-th entry of the matrix D^c, while w_i and w_j are added to the i-th entry of $\mathbf{d}^c$ and the j-th entry of $\mathbf{d}^{-c}$, respectively. Third, for $\sum_{(i,j)\in\delta_-(I_c)} w_{ij}(x_i + x_j - x_{ij})$, w_{ij} is added to the (j,i)-th entry of D^c, while w_i and w_j are added to the i-th entry of $\mathbf{d}^{-c}$ and the j-th entry of $\mathbf{d}^c$, respectively. By applying these transformations across the corresponding arc sets ($\mathcal{A}[I_c]$ for the selfish game and locally altruistic game), EBMC games can be explicitly represented within the standard IPG formulation.

The interaction structure of the EBMC games, captured by the matrix D^c, is similar to that of qIPGs. In qIPG, D^c tends to be dense because each variable in $\mathbf{x}^c$ can potentially interact with all the variables in $\mathbf{x}^{-c}$. In contrast, D^c is sparse in KPGs, where interactions are limited to shared items, meaning that each variable in $\mathbf{x}^c$ interacts with at most $n-1$ variables in $\mathbf{x}^{-c}$.

Notably, this reformulation introduces an additional term, $\mathbf{d}^{-c}\mathbf{x}^{-c}$, which does not appear in the standard IPG objective in (1). However, it still aligns with the core principle of IPGs: a player's utility depends not only

on their own decision variables but also on those of other players. Although this reformulation is theoretically possible, we use the original EBMC formulation with $\mathbf{y}$-variables for computational implementation.

Appendix 3: Counterexamples for the Existence of PNE

This section presents counterexamples for the existence of a PNE in the EBMC games $\mathcal{G}^{\text{Self}}_{k=1}$ and $\mathcal{G}^{\text{Self}}_{k\geq 2}$. For $\mathcal{G}^{\text{Self}}_{k=1}$, 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 15 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.

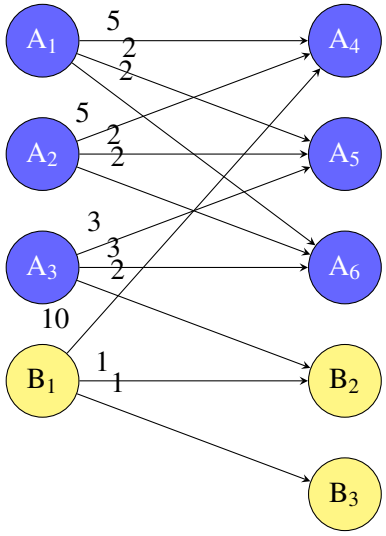


Figure 5: No PNE for $\mathcal{G}^{\text{Self}}_{k=1}$.

Table 15: 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 $\mathcal{G}^{\text{Self}}_{k=1}$, counterexamples in $\mathcal{G}^{\text{Self}}_{k\geq 2}$ are more readily identifiable. An example is presented in Figure 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 16.

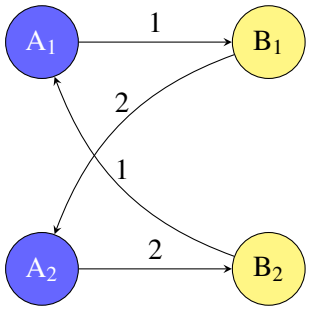


Figure 6: No PNE for $\mathcal{G}^{\text{Self}}_{k=2}$.

Table 16: 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 $\hat{\mathbf{x}}^B = (0,1)$.

Appendix 4: CR-BRD Results

We present the full results of the CR-BRD algorithm using different initial strategy profiles in various problem settings. Specifically, we report results for selfish EBMC games with $k = 1$ and $k = 4$, 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 5.1. All objective values (ϕ) are presented with 1000s and the value is rounded to the nearest tenth.

Table 17: CR-BRD Results with Different Initial Strategy Profiles for EBMC $k = 1$.

