

Solving Factorable Programs with Applications to Cluster Analysis, Risk Management, and Control Systems Design

Jitamitra Desai

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

**Doctor of Philosophy
in
Industrial and Systems Engineering**

Dr. Hanif D. Sherali, Chair

Dr. Subhash C. Sarin

Dr. Terry L. Herdman

Dr. Patrick C. Koelling

Dr. Barbara M. P. Fraticelli

June 28th, 2005

Blacksburg, Virginia

Keywords: Global optimization, nonconvex programming problems, Reformulation-Linearization Technique (RLT), factorable programs, hard and fuzzy clustering, risk management, event tree optimization, control systems design.

Solving Factorable Programs with Applications to Cluster Analysis, Risk Management, and Control Systems Design

Jitamitra Desai

(ABSTRACT)

Despite recent advances in optimization research and computing technology, deriving global optimal solutions to nonconvex optimization problems remains a daunting task. Existing approaches for solving such formidable problems are typically heuristic in nature, often leading to significantly sub-optimal solutions. This motivates the need to develop a framework for optimally solving a broad class of nonconvex programming problems, which yet retains sufficient flexibility to exploit inherent special structures. Toward this end, we focus in this dissertation on a variety of applications that occur in practice as instances of polynomial programming problems or more general nonconvex factorable programs, and we employ a central theme based on the *Reformulation-Linearization Technique* (RLT) to design theoretically convergent and practically effective and robust solution methodologies.

We begin our discussion in this dissertation by providing a basis for developing efficient solution methodologies for solving the class of nonconvex factorable programming problems. Recognizing the ability of the RLT to solve polynomial programs to (global) optimality, the basic idea is to solve the given nonconvex program via a series of polynomial programming approximations. The construction and manipulation of these polynomial approximations is conducted in a manner that achieves convergence of the overall algorithmic scheme to global optimality. Extensions to handle more complex black-box functions are also discussed.

Following this introduction, we turn our attention to the main contribution of this dissertation in designing specialized variants of this general algorithmic framework for five particular important applications that arise in practice, and for which no effective model and global optimization procedure has heretofore been developed. In each case, we demonstrate that an existing incumbent commercial global optimization software package (BARON) often fails to detect a true optimum, frequently by a significant margin. On the

other hand, the generation of tight linear programming representations via the proposed methodology leads to an effective and robust methodology for solving these computationally intractable problems.

We first present a global optimization algorithm to solve the *hard clustering problem*, where each data point is to be assigned to exactly one cluster. The field of cluster analysis is primarily concerned with the sorting of data points into different clusters so as to optimize a certain criterion. The hard clustering problem is accordingly formulated as a nonlinear program, for which a tight linear programming relaxation is constructed via the RLT in concert with additional valid inequalities that serve to defeat the inherent symmetry in the problem. This construct is embedded within a specialized branch-and-bound algorithm to solve the problem to global optimality. Pertinent implementation issues that can enhance the efficiency of the branch-and-bound algorithm are also discussed. In particular, two different types of symmetry breaking rules are implemented, and it is observed that the symmetry defeating strategy based on a lexicographic differentiation of data points is somewhat more effective when compared to an alternative cluster reordering rule. More importantly, we show that incorporating such symmetry-defeating valid inequalities in the problem leads to a reduction of 19.36% in computational effort. Another experiment related to testing two types of branching strategies is also performed. This investigation reveals that a proposed hierarchical branching strategy is more effective for larger-sized problems, in comparison with a traditional enumeration scheme that works well only for relatively smaller data sets. Results based on computational experiments performed using standard as well as synthetically generated data sets establish the relative efficacy and robustness of the proposed approach in contrast with the popular k -means algorithm, as well as in comparison with the commercial global optimization software BARON. Specifically, in the results obtained, the RLT-based branch-and-bound algorithm dominated BARON in terms of both CPU time and quality of the resulting solution (objective function value) by 34.3% and 26.5%, respectively. With regard to the k -means heuristic, even a simple rounding scheme applied to the node-zero solution for the proposed approach itself outperformed the k -means solution by 17.2% and 13.3% in terms of CPU time and objective function value, respectively. Based on the algorithmic performance at node zero, we also design a heuristic procedure to obtain a good quality solution at a relative ease of computational effort for large-scale problems. Note that in practice, cluster analysis problems can involve very large data sets, and the results in this work suggest

that designing heuristic methods based on constructs that are borrowed from strong effective exact procedures is a prudent approach for addressing such problems.

Continuing in this same vein, we next present a global optimization algorithm to solve the *fuzzy clustering problem*, where each data point is to be assigned to (possibly) several clusters, with a membership grade assigned to each data point that reflects the likelihood of the data point belonging to that cluster. We show that the objective function for the fuzzy clustering problem, based on a quadratic degree of fuzziness, can be reduced to a cubic nonconvex polynomial program. Similar to the hard clustering case, a tight linear programming relaxation is derived via the RLT and this construct is once again embedded within a specialized branch-and-bound algorithm to solve the problem to global optimality. Computational experience is reported for various problem instances, and the results validate the robustness of the proposed algorithmic procedure and exhibit its dominance over the popularly used fuzzy c-means clustering algorithm (FCMA), as well as the commercial software BARON. On an average, for data sets involving three and five cluster centers, the proposed approach required only 14.05% and 9.85%, respectively, of the CPU time taken by the FCMA, and yet yielded solutions that were respectively superior by 69.32% and 77.88% in terms of objective function value. In contrast, using the commercial software BARON to directly solve the nonconvex program for these two sets of test problems resulted in suboptimal solutions, respectively, deteriorating the objective function value by 28.53% and 53.99%, while consuming an additional 50.80% and 45.43% of CPU time as compared to the proposed approach.

Second, we describe an application from the field of *risk management*, wherein an emergency response manager faces the situation of having to decide the allocation of emergency response resources so as to mitigate risk, given that a particular catastrophic event has occurred. Specifically, we consider the problem of allocating certain available emergency response resources to mitigate risks that arise in the aftermath of a natural disaster, terrorist attack, or other unforeseen calamities. Accordingly, we develop a decision support system for this macro-level risk management problem under equity considerations. The resulting model formulation is a difficult nonconvex factorable program, for which a tight linear programming relaxation is derived by reducing the nonconvex terms in the problem to linearized functions via a suitable polyhedral outer-approximation construction process. Subsequently, this relaxation is embedded within a specialized branch-and-bound procedure and the overall proposed methodology is proven to converge to a global optimum. Various alternative partitioning strategies that could potentially be employed in the context of this branch-and-bound framework, while

preserving the theoretical convergence property, are also explored. Computational experience is provided for a hypothetical case scenario based on different parameter inputs and alternative branching strategies. The results exhibit that while consuming comparable computational effort, our algorithm performs significantly better by provably yielding optimal solutions for which the resulting objective function value improves over that produced by the commercial software BARON as well as an ad-hoc intuitive method by average margins of 14.6% and 17.4%, respectively. Moreover, sensitivity analyses conducted with respect to the equity parameters reveal that the proposed approach also yields relatively more equitable allocations when compared with these alternative methods. This work has the potential of affording great economic and social benefits to both the government and private sectors.

Next, we consider a strategic planning decision problem of allocating certain available preventive and mitigation resources to respectively reduce the failure probabilities of system safety features and the final possible ensuing consequences or losses that might arise in the aftermath of a hazardous event. A novel model formulation, based on an *event tree optimization* approach is devised to cast this micro-level cascading risk scenario problem as a nonconvex factorable program. Using an efficient polyhedral outer-approximation technique, we derive a tight linear programming relaxation and offer several theoretical insights that serve to lay the foundation for designing a specialized branch-and-bound procedure that is proven to converge to global optimality. Two alternative partitioning strategies that preserve the theoretical convergence property are designed and tested. Computational experience is reported for a hypothetical case study based on different parameter inputs and the two alternative partitioning strategies. The results demonstrate that the proposed approach dominates the commercial global optimizer BARON by more robustly yielding provable optimal solutions that are, on an average, better by 14.73% in terms of the objective function value, while consuming comparable computational effort.

Finally, we investigate the problem of determining stability margins in the context of designing robust *control systems*. As far as we know, this problem has not been tackled by the OR community, and this research leads a pioneering effort towards applying nonconvex optimization theory for computing D -stability margins in control systems. Depending on the value of p in the l_p -norm distance measure-based objective function employed in this context, we are confronted with differently structured problems, each of which requires certain specialized modifications of a proposed general algorithmic framework. Our research provides novel model formulations coupled with

RLT-based global optimization techniques for exactly computing, for the first time in the literature, D -stability margins for all possible values of p . Computational results for standard problems in the literature strongly indicate the efficacy of the proposed solution methodology with respect to determining global optimal solutions for this class of problems. On an average, the proposed optimization approach required only 54.61% of the CPU time taken by BARON and yet yielded solutions that are better in terms of objective function value by 3.16%.

A common theme in the study of the five aforementioned challenging nonconvex factorable programs is the development of tight model formulations and relaxations, and the design of effective algorithmic procedures that are not only theoretically convergent, but also yield a more robust solution methodology in practice, in comparison with the contemporary incumbent commercial global optimizer BARON. We hope that the experience gleaned from these reformulation-based modeling and algorithmic investigations are incorporated within global optimization software to make them more robust and effective, thereby advancing the frontiers of nonconvex optimization in both theory and practice.

*This work of mine, I dedicate to thee
my beloved Aparna san
For the many days in life with you, I foresee
are so beautiful, oh! meri jaan*

Acknowledgements

In the midst of the hectic activity that usually accompanies the finale of any massive undertaking, I want to pause, take a deep breath, and convey my sincere gratitude to all the people who played a part in finishing my doctoral dissertation. First and foremost, I express my heartfelt thanks to Dr. Hanif Sherali for all his help, guidance, and support. I can with no hesitation say that he has had a profound influence on me, not only as a teacher and researcher, but also as a person. I hope that I can remember, and dutifully carry with me, all the valuable lessons that I have learnt from him during the past five years. I also gratefully acknowledge the contributions and help given by the rest of my committee. Specifically, I wish to thank Dr. Subhash Sarin, for the many conversations that we have had and the wonderful advice that I have received from him. I also thank Dr. Koelling for his numerous encouragements, Dr. Barbara Fraticelli for her useful insights, and finally, Dr. Terry Herdman for all his help within the classroom and beyond.

Overwhelming thanks to my family, and in particular, to Appa and Amma, and to Anna and Reva, for all the love and affection that they have showered on me. They have been a constant source of support and enthusiasm, and I truly hope that I can live up to their expectations. A special thanks to Archana and Atul for all their hospitality and fun-filled moments that I have enjoyed with them.

To the many others who have made my stay at Virginia Tech so memorable, I truly thank you all. Here are some of them: Churlzu and Chawalit (office-mates in #206, Durham) who being one step ahead gave me a good understanding of graduate student life to come; Mei, Michael, and Ahmed (office-mates in #514 Whittemore) along with their gossip; the rest of the OR graduate student group; Gopa and Sathadamus (room-mates in #4400A, Terrace View) for distracting me time and again; and lastly, Lovedia, Dot and Kim, for all their diligent help.

Finally, my beloved fiancée, Apsy, for all that she has done to make my ambition a reality, including everyday telephone conversations, long drives on the interstates, letting me work on my laptop watching football all day long, for just being there, and for showing me that life can be so much fun when you love someone, I don't think I can find the words to amply say, thank you.

Table of Contents

Abstract	ii
Acknowledgements	viii
List of Tables	xi
List of Figures	xiv
Introduction	1
1.1. Preface	1
1.2. Motivation	2
1.3. Contributions of this dissertation	4
1.4. Organization of this dissertation	6
Literature Review	8
2.1. Introduction to global optimization	8
2.2. Polynomial and factorable programming problems	9
2.3. BARON and related global optimization software	15
Hard and Fuzzy Clustering Problems	18
3.1. Hard clustering problem	19
3.1.1. Problem reformulations and RLT-based algorithm	21
3.1.2. Computational results	31
3.1.3. Summary, conclusions and extensions for further research	41
3.2. Fuzzy Clustering Problem	44
3.2.1 Modeling and reformulation	45
3.2.2. Computational results	54
3.2.3. Summary, conclusions and extensions for further research	59
Risk Management with Equity Considerations	61
4.1. Emergency response model	64

4.2.	Branch-and-bound algorithm to solve Problem ERM	70
4.3.	Computational case study	75
4.4.	Summary, conclusions, and extensions for further research	85
Cascading Risk Management using an Event Tree Optimization Approach		87
5.1.	Event tree optimization model	91
5.2.	Global optimization branch-and-bound algorithms for solving Problem ETO	100
5.3.	Computational case study	105
5.4.	Summary and conclusions	111
Control of Linear Systems		113
6.1.	Formulation of D -stability margin problem	116
6.2.	Computation of D -stability margins	117
6.3.	Global optimization of D -stability margin problem	119
6.4.	Computational experience	128
6.5.	Discussions and conclusions	135
Conclusions and Future Research		138
7.1.	Summary and conclusions	138
7.2.	Future research	141

List of Tables

3.1.1.	Attributes for the ten data points in \mathbb{R}^2 for the illustrative example.	32
3.1.2.	Relative performance of the proposed optimization approach versus the k -means algorithm, measured in terms of different parameters, for three cluster centers.	35
3.1.3.	Relative performance of the proposed optimization approach versus the k -means algorithm, measured in terms of different parameters, for five cluster centers.	35
3.1.4.	Variations of Problem HCP4 to test the effectiveness of the different symmetry defeating strategies, measured in terms of gap ratio at node zero and the CPU time.	36
3.1.5.	Variations of Problem HCP4 to test the effectiveness of the different bounding constraints measured in terms of the gap ratio at node zero and the CPU time.	37
3.1.6.	Performance of the different branching strategies, measured in terms of the number of nodes enumerated and the CPU time.	38
3.1.7.	The performance of HCP4 via the default strategies of CPLEX-MIP 8.1.0 and HCP3 via BARON, measured in terms of CPU time.	39
3.1.8.	Results for the proposed approach and the k -means algorithm for large problem instances having three cluster centers.	40
3.1.9.	Results for the proposed approach and the k -means algorithm for large problem instances having five cluster centers.	41
3.2.1.	Relative performance of the proposed optimization approach versus the FCMA procedure for three cluster centers.	55
3.2.2.	Relative performance of the proposed optimization approach versus the FCMA procedure for five cluster centers.	55
3.2.3.	Relative performance of solving problems FCP and FCP1 via BARON versus the proposed approach, for three cluster centers.	57
3.2.4.	Relative performance of solving problems FCP and FCP1 via BARON versus the proposed approach, for five cluster centers.	57
3.2.5.	Comparative results for the proposed approach versus the FCMA procedure for randomly generated problem instances having three cluster centers.	58
3.2.6.	Comparative results for the proposed approach versus the FCMA procedure for randomly generated problem instances having five cluster centers.	59
4.1.	Hazard incidence matrix.	75
4.2.	Response relevance matrix.	75

4.3.	Hazard-area ratings matrix (r_{jk} -values).	76
4.4.	Measures relating to attenuation factors (β_{ik} -values) of $1.0 \lambda_i$, $1.5 \lambda_i$, and $2.0 \lambda_i$ for L, M, and H, respectively.	77
4.5.	Total resource unit availability (b_i -values).	77
4.6.	Minimum resource assignments as a proportion of total availability (with L_{ijk} -values given in paratheses).	77
4.7.	Resource assignments for each hazard in each area (x_{ijk} -values) corresponding to $\mu_D = 0$ obtained by solving ERM(l^0, u^0) using BARON.	78
4.8.	Resource assignments for each hazard in each area (x_{ijk} -values) corresponding to $\mu_D = 100$ obtained by solving ERM(l^0, u^0) using BARON.	78
4.9.	Resource assignments for each hazard in each area (x_{ijk} -values) corresponding to $\mu_D = 0$ obtained by solving ERM using BARON.	79
4.10.	Resource assignments for each hazard in each area (x_{ijk} -values) corresponding to $\mu_D = 100$ obtained by solving ERM using BARON.	79
4.11.	Resource assignments for each hazard in each area (x_{ijk} -values) corresponding to $\mu_D = 0$ obtained by solving ERM using the proposed algorithm.	80
4.12.	Resource assignments for each hazard in each area (x_{ijk} -values) corresponding to $\mu_D = 100$ obtained by solving ERM using the proposed algorithm.	80
4.13.	Computational results obtained for comparing the various branching strategies.	82
4.14.	Optimal objective function values obtained by solving problems ERM and ERM(l, u) using BARON along with the percentage deviations from the optimal value found using the proposed algorithm (with $\varepsilon = 10^{-6}$).	82
4.15.	Resource assignments for each hazard in each area (x_{ijk} -values) corresponding to $\mu_D = 0$ obtained by solving ERM using an ad-hoc intuitive algorithm.	84
5.1.	Logit-model coefficients (a_{im}) corresponding to preventive resources.	105
5.2.	Logit-model coefficients (b_{jn}) corresponding to mitigation resources.	106
5.3.	Per-unit-costs (c_{im}) corresponding to allocating preventive resources.	107
5.4.	Per-unit costs (d_{jn}) corresponding to allocating mitigation resources.	107
5.5.	Lower bounds corresponding to the loss accrued at each end-node j .	107
5.6.	Global optimal preventive resource assignments for each node (q_{im} -values) obtained by solving Problem ETO using the proposed algorithms.	108

5.7.	Global optimal mitigation resource assignments for each node (r_{jn} -values) obtained by solving Problem ETO using the proposed algorithms.	108
5.8.	Global optimal preventive resource assignments for each node (q_{im} -values) obtained by solving Problem ETO using BARON.	109
5.9.	Global optimal mitigation resource assignments for each node (r_{jn} -values) obtained by solving Problem ETO using the proposed algorithms.	110
5.10.	Computational results obtained for comparing the two proposed algorithms.	110
6.1.	Optimal q -values and objective values for different (selected) values of z .	130
6.2.	Global optimal solutions for Example 1 corresponding to different values of p .	132
6.3.	Global optimal solutions for Example 2 corresponding to different values of p .	135

List of Figures

3.1.1.	Branch-and-bound tree illustrating the SOS branching strategy.	33
3.1.2.	Clustering patterns obtained by solving Problem HCP4 via the proposed algorithm and by the k -means algorithm.	33
4.1.	Polyhedral Outer Approximation for $z_{jk} = \ln(y_{jk})$ over $0 < l_{jk} \leq y_{jk} \leq u_{jk} \leq 1$.	69
4.2.	Sensitivity of the R -term and the equity terms in the objective function to variations in the equity parameter μ_D .	85
5.1.	Illustration of a binary event tree depicting the occurrence of cascading risk events initiated by a gas-line rupture.	88
5.2.	Illustration of the polyhedral outer-approximation strategy.	96
5.3.	Sensitivity of the objective function with respect to the parameters t and β .	111
6.1.	Basic structure of a control system.	113
6.2.	Polyhedral Outer-approximation for $x_k = c_k y_k^p$	126
6.3.	Optimal objective value as a function of z for Example 1.	131
6.4.	Graph displaying the root-locus of the 5000 randomly generated points for Example 1.	131
7.1.	A quadratic approximation for a nonconvex univariate function that misses both the global and local optima.	145

1. Introduction

1.1 Preface

Ever since the advent of the simplex algorithm, linear programming (**LP**) has been extensively used with great success in many diverse fields. The field of discrete optimization came to the forefront as a result of the impressive developments in the area of linear programming. Although discrete optimization problems can be viewed as belonging to the class of nonconvex programs, it has only been in recent times that optimization research has confronted the more formidable class of *continuous nonconvex optimization problems*, where the objective function and constraints are often highly nonlinear and nonconvex functions, defined in terms of continuous (and bounded) decision variables. Typical classes of such problems involve polynomial, or more general factorable functions. Examples of these types of nonconvex optimization problems arise in the areas of medicine, engineering design, optimal control, risk management, manufacturing, and finance.

Given the huge potential monetary benefits that can accrue from optimal designs and operational decisions, a plethora of solution methodologies have been prescribed for addressing optimization problems. While efficient solution techniques have been developed for nearly all types of convex optimization problems, there are only a handful of efficient algorithms for continuous nonconvex optimization problems. The primary source of difficulty in determining a global optimum to a nonconvex program stems from the fact that most generic algorithmic procedures tend to converge to one of (possibly) several local optima that exist in the corresponding solution space. Existing approaches for solving nonconvex optimization problems can be classified as either *heuristic*, which often lead to sub-optimal solutions, or *enumerative*, which prove to be computationally prohibitive when applied to large-scale problems. Moreover, these approaches are usually tailor-made to solve the particular problem under consideration, and are not necessarily generalizable to a wider class of problems.

In the light of this environment, an effective, exact approach for solving general classes of nonconvex programs, which is yet capable of exploiting the inherent special structures in the problem, would indeed go a long way in advancing the field of global

optimization. It is in this regard that the *Reformulation-Linearization/Convexification Technique (RLT)* rises to prominence. The RLT methodology involves designing a variant of the branch-and-bound strategy that is specialized to the practical application under consideration, and its performance is heavily dependent on the strength of the bounding mechanisms employed. In order to derive tight lower/upper bounds on the problem, it is essential to develop suitable model formulations and to then exploit their special structures.

This dissertation focuses on employing the RLT methodology to enhance model formulations and to design effective solution techniques for solving several practical instances of continuous nonconvex optimization problems, namely, the hard and fuzzy clustering problems, risk management problems, and problems arising in control systems.

1.2 Motivation

Although the basic concept of the RLT methodology of Sherali and Tuncbilek (1992), which is the prime focal starting point in this work, is rooted in its ability to optimally solve nonconvex polynomial programming problems that involve a general polynomial objective and constraint functions, this methodology, coupled with supporting approximation schemes, can be extended to solve more general classes of nonconvex optimization problems. In particular, discrete optimization problems such as mixed-integer 0-1 programming problems can be modeled as polynomial programs by representing the binary restriction on any variable x_j as a polynomial term $x_j(1 - x_j) = 0$, where $0 \leq x_j \leq 1$. Furthermore, if we consider a general linear bounded discrete optimization problem, where x_j can take on several general discrete values in a set $S_j = \{\theta_{jk}, k = 1, \dots, n_j\}$ (which may or may not even be integral), then we can represent this via an equivalent polynomial constraint: $\prod_{k=1}^{n_j} (x_j - \theta_{jk}) = 0$. Such a construct enables the use of the RLT methodology developed for polynomial programming problems to be applied to a wide range of discrete optimization problems as well. Specialized developments of the RLT for discrete problems that stem from such

reformulations appear in Sherali and Adams (1990, 1994, 1999) and Adams and Sherali (1986).

Notwithstanding this feature, the focus of this dissertation will be on exploiting the RLT methodology as a means for solving continuous nonconvex programming problems that include polynomial programs as well as more general factorable functions (see McCormick, 1976, and Sherali and Wang, 2001). The fundamental RLT approach for such problems operates in two phases. In the *Reformulation phase*, the nonpolynomial terms appearing in the problem are replaced with suitable polynomial approximating functions, and additionally, certain classes of implied polynomial constraints are appended to the problem. Subsequently, in the *Linearization /Convexification phase*, the resulting polynomial program is linearized by substituting a new variable for each distinct variable-product term that appears in the problem. Sometimes, under special circumstances, certain classes of convex constraints are retained to relate the new and original variables in the problem. The resulting higher-dimensional representation yields a linear (or convex) representation of the original nonconvex program. Indeed, employing higher-order polynomial constraints leads to generating a hierarchy of tighter relaxations. Moreover, by embedding these RLT relaxations in a suitably designed branch-and-bound methodology, global optimal solutions can be derived.

A noteworthy feature of the RLT is that the lowest-level relaxation in the aforementioned hierarchy, which generates polynomials of degree no greater than those of the polynomial terms originally present in the problem during the reformulation phase, has been demonstrated to generate very tight relaxations in practice. This has enabled the solution of computationally intractable nonconvex optimization problems to near optimality, often via a single or few linear programs. Recent advances in LP technology, capable of solving large-sized linear programs fairly efficiently, and the widespread availability of related commercial software, has provided the facility to solve practical instances of nonconvex optimization problems to a sufficient degree of accuracy with manageable computational effort.