		CR-BRD (ϕ)		Price of Stability		References from BZR		Time (secs)		Iteration		Cycle Detected	
n	BG	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:$\tilde{\mathbf{x}}_{\text{sw}}^{300}$	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	FALSE	FALSE
	0.5	21.6	22.3	22.3	1.000	22.3	22.3	0.1	1.9	2	1	FALSE	FALSE
	0.8	22.6	22.6	24.9	1.102	23.3	23.3	0.1	0.3	2	2	FALSE	FALSE
3	0.3	22.0	27.6	27.6	1.000	27.6	27.6	0.1	1.8	2	1	FALSE	FALSE
	0.5	38.8	42.1	42.1	1.000	42.1	42.1	0.2	5.3	2	1	FALSE	FALSE
	0.8	53.1	53.1	53.1	1.000	53.1	53.1	0.1	0.2	2	1	FALSE	FALSE
5	0.3	83.5	90.4	90.4	1.000	90.4	90.4	0.4	11.6	2	1	FALSE	FALSE
	0.5	85.0	100.6	100.6	1.000	100.6	101.3	0.4	301.1	2	1	FALSE	FALSE
	0.8	146.9	156.7	156.7	1.001	156.7	157.9	0.4	301.1	2	1	FALSE	FALSE
8	0.3	130.1	130.1	149.9	1.152	134.1	134.2	1.4	4.1	2	3	FALSE	FALSE
	0.5	169.1	252.2	252.2	1.000	252.2	260.1	1.2	302.0	2	1	FALSE	FALSE
	0.8	269.9	341.4	381.3	1.117	349.2	379.6	1.3	209.3	2	3	FALSE	FALSE
10	0.3	308.1	317.2	317.2	1.000	317.2	317.3	2.0	303.4	2	1	FALSE	FALSE
	0.5	289.8	361.7	361.7	1.000	361.7	372.5	1.5	302.2	2	1	FALSE	FALSE
	0.8	412.2	539.8	539.8	1.000	538.1	558.8	1.8	303.0	2	1	FALSE	FALSE
15	0.3	630.4	725.4	725.4	1.000	725.4	742.6	4.2	306.6	2	1	FALSE	FALSE
	0.5	576.9	576.9	857.5	1.487	708.5	728.8	5.2	17.8	2	3	FALSE	FALSE
	0.8	886.3	1212.2	1219.7	1.006	1216.5	1328.9	4.8	307.7	2	2	FALSE	FALSE
20	0.3	656.1	904.5	904.5	1.000	904.5	937.0	9.9	315.5	2	1	FALSE	FALSE
	0.5	1089.0	1278.5	1529.1	1.196	1282.1	1468.4	10.1	316.4	2	3	FALSE	FALSE
	0.8	1317.6	1701.4	2415.5	1.420	2024.4	2435.7	9.7	316.3	2	3	FALSE	FALSE
25	0.3	1419.5	1765.0	1765.0	1.000	1765.0	1819.1	15.8	323.9	2	1	FALSE	FALSE
	0.5	2140.7	2597.5	2597.7	1.000	2597.4	2768.5	16.4	326.2	2	2	FALSE	FALSE
	0.8	2246.8	3312.8	3312.8	1.000	3307.4	3883.9	17.7	327.5	2	1	FALSE	FALSE
30	0.3	1780.2	2448.2	2448.2	1.000	2448.1	2568.4	23.8	336.7	2	2	FALSE	FALSE
	0.5	2704.9	3263.8	3263.8	1.000	3263.9	3509.6	27.5	340.1	2	1	FALSE	FALSE
	0.8	3724.0	4919.7	4919.7	1.000	4910.4	5574.9	24.6	337.5	2	1	FALSE	FALSE

Note. The objective values are presented with 1000s and the value is rounded to the nearest tenth.

Table 18: CR-BRD Results with Different Initial Strategy Profiles for EBMC $k = 4$.