It is important to reiterate that the derivation of tight formulations for nonconvex optimization problems is an essential step towards developing effective solution techniques. The central idea behind constructing good model formulations is to obtain a

tight relaxation that is significantly easier to solve, and yet provides a sufficiently accurate approximation for the original problem. Since some of the restrictions on the problem have been eliminated, a feasible solution to the relaxation gives a best-case bound for the original problem. On the other hand, any feasible solution to the original problem provides an incumbent value or a worst-case bound, and these two bounds can be used in conjunction to search for globally optimal solutions. Obviously, the tighter the relaxation, the closer are the two bounds, thereby leading to a more computationally effective search process.

In this dissertation, our focus will be to enhance existing methods and concepts for obtaining such tight reformulations for a wide range of polynomial and factorable programs using the RLT methodology. Some techniques that are used in obtaining relaxations for nonconvex optimization problems are convex/concave outer-envelope processes, piecewise-linear approximations, and variable transformation strategies. Specifically, we center our attention towards enhancing model formulations and prescribing robust solution methodologies for the hard and fuzzy clustering problems, certain risk management problems, and problems arising in the analysis of control systems.

1.3. Contributions of this Dissertation

Under the umbrella of the broad RLT framework, the contributions of this dissertation focus on developing models and algorithms along with related theoretical and computational results pertaining to three specific application domains. First, we prescribe an RLT-based framework geared towards solving the hard and fuzzy clustering problems. In the basic construct, through appropriate surrogation schemes and variable substitution strategies, we derive strong polyhedral approximations for the polynomial functional terms in the problem, and then rely on the demonstrated (robust) ability of the RLT for determining global optimal solutions for polynomial programming problems. The convergence of the proposed branch-and-bound algorithm follows from the tailored branching strategy coupled with consistency and exhaustive properties of the enumeration tree. In the second endeavor, we examine two risk management problems, providing novel models and algorithms. Finally, in the third part, we provide a detailed

discussion on studying stability margins for control systems using polynomial programming models along with specialized solution techniques.

We begin by studying the hard and fuzzy clustering problems. The hard clustering problem is first reformulated by generating additional valid inequalities based on approximations to the convex hull of the data points, along with certain symmetry-defeating strategies. Then, a tight equivalent 0-1 linear mixed-integer programming representation is derived and a specialized branch-and-bound algorithm is designed to determine a global optimal solution. Results based on computational experiments performed using standard as well as synthetically generated data sets establish the efficacy and robustness of the proposed approach, in contrast with the popular k -means algorithm, as well as in comparison with the global optimization package BARON. Similar to the hard clustering case, we present a global optimization algorithm to solve the *fuzzy* clustering problem, where each data point is to be assigned to (possibly) several clusters, with a membership grade assigned to each data point that reflects the likelihood of the data point belonging to that cluster. The results validate the robustness of the proposed algorithmic procedure and exhibit its dominance over the popularly used FCMA clustering technique.

Next, we develop the basis of a *decision support system* for the macro-level problem of allocating certain available emergency response resources to mitigate risks under equity considerations. More specifically, consider a situation in which an emergency is underway and critical response decisions must be made. In order to mitigate the hazards, the emergency manager would typically call into play a variety of available resources. The problem of allocating these resources so as to best control the damage that has already occurred, subject to resource, budget, and equity constraints can be formulated as a nonconvex program, for which we derive a tight linear programming relaxation. This relaxation is embedded within a specialized branch-and-bound procedure, and the proposed method is proven to converge to a global optimum. Various alternative partitioning strategies that could potentially be employed in the context of this branch-and-bound framework, while preserving the theoretical convergence property, are also explored. Computational results are reported for a hypothetical scenario based on different parameter inputs and alternative branching strategies, and comparisons with the

commercial software BARON as well as an ad-hoc intuitive method are presented. These results indicate that our proposed algorithm yields significantly improved solutions when compared to these other methods.

Delving further, we consider the strategic planning micro-level problem of allocating certain available preventive and mitigation resources to respectively reduce the failure probabilities of system safety features and the total expected loss arising in the aftermath of a hazardous event. Using an event tree optimization approach, the resulting cascading risk scenario problem is modeled as a nonconvex factorable program. We derive a tight linear programming relaxation along with several theoretical insights that serve to lay the foundation for designing a specialized branch-and-bound algorithm that is proven to converge to a global optimum. Two alternative partitioning strategies that preserve the theoretical convergence property are designed and tested in the context of this branch-and-bound framework. Computational experience is reported for a hypothetical case scenario based on different parameter inputs and using the two alternative partitioning strategies. The results demonstrate that the proposed approach dominates the commercial software BARON by more robustly yielding provable optimal solutions, while consuming comparable effort.

Finally, we address the problem of determining stability margins in the context of designing robust *control systems*. We prove that the problem of computing the maximum stability perturbation limits can be modeled as an optimization problem that belongs to the class of nonconvex programs addressed in this dissertation. Depending on the value of the parameter p , the l_p -norm based objective function yields different objective function and constraint terms. This research introduces the OR community to a new class of problems arising in control theory, and delineates (RLT-based) global optimization algorithms for computing D -stability margins for the first time in the context of control systems.

1.4. Organization of this Dissertation

The remainder of this dissertation is organized as follows. Chapter 2 presents a brief literature review on some of the existing methodologies for solving nonconvex optimization problems and certain related issues. From Chapter 3 onwards, we

demonstrate the applicability of RLT methodology by designing specialized variants of the RLT-based branch-and-bound algorithm for different practical instances of nonconvex optimization problems, and we report related computational experience. Specifically, Chapter 3 deals with the problem of determining global optimal solutions for the hard and fuzzy clustering problems, Chapters 4 and 5 discuss risk management applications, and Chapter 6 presents polynomial optimization techniques for analyzing the stability of automatic control systems. Finally, Chapter 7 provides a summary and delineates extensions to the RLT methodology, including the solution of black-box optimization problems.

2. Literature Review

2.1. Introduction to Global Optimization

Ever since the mid 1980s, nonconvex optimization models have been increasingly explored in the context of various operational and design problems. In a broad sense, a nonconvex programming problem can be defined as follows:

$$\begin{aligned} \mathbf{P}: \quad & \text{Minimize} && f(x) \\ & \text{subject to} && g(x) \leq 0 \\ & && x \in X, \end{aligned} \tag{2.1}$$

where $f: X \rightarrow \mathbb{R}$ is the objective function, $g(x): X \rightarrow \mathbb{R}^m$ are the structural constraints that define the feasible region or search space, with at least one of the functions f or g being nonconvex, $X \subseteq \mathbb{R}^n$ is a compact set (usually represented via box-constraints), and $x = (x_1, \dots, x_n)^\top$ denotes a feasible solution in this search space. Judging by the considerable research that has been done and the vast literature that is available, it is evident that global optimization of nonconvex programming problems has generated a great deal of interest.

With the development of complexity theory in the early 1970s, it became clear that since nonconvex global optimization problems were (often) NP-hard, it would be a computationally onerous task to determine exact and efficient solution procedures to determine the global optimum for such problems. Hence, a significant portion of optimization research was dedicated towards designing *heuristic* procedures that obtain good quality feasible solutions relatively quickly, while avoiding the quagmire of locating the true global optimum. Examples of such techniques are simulated annealing, genetic algorithms, and other meta-heuristic approaches. However, since nearly a decade, a rapid growth in computing technology has spurred research into developing *exact* algorithms for solving difficult nonconvex programming problems.

Despite limiting the scope to describing exact algorithms for solving nonconvex programs, it is still a daunting task to provide a comprehensive literature review on this

topic. Thus, this chapter only reviews specific methodologies for solving (nonconvex) polynomial and factorable optimization problems that are relevant to this research effort, and lays the foundation for developing effective solution techniques based on the RLT methodology. The interested reader may refer to books by Horst (1990), Horst and Tuy (1993), Horst and Pardalos (1995), and Pinter (1996) for a general discussion on global optimization. Furthermore, noting that the development of global optimization software for automating existing algorithmic procedures has now become an integral part of optimization research, a brief survey of the (currently dominant) global optimizer BARON is also presented in this chapter. (In this dissertation, BARON has been used to serve as a benchmark for comparing the results obtained via RLT-based algorithms for many computational experiments, and it is therefore particularly relevant to obtain a glimpse at the strategy employed by BARON.)

2.2. Polynomial and Factorable Programming Problems

Polynomial programs deal with seeking a global optimum to a polynomial objective function subject to a set of polynomial constraints, all defined in terms of continuous, bounded decision variables. A polynomial program can be mathematically formulated as follows:

$$\begin{aligned}
 \text{PP}(\Omega): \quad & \text{Minimize} \quad \{ \phi_0(x) : x \in Z \cap \Omega \}, \\
 \text{where,} \quad & Z = \{ x : \phi_r(x) \geq \beta_r, r = 1, \dots, R_1, \phi_r(x) = \beta_r, r = R_1 + 1, \dots, R \}, \text{ and} \\
 & \Omega = \{ x : 0 \leq l_j \leq x_j \leq u_j < \infty, j = 1, \dots, n \}, \text{ and where} \\
 & \phi_r(x) \equiv \sum_{t \in T_r} \alpha_{rt} \left[\prod_{j \in J_{rt}} x_j \right], r = 0, \dots, R. \tag{2.2}
 \end{aligned}$$

Here, T_r is an index set for the terms defining $\phi_r(\cdot)$, and α_{rt} are real coefficients for the polynomial terms $\prod_{j \in J_{rt}} x_j$, $t \in T_r$, $r = 0, \dots, R$. Note that a repetition of indices is allowed within J_{rt} . For example, if $J_{rt} = \{1, 2, 2, 3\}$, then the corresponding polynomial term is $x_1 x_2^2 x_3$. Denote $N = \{1, \dots, n\}$ and define $\bar{N} = \{N, \dots, N\}$ to be composed of δ

replicates of N , where δ is the maximum specified degree of any polynomial term appearing in $PP(\Omega)$. Then each $J_{rt} \subseteq \bar{N}$, with $1 \leq |J_{rt}| \leq \delta$, for $t \in T_r$, $r = 0, 1, \dots, R$.

Determining a global optimum to a polynomial program, as defined in (2.2), is a computationally difficult task (theoretically, this is NP-Hard), and thus requires the use of specialized algorithms. Polynomial programs have received a considerable amount of attention in the literature, with several solution approaches having been developed with varying degrees of success. Due to the involved complexity, optimization algorithms have also been designed for special cases of polynomial programs rather than for the general case.

In the context of nonlinear integer programs, a linearized cutting plane method for 0-1 constrained polynomial programming problems was proposed by Balas and Mazzola (1984a, b) and was shown to perform fairly well in practice. Later on, various linearization, algebraic, and cutting plane methods were developed for nonlinear 0-1 programs. A concise method to solve the mixed-integer 0-1 polynomial programming problem using additional 0-1 variables and auxiliary constraints was proposed by Chang and Chang (2000). Sherali and Adams (1990, 1994) developed a hierarchy of representations and related relaxations for 0-1 pure and mixed-integer polynomial programs, leading to the convex hull representation. Adams *et al.* (1998) also studied certain persistency properties of the obtained relaxations.

For continuous polynomial programs, Belousov and Klatte (2002) used the result that an orthogonal projection along a recession direction of a convex polynomial set is a convex polynomial set, and went on to prove that an extension of the Frank-Wolfe theorem can be successfully applied to solve this class of problems. Note that this approach cannot be extended when the polynomials are nonconvex over the constraint set. An approximate approach towards finding a global optimum for polynomial programs under a specified tolerance limit was presented by Li and Chang (1998), making use of the representation of a continuous variable as the sum of a discrete variable and a bounded perturbation, and subsequently, linearizing the corresponding terms. Floudas and Visweswaran (1990a, b, 1995) demonstrated that quadratic programs, along with polynomial optimization problems, can be transformed into a form that is amenable to algorithmic manipulations. They described a global optimization algorithm

based on partitioning the variable set, and decomposing the given nonconvex program into *primal* and *relaxed dual* subproblems, which are then handled using methodologies influenced by generalized Benders' decomposition (see Geoffrion, 1972). Similarly, Aggarwal and Floudas (1990) also presented a variable-splitting, Benders' decomposition-based approach, and concluded that the starting point affects the efficiency of the algorithm in locating a global optimum. Other techniques (*e.g.* concave minimization, Lipschitzian optimization) as described in Horst (1990) and Horst and Tuy (1993), are promising techniques to solve polynomial programs. However, a majority of these algorithms either take advantage of the special structure of the problem that they address, or provide approximate solutions, or converge to local optima, and thus prove to be inadequate for globally solving the general class of polynomial programming problems.

A noteworthy exception that is capable of handling general polynomial programs is the RLT-based global optimization algorithm of Sherali and Tuncbilek (1992). For solving the general polynomial program, the RLT-based approach operates as follows. Given Ω , in order to construct the linear programming bounding problem $\mathbf{LP}(\Omega)$ using RLT, implied bound-factor product constraints are generated by using *distinct* products of the bounding factors $(x_j - l_j) \geq 0$ and $(u_j - x_j) \geq 0$, $j \in N$, taken δ at a time. These constraints can be expressed as follows:

$$F_\delta(J_1, J_2) \equiv \prod_{j \in J_1} (x_j - l_j) \prod_{j \in J_2} (u_j - x_j) \geq 0, \quad (2.3)$$

where $(J_1 \cup J_2) \subseteq \bar{N}$, $|J_1 \cup J_2| = \delta$.

After including the constraints (2.3) in the problem $\mathbf{PP}(\Omega)$, the substitution

$$X_J = \prod_{j \in J} x_j, \quad \forall J \subseteq \bar{N}, \quad (2.4)$$

is applied to linearize the resulting problem, where the indices in J are assumed to be sequenced in nondecreasing order, and where $X_{\{j\}} \equiv x_j$, $\forall j \in N$, and $X_\emptyset \equiv 1$. Note that each distinct set J produces one distinct X_J variable. This yields $\mathbf{LP}(\Omega)$.

To solve $PP(\Omega)$ to global optimality, $LP(\Omega)$ is embedded in a branch-and-bound algorithm to compute lower bounds on the underlying polynomial program. A procedure of this type proposed by Sherali and Tuncbilek (1992, 1995) essentially involves the partitioning of the original set Ω into sub-hyperrectangles, each of which is associated with a node of the branch-and-bound tree. A partitioning rule geared towards identifying the variable that contributes the most to the discrepancy between a new RLT variable that contains it and the associated corresponding nonlinear product that this RLT variable represents is prescribed, and the motivation is to drive all such discrepancies to zero by creating partitions that would induce the variables to achieve their bounds, leading to a global optimal solution. (Sherali (1998) has further refined this RLT-based branch-and-bound procedure to handle polynomial programming problems having rational exponents as well.)

Sherali and Tuncbilek (1997) extended this methodology to derive an alternative linearization technique. Akin to the above, in this approach, a transformation is used to quadrify a given polynomial problem, *i.e.*, to transform the polynomial program into an equivalent quadratic polynomial program. Consider the set

$$A = \left\{ a = (a_1, a_2, \dots, a_n) \in Z_+^n : 0 \leq a_j \leq s_j, \forall j = 1, \dots, n, \text{ and } \sum_{j=1}^n a_j \leq \delta \right\}, \quad (2.5)$$

where Z_+^n denotes the set of nonnegative integral n -tuples, the quantities s_j , $\forall j = 1, \dots, n$, are specified bounds on the corresponding exponent terms a_j , and δ is the degree of the polynomial program. A new variable $R[a]$ was defined to represent the multinomial term $x^a \equiv \prod_{j=1}^n x_j^{a_j}$ for each $a \in A$. An equivalent quadrified polynomial was then defined in terms of the variables $R[a]$, $a \in A$. In this quadrification process, each polynomial term was represented as a product of *exactly* two $R[\cdot]$ variables. Note that such a representation scheme is not unique. In the next phase, a formulation that captures all such quadrifying entities within an encompassing equivalent quadratic framework was derived, which yields an “*exhaustively*” quadrified polynomial program that subsumes all possible quadrification transformations. Sherali and Tuncbilek (1997) proved that applying RLT directly to the original polynomial program provides a tighter

representation than applying it to this equivalent exhaustively quadrified problem. However, applying RLT to some equivalent quadrified problem can yield a computationally less burdensome relaxation. Sherali and Tuncbilek (1997, 1995) additionally provided several insights into the design of suitable RLT strategies that can be gainfully applied to solve challenging problem instances, and have reported related computational results.

Pursuing this idea of quadrification, note that in optimization literature, there exist a variety of solution methodologies for solving quadratic programming problems (refer Sherali and Tuncbilek, 1995, Audet *et al.*, 2000a, Floudas and Visweswaran, 1990, 1995). In particular, if suitably generated quadrification schemes are utilized for quadrifying polynomial programs, a significant advantage could then be derived. In this regard, the methodology of Shor (1990) deserves mention. Given a polynomial function, say $p(x)$, where $x = (x_1, \dots, x_n)^T$, Shor (1990) considered introducing new variables and performing quadratic substitutions of the form $x_i^2 = y_i$ and $x_j x_k = w_{jk}$. Recursively applying this substitution process yields a quadratic program involving the new and original variables. Shor then proceeded to solve this resulting quadratic program via a Lagrangian-dual based approach, and investigated conditions under which this process would yield no duality gaps.

As a further generalization of the RLT procedure, Lasserre (2001) discussed the problem of determining a global minimum for constrained (and unconstrained) multivariate polynomial programming problems by generating tight relaxations via *linear matrix inequalities* (LMIs). In essence, Lasserre coupled polynomial programs with semidefinite programming by viewing positive polynomials as sums of squares and moment sequences, and also demonstrated the equivalence of this problem to solving an infinite sequence of LMI problems.

Moving into the realm of factorable programming, the introduction of this class of problems is credited to McCormick (1976). By Fiacco and McCormick's definition (1968), factorable programs were viewed as a class of problems having nonlinear functions for which the Hessians possessed two special properties: (1) the Hessians can be computed exactly and efficiently, and (2) the Hessians occur as sums of outer-products

whose vector factors are gradients of terms in the factored sequence. Coupled with local search methods, such as the modified Newton method developed by Ghotb (1987) that deals with linearly constrained factorable programs, the above mentioned properties of the Hessian matrices of factorable programs were exploited by McCormick (1976, 1983) to obtain a global optimum via an inductive convex envelope construction process.

Since this initiative, it is surprising to note that no major algorithmic advances for solving factorable programs were made for over two decades. Beginning in the 1990s, some algorithmic developments were reported for solving (generalized) *linear multiplicative programs* (LMPs) and *convex multiplicative programs* (CMPs), which comprise a large subset of the class of nonconvex factorable programs. The thrust towards global optimization of LMPs was led by Konno and Kuno (1990), when they embedded the given nonconvex LMP in a higher dimensional space, and then applied a path-following algorithm for solving a sequence of convex programs that lead toward an optimum. Thereafter, Konno and Kuno (1992, 1995) extended this methodology to develop a parametric simplex-based approach, and provided other techniques for solving CMPs as well. Other results pertaining to CMPs have also appeared in Kuno *et al.* (1992, 1993). Later on, Ryoo and Sahinidis (2001, 2003) explored the development of good lower bounding procedures for multiplicative programs, and prescribed a global optimization branch-and-bound approach. As discussed in Chapter 1, the RLT-based global optimization approach of Sherali and Wang (2001) was the first major step toward solving general factorable programs in McCormick's (1976) work. Following this, Sahinidis and Tawarmalani (2002a, b) handle factorable programming problems by constructing *convex nonlinear* relaxations. These relaxations are subsequently synthesized into linear programming relaxations via the sandwich algorithm (Rote, 1992). Other techniques, such as the construction of polyhedral outer-approximation schemes, have also been investigated by Sahinidis and Tawarmalani (2002a) for certain classes of factorable programs.

Finally, in the last section of this literature review, we review the branch-and-reduce strategy employed by BARON, and provide some additional comments.

2.3. BARON and Related Global Optimization Software

Optimization problems having multiple local optima are encountered in many areas of engineering, economics, and decision sciences. Determining global optima for such problems is often a computationally difficult task, and requires the use of specialized software. The *Branch-and-Reduce Optimization Navigator* (**BARON**) is one of the few commercially available global optimization software that facilitates the solution of many specific classes of nonconvex programs (refer Sahinidis, 1996, Ryoo and Sahinidis, 1996, Tawarmalani and Sahinidis, 1999, 2002a, b). Given a nonconvex program as defined in (2.1), BARON attempts to determine a global optimum, by making use of two important techniques, namely *range-reduction* and enhanced *branch-and-bound* concepts, which lead to the acronym BARON.

Specifically, due to the assumed compactness of the set X in (2.1), problem P turns out to be bounded, and this property can be used in constructing lower bounding relaxations for P . For example, whenever f is lower semi-continuous over the feasible region, and g satisfies certain properties (*e.g.*, g is lower semi-continuous), the existence of a finite optimum can be guaranteed, given that the feasible region is nonempty. One of the standard approaches for solving problems of the type P is to construct a tight lower bounding problem, say \mathbf{R} , whose optimum provides a lower bound to P . This problem \mathbf{R} is referred to as a *relaxation* of P , and is usually constructed by viewing P in a higher dimension and/or deriving underestimates for f (Tawarmalani and Sahinidis, 2002b). Typically, the construction of a lower bounding relaxation problem entails the approximation of the objective function and an outer approximation of the region defined by each constraint. In most cases, the relaxations are defined in a manner that they become exact at the variable bounds and the tightness of a relaxation depends upon the tightness of the variable bounds. Also, the relaxed problem \mathbf{R} is (often) a convex program, whose solution can be determined with a relative ease in computational effort.

In the branch-and-bound scheme employed by BARON, an optimum solution l^* , realized at x^* , is obtained for problem \mathbf{R} , which yields a valid lower bound to P . Evaluating P at x^* , if it is feasible (or else, modifying this to a possibly feasible solution via a local search method) yields a valid upper bound, say u^* . For some tolerance

limit $\varepsilon \geq 0$, if $u^* - l^* \leq \varepsilon$, then the algorithm terminates. Otherwise, the compact set X is partitioned into two subregions, and a traditional branch-and-bound methodology is followed. This branch-and-bound scheme can be represented on a tree whose nodes and branches correspond to solving relaxations and partitioning the search space, respectively. During the subdivision process, the nodes of the search tree whose lower bounds are greater than or equal to the (current) upper bound (within the ε tolerance) are discarded (fathomed) from further exploration since they clearly would not lead to superior solutions than the current known incumbent. Moreover, BARON is also equipped with two types of range-reduction techniques to facilitate the reduction of the search space, known as *optimality-based range-reduction* and *feasibility-based range-reduction* (see Sahinidis, 1996, for further details). These range-reduction principles are derived from nonlinear duality theory and have been demonstrated to significantly improve BARON's computational efficiency towards solving nonconvex optimization problems. (Refer Sahinidis and Tawarmalani (2003) for a good overview of the modeling framework in BARON.)

In a recent work, Sahinidis and Tawarmalani (2003) have initiated the process of making BARON more attuned towards reformulation-linearization kinds of techniques. Since the capability of any algebraic modeling system is significantly enhanced by not only computing good lower bounds, but strong upper bounds as well, BARON is currently being upgraded to include a tool that allows the modeler to furnish *nonlinear reformulations*, which enhance the global solver's capability. Thus, RLT-based constructs are the latest addition for gearing BARON towards solving larger and more difficult nonconvex programs. In particular, state-of-the-art techniques for solving pooling problems from chemical engineering are demonstrated to be strongly dependent on RLT-based linearization techniques (see Tawarmalani and Sahinidis, 2002, and Audet *et al.*, 2000b). Indeed, a comparison of complete global optimization solvers conducted by Neumaier *et al.* (2004) concludes that BARON is currently the most dominant global optimization solver that is commercially available. Other solvers such as LINGO (Lindo Systems, Inc., 2005), OQNLP (GAMS Solver descriptions, 2003), LGO (Pinter, 1996), and COCOS (Shcherbina *et al.*, 2004, and Schichl, 2004) follow suit.