		CR-BRD (ϕ)		Price of Stability		References		Time (secs)		Iteration		Cycle Detected	
n	BG	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:$\tilde{\mathbf{x}}_{\text{sw}}^{300}$	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	0.3	1.3	3	3	FALSE	FALSE
	0.5	83.8	83.8	86.6	1.033	83.8	86.5	0.6	57.3	3	3	FALSE	FALSE
	0.8	92.1	91.4	95.2	1.042	92.8	94.4	1.1	6.5	3	2	FALSE	FALSE
3	0.3	115.0	115.0	139.7	1.215	115.0	115.0	0.5	11.8	2	3	FALSE	FALSE
	0.5	178.6	178.6	200.3	1.122	178.6	197.4	0.6	301.1	3	4	FALSE	FALSE
	0.8	252.2	252.2	253.7	1.006	252.2	258.0	0.6	301.3	3	4	FALSE	FALSE
5	0.3	380.0	380.0	399.8	1.052	380.0	380.0	1.5	302.6	3	4	FALSE	FALSE
	0.5	557.6	557.6	577.3	1.035	557.6	614.6	1.5	302.4	3	3	FALSE	FALSE
	0.8	650.1	650.1	661.9	1.018	650.1	671.5	1.6	302.8	4	4	FALSE	FALSE
8	0.3	862.1	862.1	995.8	1.155	862.1	971.8	3.9	306.8	3	3	FALSE	FALSE
	0.5	1214.5	1214.5	1387.9	1.143	1214.5	1527.0	3.9	306.6	4	4	FALSE	FALSE
	0.8	1661.5	1661.5	1721.0	1.036	1661.5	1773.7	4.3	307.0	5	4	FALSE	FALSE
10	0.3	1265.2	1265.2	1521.6	1.203	1265.2	1508.8	15.4	309.9	31	3	TRUE	FALSE
	0.5	2083.2	2084.2	2253.9	1.081	2084.2	2615.5	6.7	310.6	4	4	FALSE	FALSE
	0.8	2612.3	2612.3	2685.2	1.028	2612.3	2756.1	6.9	310.6	5	4	FALSE	FALSE
15	0.3	2916.4	2916.4	3440.2	1.180	2916.4	3501.6	16.0	327.0	3	3	FALSE	FALSE
	0.5	4476.8	4476.8	5030.1	1.124	4476.8	5726.9	16.4	326.3	4	5	FALSE	FALSE
	0.8	-	-	6206.3	-	-	6423.7	265.4	576.1	200	200	TRUE	TRUE
20	0.3	5373.6	5373.6	6284.2	1.169	5373.6	6476.4	32.3	350.7	4	5	FALSE	FALSE
	0.5	-	-	9019.4	-	-	10639.6	368.2	670.7	200	200	TRUE	TRUE
	0.8	10149.7	10149.7	10510.8	1.036	10149.7	10855.5	32.5	346.6	6	5	FALSE	FALSE
25	0.3	8023.9	8023.9	9502.9	1.184	8023.9	9787.5	48.0	378.6	2	4	FALSE	FALSE
	0.5	12640.7	12640.7	13936.0	1.102	12640.7	17424.1	54.2	380.2	5	5	FALSE	FALSE
	0.8	16255.3	-	16806.1	1.034	16255.3	17520.3	154.3	781.7	58	200	TRUE	TRUE
30	0.3	11151.4	11151.4	13520.2	1.212	11151.4	13751.2	86.9	424.4	7	7	FALSE	FALSE
	0.5	17342.4	17342.4	19631.8	1.132	17342.4	25103.4	82.7	414.1	5	4	FALSE	FALSE
	0.8	-	-	24180.9	-	-	25028.7	648.0	1017.9	200	200	TRUE	TRUE

Note. The objective values are presented with 1000s and the value is rounded to the nearest tenth.

Table 19: CR-BRD Results with Different Initial Strategy Profiles for Type A KPG.

		CR-BRD (ϕ)		Price of Stability		References		Time (secs)		Iteration		Cycle Detected	
n	BG	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:$\tilde{\mathbf{x}}_{\text{sw}}^{300}$	init:$\mathbf{0}^N$	init:$\tilde{\mathbf{x}}_{\text{sw}}^{300}$
2	0.2	8.1	8.2	8.4	1.022	8.3	8.3	0.4	0.7	3	2	FALSE	FALSE
	0.5	17.7	18.3	18.3	1.002	18.3	18.3	0.1	0.3	3	2	FALSE	FALSE
	0.8	18.4	18.5	18.7	1.011	18.5	18.5	0.1	0.4	3	3	FALSE	FALSE
3	0.2	14.5	14.8	15.2	1.027	14.8	14.9	0.5	0.8	4	3	FALSE	FALSE
	0.5	32.2	32.4	32.8	1.011	32.7	32.7	1.4	5.8	5	3	FALSE	FALSE
	0.8	37.0	37.5	37.8	1.010	37.6	37.6	0.9	3.9	4	3	FALSE	FALSE
5	0.2	27.0	27.2	29.3	1.079	27.3	27.4	1.1	2.6	4	4	FALSE	FALSE
	0.5	59.8	59.8	61.4	1.027	61.0	61.0	1.1	41.2	5	4	FALSE	FALSE
	0.8	128.9	131.3	131.3	1.000	131.3	131.3	1.0	28.0	5	1	FALSE	FALSE
8	0.2	84.7	86.6	87.8	1.013	87.6	87.7	2.5	25.0	6	3	FALSE	FALSE
	0.5	110.4	111.6	116.7	1.046	114.0	114.1	3.2	224.9	5	4	FALSE	FALSE
	0.8	288.3	294.6	295.1	1.002	295.4	295.6	6.8	301.7	6	2	FALSE	FALSE
10	0.2	138.7	140.3	143.9	1.026	143.0	143.0	0.5	174.2	7	4	FALSE	FALSE
	0.5	254.0	260.2	262.8	1.010	263.0	263.1	6.2	303.7	9	5	FALSE	FALSE
	0.8	502.5	504.6	508.7	1.008	506.1	506.5	3.2	302.5	5	3	FALSE	FALSE
15	0.2	248.5	254.7	257.6	1.012	258.1	258.3	4.1	305.0	7	7	FALSE	FALSE
	0.5	590.1	600.4	604.2	1.006	605.2	605.4	9.6	305.1	9	4	FALSE	FALSE
	0.8	1049.6	1057.6	1058.6	1.001	1060.8	1061.3	7.9	303.6	11	5	FALSE	FALSE
20	0.2	593.8	610.5	612.8	1.004	615.1	615.5	6.7	303.3	13	4	FALSE	FALSE
	0.5	1099.9	1121.1	1126.2	1.005	1128.9	1129.0	9.5	306.4	11	6	FALSE	FALSE
	0.8	1670.0	1687.6	1689.9	1.001	1693.9	1694.4	6.2	304.8	6	5	FALSE	FALSE
25	0.2	693.7	705.1	711.5	1.009	714.3	714.5	11.5	305.7	13	4	FALSE	FALSE
	0.5	1781.9	1824.9	1833.3	1.005	1838.6	1838.7	13.5	311.3	9	5	FALSE	FALSE
	0.8	2593.1	2627.1	2628.1	1.000	2637.2	2637.4	9.4	307.0	7	4	FALSE	FALSE
30	0.2	1074.5	1097.2	1104.5	1.007	1109.8	1109.9	12.1	307.0	10	4	FALSE	FALSE
	0.5	2487.2	2527.3	2535.4	1.003	2544.8	2545.1	19.6	317.6	12	8	FALSE	FALSE
	0.8	3309.0	3332.6	3352.1	1.006	3356.3	3356.9	21.1	314.0	11	5	FALSE	FALSE