We are now ready to demonstrate the broad applicability of the RLT methodology by specializing this for various practical applications such as hard and fuzzy clustering problems, risk management problems, and problems encountered in control systems. The underlying structure of many of these problems conforms to that of polynomial or factorable programming problems, thereby facilitating an application of the RLT concept. We hope that modeling and algorithmic developments in this dissertation are incorporated within global optimization software to make them more robust and effective, thereby advancing the computational solvability of nonconvex programs.

3. Hard and Fuzzy Clustering Problems

In many applications, data is generated that needs to be analyzed and deciphered in order to extract patterns or information from it. One approach to sift this data is to solve the underlying *clustering problem*. In a broad sense, this involves the process of partitioning the given data set into subsets called *clusters*, such that some accumulated distance measure between points belonging to common clusters is minimized. Several clustering approaches have been developed to effectively analyze and interpret large volumes of data information. Such clustering problems arise in a wide scope of applications related to cellular manufacturing, medicine, archaeology, and marketing (see Hartigan (1975) for a detailed survey on applications of cluster analysis).

Mangiameli *et al.* (1996) have shown that the clustering problem is NP-Hard and thus, finding a global optimum to this problem is a computationally onerous task. However, a significant reduction in computational effort can be achieved by considering judiciously defined subsets of the original data set and applying a more refined partitioning scheme, separately to each such subset, to arrive at the final clustering pattern. This concept has led to a variety of clustering approaches such as statistical methods, self-organizing maps, hierarchical clustering, and a limited number of optimization techniques.

The most popular among these methods is the hierarchical clustering method. Hierarchical clustering of data sets can be achieved by two types of splitting methods: partitive splitting and agglomerative merging. The hierarchical clustering technique that is most widely in vogue is the agglomerative approach (see Ward, 1963, Sultan *et al.*, 2002). This begins with individual data points being singleton clusters, and then at successive iterations, merges them to generate a tree structure. This tree is referred to as a *dendrogram*. The dendrogram is cut off at some level at which a large distance is observed between pairs of clusters. This approach does not usually provide a unique clustering, and in fact, does not guarantee that intra-cluster distance is minimized. To obtain an optimal clustering using this approach, the dendrogram must be subdivided at several points (Sultan *et al.*, 2002).

In contrast with this method, partitive clustering initially divides the data set into a predefined number of clusters by minimizing some criterion (usually a distance measure). Then, at each iteration, the intra-cluster distance is minimized and the inter-cluster distance is maximized (Sultan *et al.*, 2002). In general, solution techniques based on hierarchical clustering have problems related to robustness and uniqueness of the solution obtained (Lukashin and Fuchs, 2000). On the other hand, the limited number of optimization techniques that are available cannot guarantee that the derived solution is a global optimum. Moreover, while the number of clusters used is typically prescribed as a fixed, external parameter for the algorithm being utilized, there is some interest in also simultaneously determining an optimal number of clusters to use, perhaps given a fixed cost associated with constructing each cluster (see Dubes, 1987, Jung *et al.*, 2003).

Another important factor involved in solving clustering problems is the distance measure under consideration. Obviously, optimal clustering depends on the distance measure being used. Distance measures are divided into *metric* and *semi-metric* measures (Sultan *et al.*, 2002), and most hierarchical procedures (that are based on the nearest neighbor approach) utilize either one of these measures. A semi-metric distance measure is one that satisfies the following properties for any two vectors i and j in a given data set: (1) the distance between i and j is positive, *i.e.*, $d_{ij} > 0$; (2) $d_{ij} = d_{ji}$, and (3) $d_{ii} = 0$. In addition to the above properties, if a distance measure satisfies the triangle inequality, *i.e.*, $d_{ij} + d_{jk} \geq d_{ik}$, then it qualifies as a metric measure.

In this chapter, we first consider the *hard clustering* problem wherein each data point must be assigned to *exactly* one cluster. Subsequently, we consider clustering problems where a data point may belong to several clusters with a membership grade assigned to each data point that represents the likelihood of the data point belonging to that cluster. Such a problem is referred to as a *fuzzy clustering* problem (the word fuzzy is derived from fuzzy programming, and reflects the fact that the specific cluster to which a data point belongs is only fuzzily identified, and is not described deterministically).

3.1. Hard Clustering Problem

The *hard clustering problem* (**HCP**) can be defined as follows. Given a set of n

data points, each having some s attributes, we are required to assign each of these points to exactly one of some c clusters (where c is given), so as to minimize the total squared Euclidean distance between the data points and the centroid of the clusters to which they are assigned. That is to say, if data point i , having a location descriptor $a_i \in \mathbb{R}^s$ is assigned to cluster j having a to-be-determined centroid $z_j \in \mathbb{R}^s$, then the associated penalty is assumed to be proportional to the square of the straight line distance separation between a_i and z_j in \mathbb{R}^s . An optimal solution to the clustering problem determines the cluster configuration such that the sum of all such distances is minimized. This problem can be mathematically stated as follows.

$$\text{HCP:} \quad \text{Minimize} \quad \sum_{i=1}^n \sum_{j=1}^c w_{ij} \|a_i - z_j\|^2 \quad (3.1.1a)$$

$$\text{subject to} \quad \sum_{j=1}^c w_{ij} = 1, \quad \forall i = 1, \dots, n, \quad (3.1.1b)$$

$$w \geq 0, \quad (3.1.1c)$$

where $a_i = (a_{ik}, k = 1, \dots, s)^T$, and $z_j = (z_{jk}, k = 1, \dots, s)^T$, and the norm $\| \cdot \|$ in (3.1.1a) represents the Euclidean distance between the two points in its argument in the s -dimensional space under consideration. We assume that $n > c$, because otherwise, the problem would be trivially solved by simply designating each point to constitute a cluster by itself. Observe also that for any fixed z , w will automatically be binary-valued at a resultant extreme point optimum.

The hard clustering problem has been extensively dealt with in the literature and there are several approaches that have been explored to solve this problem. The first attempt to solve the clustering problem was by using the k -means algorithm (Forgy, 1966, and McQueen, 1967). This method is widely used in practice, but often fails to produce a global optimum. Several optimization techniques such as dynamic programming (Jensen, 1969), convexity cuts (Selim, 1982), alternative cutting plane algorithms (Groetschel and Wakabayashi, 1989), lagrangian relaxation methods (Mulvey and Crowder, 1979) and integer programming formulations coupled with branch-and-bound strategies (Vinod, 1969, Rao, 1971, and Koontz *et al.*, 1975) have been used to solve the hard clustering problem. Of recent flavor are meta-heuristic search methods such as simulated annealing, tabu search, and the genetic algorithm. Klein and Dubes

(1989) and Selim and Al-Sultan (1991) were the first to study a simulated annealing approach in this context, and thenceforth, several other modifications of this procedure have been proposed. Al-Sultan (1995) developed a tabu search algorithm, and Bhuyan *et al.* (1991) and Krovi (1992) have advocated a framework using the genetic algorithm to solve the hard clustering problem. Computational experience along with a comparison between four heuristic algorithms that solve the hard clustering problem has been provided by Al-Sultan and Khan (1996).

In this research effort, we design an optimization approach based on the *Reformulation-Linearization Technique* (RLT) (refer Sherali and Adams, 1990, 1994, 1999, and Sherali and Tuncbilek, 1992, 1997) to solve the hard clustering problem. The underlying nonlinear, discrete optimization problem is transformed into an equivalent 0-1 mixed-integer program having a tight linear programming (LP) relaxation as prescribed by the RLT, and a specialized algorithm is designed to derive a global optimum.

The remainder of Section 3.1 is organized as follows. Section 3.1.1 provides a series of enhanced formulations of the problem based on RLT constructs as well as the derivation of certain classes of valid inequalities. Accordingly, a tailored branch-and-bound global optimization algorithm is also delineated in Section 3.1.1. Section 3.1.2 presents computational results using certain standard test problems from the literature as well as using larger synthetically generated data sets, and explores the performance of different formulations and implementation strategies. Finally, Section 3.1.3 concludes the paper with a summary and a discussion on further avenues for research in this area.

3.1.1. Problem Reformulations and RLT-based Algorithm

From the hard clustering problem, as defined in (3.1), for a fixed w , optimality of the resulting convex objective function in z requires that

$$\sum_{i=1}^n w_{ij} (z_{jk} - a_{ik}) = 0, \quad \forall j, k, \quad (3.1.2a)$$

that is,

$$z_{jk} = \frac{\sum_{i=1}^n w_{ij} a_{ik}}{\sum_{i=1}^n w_{ij}}, \quad \forall j, k, \quad (3.1.2b)$$

where the denominator in (3.1.2b) is positive at optimality under our assumption that $n > c$.

Consequently under the conditions (3.1.2a) and (3.1.2b), we have that the objective function (3.1.1a) is equivalently given by

$$\begin{aligned}
& \sum_{i=1}^n \sum_{j=1}^c \sum_{k=1}^s w_{ij} (z_{jk} - a_{ik})^2 \\
&= \sum_{i=1}^n \sum_{j=1}^c \sum_{k=1}^s w_{ij} (z_{jk} - a_{ik}) z_{jk} - \sum_{i=1}^n \sum_{j=1}^c \sum_{k=1}^s w_{ij} (z_{jk} - a_{ik}) a_{ik} \\
&= \sum_{i=1}^n \sum_{j=1}^c \sum_{k=1}^s w_{ij} a_{ik}^2 - \sum_{i=1}^n \sum_{j=1}^c \sum_{k=1}^s a_{ik} w_{ij} z_{jk}.
\end{aligned} \tag{3.1.3}$$

By (3.1.1b), noting that $\sum_{i=1}^n \sum_{j=1}^c \sum_{k=1}^s w_{ij} a_{ik}^2 = \sum_{i=1}^n \sum_{k=1}^s a_{ik}^2$, a constant, we have that HCP can be equivalently solved via the following problem.

$$\text{HCP1:} \quad \text{Maximize} \quad \sum_{i=1}^n \sum_{j=1}^c \sum_{k=1}^s a_{ik} w_{ij} z_{jk} \tag{3.1.4a}$$

$$\text{subject to} \quad z_{jk} \sum_{i=1}^n w_{ij} - \sum_{i=1}^n a_{ik} w_{ij} = 0, \quad \forall j, k \tag{3.1.4b}$$

$$\sum_{j=1}^c w_{ij} = 1, \quad \forall i = 1, \dots, n \tag{3.1.4c}$$

$$w \text{ binary}, \tag{3.1.4d}$$

where (3.1.4d) has been explicitly imposed to exploit this optimality condition in the algorithmic process.