Note. The objective values are presented with 1000s and the value is rounded to the nearest tenth.

Table 20: CR-BRD Results with Different Initial Strategy Profiles for Type B KPG.

		CR-BRD (ϕ)		Price of Stability		References		Time (secs)		Iteration		Cycle Detected	
n	BG	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:$\tilde{\mathbf{x}}_{\text{sw}}^{300}$	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	0.4	0.2	3	2	FALSE	FALSE
	0.5	14.1	14.2	14.4	1.016	14.2	14.2	0.2	0.4	3	3	FALSE	FALSE
	0.8	18.9	19.0	19.1	1.005	19.1	19.1	0.1	0.4	3	3	FALSE	FALSE
3	0.2	15.1	15.5	15.9	1.030	15.6	15.6	0.6	1.3	4	3	FALSE	FALSE
	0.5	28.5	28.7	29.4	1.025	29.0	29.1	2.2	7.7	6	4	FALSE	FALSE
	0.8	39.5	39.8	40.3	1.012	40.0	40.1	2.0	5.0	5	3	FALSE	FALSE
5	0.2	39.6	39.9	40.9	1.026	40.5	40.8	1.3	5.5	6	5	FALSE	FALSE
	0.5	76.3	77.6	78.8	1.016	78.0	78.7	2.1	22.3	5	5	FALSE	FALSE
	0.8	106.4	107.6	108.7	1.010	108.3	108.6	2.4	32.9	4	5	FALSE	FALSE
8	0.2	87.5	91.1	92.6	1.017	91.6	92.7	4.7	303.5	5	4	FALSE	FALSE
	0.5	183.3	188.1	189.9	1.010	188.6	189.9	6.4	303.7	6	4	FALSE	FALSE
	0.8	267.4	271.7	273.1	1.005	272.9	273.1	6.5	305.8	5	6	FALSE	FALSE
10	0.2	131.9	135.0	136.9	1.014	135.9	137.0	5.4	303.2	7	4	FALSE	FALSE
	0.5	281.7	287.0	289.2	1.007	287.8	289.4	11.8	303.9	12	4	FALSE	FALSE
	0.8	420.4	426.6	427.7	1.003	426.6	428.2	7.8	302.6	7	3	FALSE	FALSE
15	0.2	282.0	286.4	289.2	1.010	287.6	290.5	10.8	309.2	8	6	FALSE	FALSE
	0.5	627.7	641.8	644.0	1.003	642.1	645.4	8.9	304.3	9	3	FALSE	FALSE
	0.8	956.0	963.0	965.3	1.002	963.8	966.5	6.3	302.4	9	2	FALSE	FALSE
20	0.2	478.4	487.7	491.1	1.007	489.2	492.9	4.9	303.6	8	4	FALSE	FALSE
	0.5	1102.7	1121.6	1125.9	1.004	1123.7	1129.0	8.0	303.3	13	3	FALSE	FALSE
	0.8	1663.0	1672.0	1672.9	1.001	1671.6	1677.3	12.6	303.8	14	2	TRUE	FALSE
25	0.2	752.9	760.6	763.5	1.004	762.0	767.5	9.5	306.4	10	4	FALSE	FALSE
	0.5	1689.0	1713.9	1723.7	1.006	1719.0	1728.5	13.7	310.1	11	7	FALSE	FALSE
	0.8	2577.7	2608.5	2614.3	1.002	2610.9	2622.8	16.5	306.1	16	4	FALSE	FALSE
30	0.2	1055.0	1085.9	1089.0	1.003	1085.4	1094.6	45.5	319.7	14	6	FALSE	FALSE
	0.5	2422.2	2460.7	2467.2	1.003	2460.2	2473.3	52.3	319.9	17	6	FALSE	FALSE
	0.8	3728.0	3745.0	3750.8	1.002	3749.0	3763.9	18.9	309.6	12	4	FALSE	FALSE

Note. The objective values are presented with 1000s and the value is rounded to the nearest tenth.