Note that by (3.1.4b), if we denote for any given solution w to HCP1, the sets

$$S_j = \{i : w_{ij} = 1\}, \quad \forall j, \tag{3.1.5}$$

then we have

$$z_{jk} = \sum_{i \in S_j} a_{ik} / |S_j|, \quad \forall k, \text{ for each } j, \tag{3.1.6}$$

or that the vector z_j is a convex combination (with equal weights) of the points a_i , $i \in S_j$. Let us now define $I_j \subseteq \{1, \dots, n\}$ as the set of potential points $i \in \{1, \dots, n\}$ that are

assignable to cluster j (in the absence of any relevant information or algorithmic restrictions, we would have $I_j \equiv \{1, \dots, n\}$), $\forall j = 1, \dots, c$, and let us denote

$$\begin{aligned} H(I_j) &= \text{conv} \{a_i : i \in I_j\} \\ &\subseteq \left\{ z_j : \sum_{k=1}^s \gamma_{qk}^j z_{jk} \leq \gamma_{q0}^j \text{ for } q = 1, \dots, Q_j \right\} \equiv \bar{H}(I_j), \text{ say, } \forall j, \end{aligned} \quad (3.1.7a)$$

where the set of Q_j inequalities in (3.1.7a) defines some bounded superset $\bar{H}(I_j)$ of $H(I_j)$. Observe that for notational convenience, we have used the superscript j in lieu of I_j for the inequalities describing $\bar{H}(I_j)$ in (3.1.7a), and also, note that for isomorphic subsets of $\{1, \dots, n\}$, we can use the same description of $\bar{H}(\cdot)$. In the simplest case, $\bar{H}(I_j)$ might be taken as an enclosing hyperrectangle as expounded below. Note that $H(I_j)$ is efficiently computable in polynomial time for points in two-dimensions using the method described in Manber (1989). (For example, the Graham's scan algorithm produces the convex hull in $O(n \log n)$ steps.) However, for higher dimensions, computing the convex hull can prove to be an expensive task. Nevertheless, under some specific assumptions, it has been shown in the literature that the convex hull can be obtained for higher dimensions using techniques such as neural networks (refer Leung *et al.*, 1997), cutting planes (Chazelle, 1991), and direct convex hull computations for convex polyhedra (refer Klapper, 1987, Balas, 1988.). In the context of our problem, we can gainfully employ any such technique to derive suitable valid inequalities for constructing $\bar{H}(I_j)$, $\forall j$. For simplicity, regardless of problem dimension, we will take $\bar{H}(I_j)$ to be a hyperrectangle that bounds the collection of points a_i , $i \in I_j$, as defined below for each j .

$$\bar{H}(I_j) = \{z_j : \alpha_k^j \leq z_{jk} \leq \beta_k^j, k = 1, \dots, s\} \quad (3.1.7b)$$

where,

$$\alpha_k^j = \min\{a_{ik} : i \in I_j\}, \quad \forall k, \text{ and } \beta_k^j = \max\{a_{ik} : i \in I_j\}, \quad \forall k, \text{ for each } j. \quad (3.1.7c)$$

Additionally, we could incorporate other valid inequalities that are valid for $H(I_j)$ within (3.1.7b). In order to maintain generality in presentation of these various viable algorithmic strategies, we will henceforth assume that some such suitable set $\bar{H}(I_j)$ as given by (3.1.7a) has been obtained.

Now, we can impose the implied constraints defining $\bar{H}(I_j)$ for each j within HCP1, where, prior to any further analysis, $I_j \equiv \{1, \dots, n\}$, $\forall j$. (Subsequently, we will be modifying the sets I_j iteratively in a branch-and-bound context.) However, instead of simply imposing these constraints, let us impose the product of these constraints with each w_{ij} and $(1-w_{ij})$, $\forall i \in I_j$, for each $j = 1, \dots, n$, in the spirit of RLT. This yields the following restatement of HCP1.

$$\mathbf{HCP2:} \quad \text{Maximize} \quad \sum_{i=1}^n \sum_{j=1}^c \sum_{k=1}^s a_{ik} w_{ij} z_{jk} \quad (3.1.8a)$$

$$\text{subject to} \quad z_{jk} \sum_{i=1}^n w_{ij} - \sum_{i=1}^n a_{ik} w_{ij} = 0, \quad \forall j, k \quad (3.1.8b)$$

$$\sum_{k=1}^s \gamma_{qk}^j z_{jk} w_{ij} \leq \gamma_{q0}^j w_{ij}, \quad \forall i \in I_j, \forall j, \forall q \quad (3.1.8c)$$

$$\sum_{k=1}^s \gamma_{qk}^j (z_{jk} - z_{jk} w_{ij}) \leq \gamma_{q0}^j (1 - w_{ij}), \quad \forall i \in I_j, \forall j, \forall q \quad (3.1.8d)$$

$$\sum_{j=1}^c w_{ij} = 1, \quad \forall i = 1, \dots, n \quad (3.1.8e)$$

$$w \in W, \quad (3.1.8f)$$

where,

$$W = \{w \text{ binary} : w_{ij} = 0 \text{ for all } (i, j) \in I^-, w_{ij} = 1 \text{ for all } (i, j) \in I^+\} \quad (3.1.9)$$

and where

$$I^+ = \{(i, j) : w_{ij} \text{ has been fixed at 1 (subject to (8e))}\}, \quad (3.1.10a)$$

$$I^- = \{(i, j) : w_{ij} \text{ has been fixed at 0}\}, \quad (3.1.10b)$$

$$I^f = \{(i, j) : w_{ij} \text{ is free (i.e., not fixed)}\}. \quad (3.1.10c)$$

Note that,

$$I_j = \{i \in \{1, \dots, n\} : (i, j) \in I^+ \cup I^f\} \equiv \{i \in \{1, \dots, n\} : (i, j) \notin I^-\}. \quad (3.1.11a)$$

Also, for each $i \in \{1, \dots, n\}$, define $J_i \subseteq \{1, \dots, c\}$ as the set of assignable clusters for data point i , *i.e.*,

$$J_i \equiv \{j \in \{1, \dots, c\} : (i, j) \in I^+ \cup I^f\} \equiv \{j \in \{1, \dots, c\} : (i, j) \notin I^-\}. \quad (3.1.11b)$$

Hence, whenever $I^- = \emptyset$ (*e.g.* to initialize the algorithm), we have $J_i \equiv \{1, \dots, c\}$, $\forall i = 1, \dots, n$.

There are two other classes of constraints that we can add to (3.1.8a) - (3.1.8f) in order to tighten its representation. The first is based on the valid restrictions

$$1 \leq \sum_{i=1}^n w_{ij} \leq n - c + 1, \quad \forall j = 1, \dots, c, \quad (3.1.12a)$$

which asserts that each cluster should be assigned at least one point, and so, each cluster contains at most $n - c + 1$ points due to hard clustering. Furthermore, constraints (3.1.12a) can also be multiplied by z_{jk} , $\forall k$, for each j , in order to generate the following RLT constraints

$$z_{jk} \leq \sum_{i=1}^n w_{ij} z_{jk} \leq (n - c + 1) z_{jk}, \quad \forall (j, k). \quad (3.1.12b)$$

Remark 3.1.1. Note that the right-hand inequalities in (3.1.12a) itself, being implied by (3.1.8e, f) and the left-hand inequalities in (3.1.12a), can be omitted. While the right-hand inequalities in (3.1.12b) might be useful, their worth is questionable. Hence, these inequalities, as well as the utility of other RLT constraints (including those in (3.1.12a, b)) and related modeling strategies were empirically investigated to ascertain their merit, before proposing a final model. Computational results indicated that these inequalities did at least marginally improve the algorithmic convergence (refer Table 3.1.5 for relevant results). Let us refer to HCP2 enhanced by the additional valid inequalities (3.1.12a) and (3.1.12b) as **HCP3**.

Remark 3.1.2. A strong factor that can potentially weaken the relaxation of HCP2 and contribute towards its difficulty in solving via a branch-and-bound approach is the *symmetry* in the problem structure. Note that for any given solution, alternative

equivalent solutions could be obtained by simply re-indexing each cluster composition. To circumvent this difficulty, we propose two alternative sets of hierarchical constraints that could be used to defeat the symmetry (see Sherali and Smith, 2001, for a general discussion on this subject).

Symmetry Strategy 3.1.1.

Impose the following constraints:

$$w_{11} = 1, \quad w_{1j} = 0, \quad \forall j = 2, \dots, c \quad (3.1.13a)$$

$$\sum_{i=1}^n w_{ij} \geq \sum_{i=1}^n w_{i, j+1}, \quad \forall j = 2, \dots, c-1. \quad (3.1.13b)$$

Note that we can arbitrarily assign some point, say point $i = 1$ as in (3.1.13a), to the first cluster. For the remaining clusters, to impart some distinctive identity to these sets, we can require that the indexing be performed in nonincreasing order of their size. This is represented by (3.1.13b). Of course, whenever a solution includes clusters having common sizes, we could still produce alternative equivalent solutions by re-indexing. However, (3.1.13b) does curtail this phenomenon.

Symmetry Strategy 3.1.2.

Initialization: Put the counter $r = 1$. Find a point $p_1 \in \underset{i=1, \dots, n}{\operatorname{arg\,lexmin}} \{a_{i1}, a_{i2}, \dots, a_{is}\}$.

Step 1. If $r = c - 1$, go to Step 3. Else proceed to Step 2.

Step 2. Find a point $p_{r+1} \in \underset{\substack{i=1, \dots, n \\ i \neq p_1, \dots, p_r}}{\operatorname{arg\,max}} \left\{ \underset{t=1, \dots, r}{\operatorname{minimum}} \left\| a_i - a_{p_t} \right\|^2 \right\}$.

Increment r by 1, and return to Step 1.

Step 3. Impose the constraints

$$\left\{ w_{p_r j} = 0 \text{ for } j = r+1, \dots, c \right\}, \text{ for each } r = 1, \dots, c-1. \quad (3.1.14)$$

Note that the assertion (3.1.14) is valid because we can restrict each of the identified points p_r to belong to one of the first r clusters, for each $r = 1, \dots, c-1$. By having selected a dispersed set of points following the process in Step 2 above (given the ‘‘corner’’ point p_1 selected at the initialization step), we enhance the likelihood that these

points also turn out to belong to different clusters, thereby imparting a specific identity to each cluster. As before, this tends to eliminate the symmetry effect, although not completely. In our computational experiments, we test the relative merits of these two symmetry-defeating strategies.

The augmented problem HCP2 using (3.1.12a, b) along with (3.1.13a, b) or (3.1.14) can be restated as follows, where we have substituted

$$y_{ijk} = w_{ij} z_{jk}, \quad \forall i, j, k \quad (3.1.15)$$

in the spirit of RLT, recognizing Proposition 3.1.1 as given below.

$$\mathbf{HCP4:} \quad \text{Maximize} \quad \sum_{i \in I_j} \sum_{j \in J_i} \sum_{k=1}^s a_{ik} y_{ijk} \quad (3.1.16a)$$

$$\text{subject to} \quad \sum_{i \in I_j} y_{ijk} - \sum_{i \in I_j} a_{ik} w_{ij} = 0, \quad \forall j, k \quad (3.1.16b)$$

$$\sum_{k=1}^s \gamma_{qk}^j y_{ijk} \leq \gamma_{q0}^j w_{ij}, \quad \forall i \in I_j, \forall j, \forall q \quad (3.1.16c)$$

$$\sum_{k=1}^s \gamma_{qk}^j (z_{jk} - y_{ijk}) \leq \gamma_{q0}^j (1 - w_{ij}), \quad \forall i \in I_j, \forall j, \forall q \quad (3.1.16d)$$

$$\sum_{j \in J_i} w_{ij} = 1, \quad \forall i = 1, \dots, n \quad (3.1.16e)$$

$$\sum_{i \in I_j} w_{ij} \geq 1, \quad \forall j = 1, \dots, c \quad (3.1.16f)$$

$$z_{jk} \leq \sum_{i \in I_j} y_{ijk} \leq (n - c + 1) z_{jk}, \quad \forall j, k \quad (3.1.16g)$$

$$\text{Constraints (3.1.13) or (3.1.14)} \quad (3.1.16h)$$

$$w \in \mathcal{W}. \quad (3.1.16i)$$

Proposition 3.1.1. For any feasible solution to (3.1.16), we have that (3.1.15) holds true. Hence, (3.1.16a) – (3.1.16i) is an equivalent linear 0-1 mixed integer programming (MIP) representation of HCP.

Proof: For any (i, j) , suppose that $w_{ij} = 0$. Since $\bar{H}(I_j)$ (as defined in (3.1.7)) is a bounded set, its homogeneous system has a unique solution given by the 0-vector. Hence,

by (3.1.16c), we have that $y_{ijk} \equiv 0, \forall k$, and therefore, (3.1.15) holds true in this case. Similarly, if $w_{ij} = 1$, for any (i, j) , then (3.1.16d) implies that $(z_{jk} - y_{ijk}) = 0, \forall k$, or that (3.1.15) again holds true. This completes the proof. \square

We can now design a branch-and-bound algorithm to solve HCP4 based on the following specialized features, as opposed to using default strategies of a standard MIP solver such as CPLEX-MIP 8.1.0 for this purpose.

- (a) Upper bounds can be computed by using the LP relaxation to (3.1.16a) - (3.1.16i). Note that in formulating (3.1.16a) - (3.1.16i), given (3.1.10a) - (3.1.10c), for any partial solution corresponding to a node subproblem, we redefine I_j, J_i , and $\bar{H}(I_j)$ as in (3.1.11a), (3.1.11b), and (3.1.7a) - (3.1.7c) respectively, and use this to reconstruct the model representation, including the derivation of (3.1.16c) and (3.1.16d).
- (b) Heuristic solutions can be derived at each node based on a rounding scheme applied to the LP solution. Specifically, denoting \bar{w} as part of the LP relaxation solution obtained for any node subproblem, if \bar{w} is binary-valued, then by Proposition 3.1.1, the LP solution is optimal for the node subproblem and directly provides a feasible solution for HCP4 (as well as HCP). We can therefore fathom this node and update the incumbent solution, if necessary. Otherwise, we can round the \bar{w} solution to the nearest binary solution subject to (3.1.16e) (also subsequently ensuring (3.4.16f), *i.e.*, each cluster inherits at least one assignable point). Here, for each data point u , we determine $\bar{w}_{uv} = \max \{ \bar{w}_{uj} : j \in J_u \}$, with ties broken by selecting a cluster having the smallest value for $|I_j|$ and we assign $\bar{w}_{uv} = 1$ and $\bar{w}_{uj} = 0, \forall j \in J_u, j \neq v$. (A more comprehensive tie-breaking rule would be to evaluate the objective function (given by (3.1.1a)) corresponding to all possible alternative rounded solutions, and pick the best one among them. However, this would lead to a considerably greater computational effort at each node, and was therefore not implemented in our computations.) Using this resulting binary \bar{w} solution, we then compute the corresponding z -values using (3.1.2b), and hence obtain a feasible solution for HCP,

which can be used to possibly update the incumbent solution for HCP4, upon invoking (3.1.15). (For large-scale problems, the overall procedure could be terminated after applying such an LP based heuristic method at node-zero itself, or by using some limited branching scheme in order to prescribe a heuristic solution to the problem.)

- (c) To select a branching variable, we compute the total absolute discrepancy in the linearized objective terms in (3.1.16a) relative to the nonlinear product terms these represent according to (3.1.15), as given by

$$\theta_{ij} = \sum_{k=1}^s \left| a_{ik} (\bar{y}_{ijk} - \bar{w}_{ij} \bar{z}_{jk}) \right|, \quad \forall (i, j) \quad (3.1.17a)$$

where $(\bar{w}, \bar{z}, \bar{y})$ solves the LP relaxation $\overline{\mathbf{HCP4}}$ to HCP4 given by (3.1.16). Then, in one partitioning strategy, we branch on the dichotomy that $w_{uv} = 0$ or 1, where

$$(u, v) \in \arg \max_{(i, j) \in I^f} \{ \theta_{ij} \}. \quad (3.1.17b)$$

Naturally, on the branch $w_{uv} = 1$, we also set $w_{uj} = 0, \forall j \neq v$, and on the branch $w_{uv} = 0$, the sets I_v and J_u would now not include the respective indices u and v , and $\overline{H}(I_v)$ would accordingly exclude a_u in the convex hull computation or its approximation.

Remark 3.1.3. Exploiting the structure of the inherent generalized upper bounding (GUB) constraints (16e), we also explore an alternative *specially ordered set* (SOS) branching strategy. In this scheme, defining θ_{ij} as in (3.1.17a), and denoting $\theta_i \equiv \sum_{j \in J_i} \theta_{ij}$,

we compute $u \in \arg \max_i \{ \theta_i \}$. (Note that by Proposition 1, if $\theta_u > 0$ then the vector

$(\bar{w}_{uj}, j \in J_u)$ is not binary-valued; else, we simply select u such that the total fractionality of the components of this latter vector is a maximum.) We now partition J_u into two

children nonempty sets, J_{u1} and J_{u2} , as follows, where we then construct two subproblem nodes in the branch-and-bound tree corresponding to the respective imposed

branching restrictions $\sum_{j \in J_{u1}} w_{uj} = 1$ and $\sum_{j \in J_{u2}} w_{uj} = 1$. To determine this partition J_{u1} and

J_{u_2} of J_u , we first arrange the θ_{u_j} values, $j \in J_u$, in nonincreasing order. Let this sorted set be $\{\theta_{u_{j_1}}, \theta_{u_{j_2}}, \dots, \theta_{u_{j_l}}\}$, where $l = |J_u| \geq 2$. Now, define $p \geq 1$ to be the smallest integer such that $\sum_{r=1}^p \theta_{u_{j_r}} \geq \theta_u / 2$. Note that $1 \leq p < l$, by virtue of the sorted list. Accordingly, we then define $J_{u_1} = \{j_1, \dots, j_p\}$ and $J_{u_2} = \{j_{p+1}, \dots, j_l\}$. Likewise, for each of these children nodes, the sets I_j would then be revised accordingly.

(d) Using the LP dual solution, a reduced cost cut based on requiring the objective function to be greater than or equal to the incumbent lower bound can be constructed in terms of w , by surrogating and dualizing all constraints except for (3.1.16e) and $0 \leq w_{ij} \leq 1, \forall (i, j)$. Logical tests can be conducted on this in order to possibly fix some w -variables at 0 or 1 values, and thereby tighten the relaxation further (at least for the children nodes, if not for resolving the LP at the same node).

Remark 3.1.4. In our implementation of the branch-and-bound algorithm that includes features (a)-(d) outlined above, a depth-first strategy was adopted to develop the enumeration tree. For the purpose of obtaining tight lower bounds, and to possibly update incumbent solutions, a rounding heuristic as proposed in (b) was employed at every node of the branch-and-bound tree. Also, based on our computational experiments, the SOS branching was determined to be the best branching strategy for larger problem instances (refer Table 3.1.6 for pertinent results), and was therefore taken as the primary branching scheme. Here, in the depth-first framework, we branched first along the J_{u_1} side to explore the corresponding child node. (Note that for problems having a large number of clusters, to find good feasible solutions more quickly, we could first explore the child node along the side having the smaller $|J_{u_1}|$ or $|J_{u_2}|$ value, breaking ties by choosing J_{u_1} .) The overall branch-and-bound algorithm was implemented in C++, and the commercial software CPLEX 8.1.0 was invoked for the purpose of solving the LP relaxations at each node. Furthermore, the optimal basis for the parent node was used as an advanced-start basis for the two children nodes, thereby enabling a quicker update for

the solutions to each of the node subproblems. Note that the CPLEX 8.1.0 command options facilitate these implementations.

3.1.2. Computational Results

Throughout this section, we will use the following terminology:

- UB_0 : Optimal objective function value of $\overline{\text{HCP4}}$ at node zero.
- LB_0 : Objective function value of the heuristic solution to HCP4 found at node zero.
- Z_0^* : Objective function value of HCP corresponding to the heuristic solution found at node zero.
- LB^* : Optimal objective function value of HCP4.
- Z^* : Optimal objective function value of HCP, evaluated at the optimal solution to HCP4.
- $Z_{k\text{-means}}^*$: Best objective function value obtained via the k -means algorithm.
- CPU^* : CPU time required to determine a global optimum for HCP4 via the proposed branch-and-bound algorithm.
- $\text{CPU}_{k\text{-means}}$: CPU time required for the k -means algorithm.
- CPU_0 : CPU time required to determine a heuristic solution at node zero via the solution to $\overline{\text{HCP4}}$.

First, for the purpose of illustration, consider the following clustering problem having ten data points to be divided into three clusters, where each data point is assigned two attributes, (*i.e.*, $s = 2$). Table 3.1.1 provides the input data, for this example problem. Using the above data, the LP relaxation of Problem HCP4 was solved and the optimal solution value at node zero (UB_0) was found to be 43156. Applying the rounding heuristic described in Section 3.1.2, we obtained an incumbent value (LB_0) of 22434.68. Hence, the gap ratio at node zero is given by $UB_0/LB_0 = 1.9236$. Actually, since $LB^* = 27884.5$, as determined below, the true LP-IP gap at node zero is $UB_0/LB^* = 1.5476$. Also, the objective function value for the minimization problem HCP (computed by substituting the (w, z) parts of the heuristically determined node zero incumbent

solution into (3.1.1a)) was found to be $Z_0^* = 18484.35$. Next, using the SOS branching strategy designed in (3.1.17a) and Remark 3.1.3, from the LP solution at node zero we

Points Attributes	1	2	3	4	5	6	7	8	9	10
1	-57	54	46	8	-36	-22	34	74	-6	21
2	28	-65	79	111	52	-76	129	6	-41	45

Table 3.1.1: Attributes for the ten data points in \mathbb{R}^2 for the illustrative example.

get $\theta_3 = \arg \max_i \{\theta_i\} = 1174$, and the corresponding $\theta_{3,j}$ values are given by $\{766, 300.44, 107.56\}$. Hence, $p = 1$, and we can now formulate the subnode problems by splitting $J_3 = \{1, 2, 3\}$ into two subsets, $J_{3,1} = \{1\}$ and $J_{3,2} = \{3, 2\}$, and respectively imposing the constraints $w_{31} = 1$ and $w_{32} + w_{33} = 1$ for the corresponding subnode problems. Employing a depth-first strategy, Figure 3.1.1 depicts the (partial) branch-and-bound tree generated up to node eight, and illustrates the SOS branching computations. Continuing to completion, the optimal objective function value for Problem HCP4 was found to be $LB^* = 27884.5$, and the corresponding $Z^* = 15805.25$. A total of 27 nodes were enumerated in determining this optimal solution. For the purpose of comparison, the above problem was solved via the k -means algorithm, and the optimal objective function value for Problem HCP was found to be $Z_{k\text{-means}}^* = 34404.857$. (Note that, since the k -means algorithm requires the cluster centers as an input, its performance can be significantly enhanced by a good estimate of the initial cluster centers. In our computations, five randomly generated cluster center configurations were examined, and the best resulting solution was used for the above comparison.) Considering the ratio $Z_{k\text{-means}}^*/Z^* = 2.176$, it is evident that the performance of the optimization algorithm is considerably superior to the k -means algorithm. Indeed, even from the ratio $Z_{k\text{-means}}^*/Z_0^* = 1.86$, we see that the feasible solution obtained from node zero of the optimization problem itself is significantly better than the k -means solution. Figure 3.1.2

displays the optimal clustering patterns obtained via the proposed optimization algorithm (solid lines) and via the k -means algorithm (dashed lines).