Table 21: CR-BRD Results with Different Initial Strategy Profiles for Type C KPG.

		CR-BRD (ϕ)		Price of Stability		References		Time (secs)		Iteration		Cycle Detected	
n	BG	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:$\tilde{\mathbf{x}}_{\text{sw}}^{300}$	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	0.2	0.2	3	2	FALSE	FALSE
	0.5	7.7	7.8	7.8	1.008	7.8	7.8	0.2	0.4	3	2	FALSE	FALSE
	0.8	8.4	8.4	8.4	1.008	8.4	8.4	0.2	0.1	3	2	FALSE	FALSE
3	0.2	7.1	7.1	7.2	1.012	7.1	7.1	0.4	1.1	4	4	FALSE	FALSE
	0.5	10.3	10.3	10.4	1.011	10.3	10.3	0.4	0.7	4	3	FALSE	FALSE
	0.8	11.5	11.6	11.8	1.024	11.6	11.6	0.2	0.1	3	3	FALSE	FALSE
5	0.2	10.2	10.2	10.4	1.021	10.3	10.3	0.8	1.2	4	3	FALSE	FALSE
	0.5	14.1	14.2	14.6	1.031	14.3	14.4	0.8	1.1	4	3	FALSE	FALSE
	0.8	13.6	13.6	15.0	1.096	13.7	13.7	0.3	0.4	4	3	FALSE	FALSE
8	0.2	15.6	15.6	16.2	1.035	15.8	16.0	0.9	11.5	4	3	FALSE	FALSE
	0.5	15.6	15.7	17.6	1.118	15.9	16.5	1.6	7.1	6	27	FALSE	TRUE
	0.8	15.5	15.5	18.5	1.193	15.7	16.6	1.0	2.4	5	4	FALSE	FALSE
10	0.2	16.5	16.4	17.5	1.064	16.6	17.3	5.2	56.9	6	6	FALSE	FALSE
	0.5	16.5	16.6	20.3	1.218	17.0	18.4	22.3	32.7	85	68	TRUE	TRUE
	0.8	16.1	16.1	20.3	1.262	16.4	18.0	0.9	9.3	6	6	FALSE	FALSE
15	0.2	19.6	19.5	21.8	1.116	20.0	21.5	4.1	212.1	7	7	FALSE	FALSE
	0.5	16.1	16.2	23.0	1.425	16.5	19.7	2.5	113.5	15	28	TRUE	TRUE
	0.8	-	-	23.0	-	-	20.2	25.0	116.7	200	200	TRUE	TRUE
20	0.2	20.7	20.5	24.5	1.184	20.7	24.0	7.1	355.3	17	140	FALSE	FALSE
	0.5	-	-	24.7	-	-	21.4	32.1	337.0	200	200	TRUE	TRUE
	0.8	16.7	16.8	25.5	1.516	17.1	23.0	39.1	309.6	189	30	TRUE	TRUE
25	0.2	20.6	20.4	26.2	1.271	20.9	28.6	28.7	356.1	60	113	FALSE	FALSE
	0.5	-	-	25.7	-	-	24.5	56.7	358.2	200	200	TRUE	TRUE
	0.8	-	-	26.4	-	-	25.5	64.1	365.5	200	200	TRUE	TRUE
30	0.2	-	-	27.8	-	-	28.6	103.8	398.2	200	200	TRUE	TRUE
	0.5	15.9	-	27.3	1.712	16.1	36.1	38.5	386.8	111	200	TRUE	TRUE
	0.8	-	-	27.6	-	-	27.8	91.6	404.3	200	200	TRUE	TRUE

Note. The objective values are presented with 1000s and the value is rounded to the nearest tenth.

Appendix 5: County-level Utility Values

Table 22 presents the complete set of county-level utility values under different strategy profiles, as introduced in Table 11 of the main manuscript.

Table 22: County-level Utility Values Using Different Strategy Profiles.

County	$u_c^{\text{Self}}(\bar{\mathbf{x}}_{\text{ng}})$	$u_c^{\text{Self}}(\hat{\mathbf{x}}^1_{\text{pne}})$	$u_c^{\text{Self}}(\hat{\mathbf{x}}^*_{\text{pne}})$	$u_c^{\text{Self}}(\tilde{\mathbf{x}}^{3600}_{\text{sw}})$	County	$u_c^{\text{Self}}(\bar{\mathbf{x}}_{\text{ng}})$	$u_c^{\text{Self}}(\hat{\mathbf{x}}^1_{\text{pne}})$	$u_c^{\text{Self}}(\hat{\mathbf{x}}^*_{\text{pne}})$	$u_c^{\text{Self}}(\tilde{\mathbf{x}}^{3600}_{\text{sw}})$
Aitkin	23746	24921	24982	29013	Martin	24	24	24	24
Anoka	11292	12407	12495	15723	Mcleod	7194	7488	7503	8933
Becker	24219	26424	26497	31712	Meeker	8866	9401	9428	9964
Beltrami	18720	20001	20061	23756	Mille lacs	1778	1807	1818	1886
Benton	1571	1686	1688	1779	Morrison	5243	5458	5466	6184
Big stone	4555	4816	4827	5664	Mower	43	43	43	48
Blue earth	4712	4926	4932	5439	Murray	3481	3586	3590	3906
Brown	2338	2434	2436	2805	Nicollet	32	32	32	32
Carlton	5393	5801	5813	6910	Nobles	2328	2393	2396	2520
Carver	6664	7304	7331	8211	Norman	8	8	8	8
Cass	40151	42384	42500	48926	Olmsted	587	595	598	852
Chippewa	60	60	60	60	Otter tail	43479	47605	47863	58200
Chisago	5463	5905	5930	6827	Pennington	22	22	22	22
Clay	2075	2474	2485	2893	Pine	6061	6319	6321	7088
Clearwater	8484	9205	9235	10484	Pipestone	38	73	73	78
Cook	28157	29937	30031	34442	Polk	5584	6030	6037	6906
Cottonwood	2713	2818	2824	3212	Pope	10157	10995	11032	13728
Crow wing	38044	39661	39789	46254	Ramsey	6995	7706	7761	9372
Dakota	6958	7347	7406	8664	Redwood	177	177	177	177
Dodge	7	7	7	7	Renville	837	865	871	1056
Douglas	16827	18046	18136	22158	Rice	4496	4844	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	81564	81770	91172
Grant	6635	7360	7391	8817	Scott	4983	5240	5253	6381
Hennepin	21136	22474	22474	27024	Sherburne	5077	5538	5564	6698
Hubbard	22780	24383	24444	29543	**Sibley**	**2390**	**2388**	**2388**	**2429**
Isanti	3635	3876	3884	4227	Stearns	20200	21482	21554	26076
Itasca	51214	53338	53640	61367	Steele	372	372	372	372
Jackson	3061	3170	3171	3474	Stevens	2938	3366	3382	4142
Kanabec	3697	3912	3927	4634	Swift	1287	1456	1461	1778
Kandiyohi	15551	16410	16450	18723	Todd	9108	9815	9842	11661
Kittson	254	254	254	254	Traverse	1427	1466	1466	1523
Koochiching	2217	2284	2284	2314	Wabasha	331	331	331	331
Lac qui parle	1331	1376	1376	1405	Wadena	1834	1946	1950	2289
Lake	30194	31483	31640	36702	Waseca	2940	3072	3080	3089
Lake of t. w.	1009	1009	1009	1009	Washington	11252	11885	11968	15845
Le sueur	6804	7143	7157	7929	Watonwan	946	955	955	1114
Lincoln	4221	4292	4292	4501	Wilkin	94	102	102	128
Lyon	2752	2934	2939	3314	**Winona**	**1561**	**1579**	**1579**	**1444**
Mahnomen	4177	4404	4428	6012	Wright	22047	23687	23805	29464
Marshall	4585	5064	5078	6081	Yellow med.	829	858	858	938

Note. "yellow medicine" is abbreviated as "yellow med." and "lake of the woods" is abbreviated as "lake of t. w.".

Appendix 6: KPG Results

Table 23 and Table 24 present the complete results of KPG types (A) and (C). The metrics and computational settings used in these experiments are described in Section 5.1. All objective values (ϕ) are presented with 1000s and the value is rounded to the nearest tenth.

Table 23: Comparison of BZR and ZR for Type A KPG.

		# PNEs Found		Best PNE (ϕ)		Dual Bounds		Price of Stability		Time (secs)				# EI Cuts (PNE Cuts)	
n	BG	$\#_{\text{pne(ZR)}}$	$\#_{\text{pne(BZR)}}$	ZR	BZR	ZR	BZR	$\widetilde{\text{OSW}}$	$\widetilde{\text{POS}}$	T(ZR$^{\text{1st}}$)	T(ZR)	T(BZR$^{\text{1st}}$)	T(BZR)	$\#_{\text{cuts(ZR)}}$	$\#_{\text{cuts(BZR)}}$
2	0.2	3	**8**	8.3	8.3	8.3	8.3	8.4	1.007	6.7	7.7	0.4	4.0	677	643(16)
	0.5	1	**10**	18.3	18.3	18.3	18.3	18.3	1.002	2.6	2.9	0.2	1.9	155	123(20)
	0.8	1	**3**	18.5	18.5	18.5	18.5	18.7	1.011	0.7	1.1	0.1	1.0	2031	2030(6)
3	0.2	0	**65**	-	**14.8**	14.9	**14.8**	15.2	1.025	-	1800.1	0.6	346.1	2502	1143(161)
	0.5	0	**124**	-	**32.7**	32.7	32.7	32.8	1.003	-	1800.4	0.4	126.0	3640	758(363)
	0.8	5	**6**	37.6	37.6	37.6	37.6	37.8	1.006	14.0	17.4	0.2	2.2	1058	1009(17)
5	0.2	0	**18**	-	**27.3**	27.4	**27.3**	29.3	1.072	-	1800.2	0.8	102.8	3927	3812(49)
	0.5	0	**144**	-	**61.0**	61.0	61.0	61.4	1.016	-	1800.5	0.6	1800.2	2613	2152(626)
	0.8	3	**6**	131.3	131.3	131.3	131.3	131.3	1.000	3.3	30.8	0.7	21.8	58	62(30)
8	0.2	0	**55**	-	**87.6**	87.7	**87.6**	87.8	1.012	-	1800.2	1.6	1800.6	544	682(356)
	0.5	0	**44**	-	**114.0**	114.1	**114.0**	116.7	1.039	-	1800.4	2.1	1800.5	4882	4724(254)
	0.8	0	**11**	-	**295.4**	295.6	**295.4**	295.1	1.000	-	1800.6	2.9	1800.4	1374	161(88)
10	0.2	0	**59**	-	**143.0**	143.0	143.0	143.9	1.020	-	1800.5	0.9	1800.6	1838	1661(485)
	0.5	0	**47**	-	**263.0**	263.1	**263.0**	262.8	1.007	-	1801.1	3.3	1800.5	560	788(433)
	0.8	0	**7**	-	**506.1**	506.5	**506.1**	508.7	1.006	-	1801.3	3.4	1800.2	2638	2133(55)
15	0.2	0	**26**	-	**258.1**	258.3	**258.1**	257.6	1.009	-	1801.1	11.3	1800.6	275	646(327)
	0.5	0	**33**	-	**605.2**	605.4	**605.2**	604.2	1.006	-	1801.4	13.3	1801.0	333	799(437)
	0.8	0	**9**	-	**1060.8**	1061.3	**1060.8**	1058.6	1.000	-	1801.7	11.2	1800.7	371	208(96)
20	0.2	0	**9**	-	**615.1**	615.5	**615.1**	612.8	1.004	-	1800.4	4.4	1800.4	403	351(164)
	0.5	0	**14**	-	**1128.9**	1129.0	1128.9	1126.2	1.004	-	1800.5	6.4	1800.4	252	435(222)
	0.8	0	**9**	-	**1693.9**	1694.4	**1693.9**	1689.9	1.000	-	1800.6	8.5	1800.5	279	276(121)
25	0.2	0	**11**	-	**714.3**	714.5	**714.3**	711.5	1.008	-	1800.5	8.7	1800.5	305	463(217)
	0.5	0	**8**	-	**1838.6**	1838.7	**1838.6**	1833.3	1.004	-	1800.9	8.3	1800.5	242	398(179)
	0.8	0	**8**	-	**2637.2**	2637.4	**2637.2**	2628.1	1.000	-	1800.9	9.6	1800.4	309	372(166)
30	0.2	0	**5**	-	**1109.8**	1109.9	**1109.8**	1104.5	1.008	-	1800.8	9.2	1800.6	354	374(149)
	0.5	0	**7**	-	**2544.8**	2545.1	**2544.8**	2535.4	1.002	-	1802.8	11.4	1800.4	290	429(182)
	0.8	0	**5**	-	**3356.3**	3356.9	**3356.3**	3352.1	1.005	-	1800.5	11.2	1800.5	197	350(147)

Note. The objective values are presented with 1000s and the value is rounded to the nearest tenth.
Boldfaced values indicate better solutions found within the time limit, tighter dual bounds, or a greater number of PNEs.

Table 24: Comparison of BZR and ZR for KPG with interaction type C.

		# PNEs Found		Best PNE (ϕ)		Dual Bounds		Price of Stability		Time (secs)				# EI Cuts (PNE Cuts)	
n	BG	$\#_{\text{pne(ZR)}}$	$\#_{\text{pne(BZR)}}$	ZR	BZR	ZR	BZR	$\widetilde{\text{OSW}}$	$\widetilde{\text{POS}}$	T(ZR$^{\text{1st}}$)	T(ZR)	T(BZR$^{\text{1st}}$)	T(BZR)	$\#_{\text{cuts(ZR)}}$	$\#_{\text{cuts(BZR)}}$
2	0.2	1	1	4.7	4.7	4.7	4.7	4.7	1.007	0.8	1.0	0.2	1.2	4075	4077(2)
	0.5	3	**13**	7.8	7.8	7.8	7.8	7.8	1.007	3.7	5.3	0.1	3.9	4159	4184(26)
	0.8	1	**2**	8.4	8.4	8.4	8.4	8.4	1.008	0.8	1.0	0.1	0.8	3567	3570(4)
3	0.2	1	**2**	7.1	7.1	7.1	7.1	7.2	1.012	6.4	6.8	0.4	3.2	4385	4387(5)
	0.5	1	**6**	10.3	10.3	10.3	10.3	10.4	1.011	36.2	59.9	0.4	14.7	4443	4439(14)
	0.8	3	**4**	11.6	11.6	11.6	11.6	11.8	1.024	1.3	1.7	0.2	1.6	5627	5625(8)
5	0.2	0	**10**	-	**10.3**	10.3	10.3	10.4	1.012	-	1800.4	0.6	1128.1	4196	4227(30)
	0.5	0	**24**	-	**14.3**	14.5	**14.4**	14.6	1.021	-	1800.2	0.5	1800.4	5012	5081(55)
	0.8	2	**27**	13.7	13.7	13.7	13.7	15.0	1.092	194.5	376.4	0.4	70.4	4119	4084(35)
8	0.2	0	**19**	-	**15.8**	16.1	**16.0**	16.2	1.024	-	1800.7	1.3	1800.2	2713	2801(47)
	0.5	0	**38**	-	**15.9**	16.7	**16.5**	17.6	1.107	-	1800.4	1.3	1800.4	2275	2524(129)
	0.8	0	**165**	-	**15.7**	16.8	**16.6**	18.5	1.183	-	1800.4	0.8	1800.5	2970	3463(273)
10	0.2	0	**26**	-	**16.6**	17.4	**17.3**	17.5	1.054	-	1800.6	3.2	1800.2	1559	1711(98)
	0.5	0	**70**	-	**17.0**	18.9	**18.4**	20.3	1.193	-	1800.6	19.1	1800.5	1564	2215(323)
	0.8	0	**162**	-	**16.4**	18.6	**18.0**	20.3	1.242	-	1800.4	1.2	1802.3	1775	3390(702)
15	0.2	0	**20**	-	**20.0**	21.7	**21.5**	21.8	1.094	-	1800.6	21.7	1800.3	1608	1751(126)
	0.5	0	**36**	-	**16.5**	20.1	**19.7**	23.0	1.400	-	1800.4	20.1	1800.5	1651	2067(217)
	0.8	0	0	-	-	20.2	20.2	23.0	-	-	1800.4		1800.3	1633	1633
20	0.2	0	**10**	-	**20.7**	24.1	**24.0**	24.5	1.183	-	1800.3	13.6	1800.3	140	288(144)
	0.5	0	0	-	-	21.7	**21.4**	24.7	-	-	1800.5		1800.2	160	180
	0.8	0	**16**	-	**17.1**	**22.6**	23.0	25.5	1.492	-	1800.3	8.4	1800.3	200	365(150)
25	0.2	0	**7**	-	**20.9**	**26.1**	28.6	26.2	1.253	-	1800.5	7.5	1800.5	150	281(91)
	0.5	0	0	-	-	24.5	24.5	25.7	-	-	1800.4		1800.3	175	175
	0.8	0	0	-	-	25.5	25.5	26.4	-	-	1800.5		1800.3	175	175
30	0.2	0	0	-	-	**28.4**	28.6	27.8	-	-	1800.4		1800.4	150	150
	0.5	0	**9**	-	**16.1**	**27.3**	36.1	27.3	1.690	-	1800.5	47.9	1800.3	180	481(114)
	0.8	0	0	-	-	**27.7**	27.8	27.6	-	-	1800.3		1800.3	210	210

Note. The objective values are presented with 1000s and the value is rounded to the nearest tenth.
Boldfaced values indicate better solutions found within the time limit, tighter dual bounds, or a greater number of PNEs.