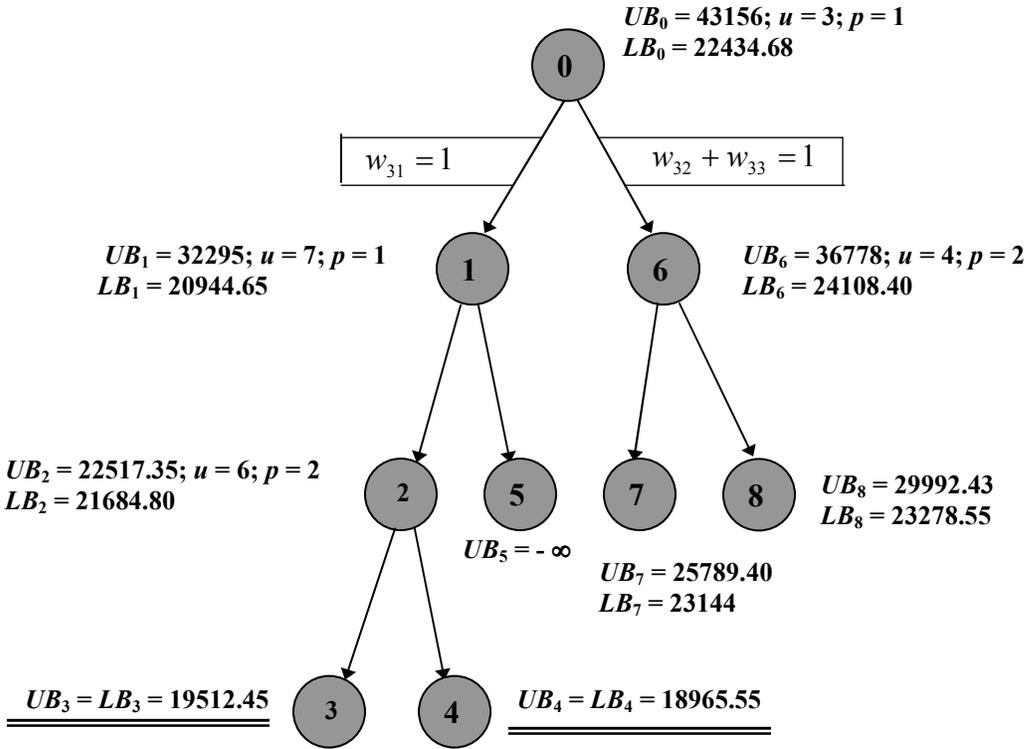
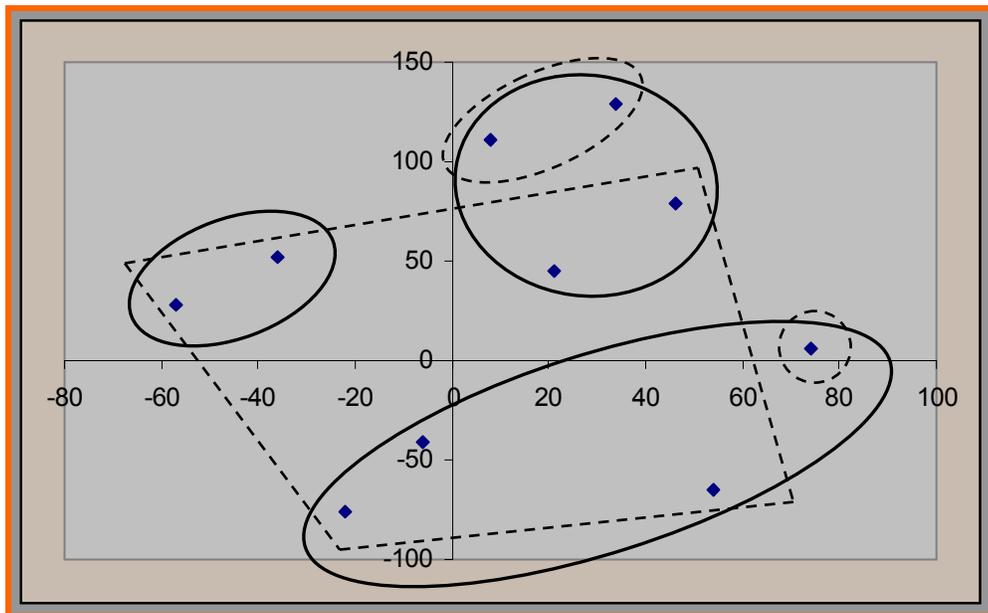


Figure 3.1.1: Branch-and-bound tree illustrating the SOS branching strategy.



Solid line: Optimal clustering; Dashed line: k -means clustering.

Figure 3.1.2: Clustering patterns obtained by solving Problem HCP4 via the proposed algorithm and by the k -means algorithm.

Next, we used the following standard data sets given in Späth (1980) to test our proposed methodology:

1. **Data Set 1.** This is a set of Cartesian coordinates for 22 German towns, which yields a clustering problem having 22 points in a two-dimensional space.
2. **Data Set 2.** This is a set of Cartesian coordinates for 59 German towns, which yields a clustering problem having 59 points in a two-dimensional space.
3. **Data Set 3.** This pertains to 89 postal zones in Germany, where each zone has three attributes, namely, surface area (measured in square kilometers), population, and the density of population. This yields a clustering problem having 89 points in a three-dimensional space.
4. **Data Set 4.** This is also based on the 89 postal zones of Data Set 3, but considers four attributes, namely, the number of self-employed people, civil servants, clerks, and manual workers. This yields a clustering problem having 89 points in a four-dimensional space.

The above-mentioned data sets were used to provide the input data for Problem HCP4, and the performance of the proposed approach was compared with the k -means algorithm. Tables 3.1.2 and 3.1.3 display the results obtained for the cases of three and five cluster centers, respectively.

Note that, on an average, for the case of three-cluster centers, the k -means algorithm required 31.65% of the CPU time consumed by the proposed approach, but the quality of the solution (with respect to the HCP objective values) was significantly inferior being worse (greater) by a factor of 4.862. Indeed, the heuristic at node zero itself uniformly dominated the k -means algorithm, determining an objective function value that is 11.56% better (lesser) on an average, while consuming only 12.05% of CPU time. Similarly, in the case of five-cluster centers, the k -means algorithm required 22.76% of the time taken by the proposed exact approach, but produced a solution that was greater by a factor of 4.34. Again, the solution obtained by our method at node zero itself dominated the k -means solution, improving it on an average by 15.66%, while consuming only 11.45% of the CPU time required. Furthermore, the UB_0/LB^* column in Tables 4.1.2 and 4.1.3 records the LP-IP gap having an average value of 1.90 and 1.66,

for three and five cluster centers, respectively. Also, a comparison of this ratio with UB_0/LB_0 reflects the extent of improvement in the final objective value attained (LB^*) versus the node zero incumbent value LB_0 .

Parameters Data Sets	$\frac{UB_0}{LB_0}$	$\frac{UB_0}{LB^*}$	$\frac{Z_0^*}{Z^*}$	$\frac{Z_{k\text{-means}}^*}{Z^*}$	CPU* (s)	CPU _{<i>k</i>-means} (s)	$\frac{CPU^*}{CPU_{k\text{-means}}}$	$\frac{CPU_0}{CPU_{k\text{-means}}}$
1	4.67	1.44	3.15	3.41	0.20	0.14	1.428	0.052
2	6.15	2.02	5.43	7.688	0.40	0.20	2.0	0.074
3	6.72	2.30	4.55	4.73	1.28	0.30	4.267	0.195
4	5.09	1.87	4.08	3.62	0.90	0.24	3.75	0.161
Averages	5.65	1.90	4.30	4.862	0.695	0.22	2.861	0.1205

Table 3.1.2: Relative performance of the proposed optimization approach versus the *k*-means algorithm, measured in terms of different parameters, for three cluster centers.

Parameters Data Sets	$\frac{UB_0}{LB_0}$	$\frac{UB_0}{LB^*}$	$\frac{Z_0^*}{Z^*}$	$\frac{Z_{k\text{-means}}^*}{Z^*}$	CPU* (s)	CPU _{<i>k</i>-means} (s)	$\frac{CPU^*}{CPU_{k\text{-means}}}$	$\frac{CPU_0}{CPU_{k\text{-means}}}$
1	3.73	1.12	1.55	1.80	0.355	0.14	2.535	0.089
2	5.85	2.09	6.15	4.57	0.59	0.22	2.68	0.10
3	7.71	2.35	4.50	6.22	2.10	0.28	7.5	0.16
4	4.20	1.08	2.44	4.78	1.70	0.44	3.86	0.109
Averages	5.37	1.66	3.66	4.34	1.186	0.27	4.14	0.1145

Table 3.1.3: Relative performance of the proposed optimization approach versus the *k*-means algorithm, measured in terms of different parameters, for five cluster centers.

Note that the performance of the branch-and-bound algorithm is influenced by three factors: First, choosing the best model formulation among variations of Problem HCP4; second, selecting an appropriate strategy to ameliorate the effects of symmetry and third, implementing a judicious branching mechanism. Hence, prior to evaluating the robustness of the proposed approach on relatively larger problem instances, several computational tests were performed on some sample problems to ascertain the effects of

the foregoing three features and thereby, compose a suitable algorithmic approach. The results of these various experimental runs are recorded in Tables 3.1.4 through 3.1.7.

To begin with, Table 3.1.4 displays the comparative results obtained for the two proposed symmetry-defeating strategies. For this purpose, four randomly generated sample problems involving three clusters, and having the number of data points and attributes (dimension) as indicated in Table 3.1.4 were solved, with the model variations being HCP4 without (3.1.16f, g), but including either (3.1.13a, b), or (3.1.14), or neither. Also, in Table 3.1.4, the values in the parentheses recorded for each cell indicate the two-tuple $(UB_0/LB_0, CPU^*)$, *i.e.*, the gap ratio at node zero and the total CPU time. Based on the results obtained, note that Problem HCP4 with (3.1.14) included obtains an average UB_0/LB_0 gap ratio of 10.1 (which is marginally worse than that for HCP4 with (3.1.13a, b) included) but consumes the least amount of CPU time. Evidently, attempting to identify distinct clusters based on the allocation of a set of most dispersed points serves to provide an effective symmetry-defeating strategy, and is the one we propose to implement henceforth. Note that ignoring the effects of symmetry takes 19.36% greater effort, and is clearly not advisable.

Data Set Problem Type	1	2	3	4	Averages
	(250, 4)	(250, 6)	(500, 4)	(500, 6)	
HCP4 without (16f, g) and only (13) included	(9.05, 144.11)	(8.74, 288.42)	(6.87, 312.62)	(14.67, 421.6)	(9.83, 280.68)
HCP4 without (16f, g) and only (14) included	(8.75, 128.40)	(8.74, 269.10)	(8.08, 279.05)	(14.88, 387.2)	(10.1, 271.05)
HCP4 without (16f, g) and neither (13, 14)	(9.24, 144.45)	(11.67, 298.4)	(12.43, 344.5)	(15.45, 474.5)	(12.19, 315.4)

Table 3.1.4: Variations of Problem HCP4 to test the effectiveness of the different symmetry defeating strategies, measured in terms of gap ratio at node zero and the CPU time.

Next, the efficacy of including the constraints (3.1.16f) and (3.1.16g) was tested. Here, the performance of HCP4 measured according to the gap ratio at node zero, both with respect to the node zero incumbent value LB_0 and the optimal solution value LB^* , as well as the CPU time consumed was examined for the three cases corresponding to

Data Set	1	2	3	4	Averages
Problem Type	(250, 4)	(250, 6)	(500, 4)	(500, 6)	
HCP4 with only (16f) included	(9.24, 4.36, 138.76)	(10.3, 5.75, 266.0)	(8.55, 4.08, 332.62)	(15.45, 7.88, 442.6)	(10.9, 5.51, 295.0)
HCP4 with both (16f, g) included	(7.44, 2.84, 116.81)	(8.74, 3.90, 269.10)	(7.90, 3.37, 258.2)	(12.33, 4.97, 369.3)	(9.1, 3.77, 253.35)
HCP4 with neither (16f, g) included	(9.24, 4.36, 144.45)	(11.67, 6.44, 298.4)	(12.43, 8.15, 344.5)	(15.45, 8.03, 474.5)	(12.19, 6.74, 315.4)

Table 3.1.5: Variations of Problem HCP4 to test the effectiveness of the different bounding constraints measured in terms of the gap ratio at node zero and the CPU time.

using only (3.1.16f), using (3.1.16f) and (3.1.16g), and using neither. (Partial results for the most latter case are given in Table 3.1.4.) Table 3.1.5 displays the results obtained for these various formulations as a three-tuple given by $(UB_0/LB_0, UB_0/LB^*, CPU^*)$, and provides a measure of the quality of the LP relaxation. Observe that including constraints (3.1.16f) and (3.1.16g) leads to a decrease in the node zero gap ratio, with respect to both the node zero incumbent value LB_0 and the optimal solution value LB^* , as well as reduces the overall CPU effort. In comparison with the case wherein neither of (3.1.16f, g) was included, these values decreased by 25.34%, 44.06%, and 19.67%, respectively. Likewise, there is a decrement of 16.51%, 31.57%, and 14.1% in these respective values, in comparison with the case when only (3.1.16f) is present. Hence, we recommend incorporating both (3.1.16f) and (3.1.16g) in the model formulation.

The third test performed was to evaluate the two alternative branching strategies proposed in Section 3.1.2, based on imposing the dichotomy $w_{uv} = 0$ or 1 on a single variable as identified by (3.1.17a, b), or based on the SOS partitioning scheme as designated in Remark 3.1.3. The performance of these branching strategies is reported in Table 3.1.6 in terms of the two-tuple: (number of nodes enumerated in the branch-and-bound tree, CPU time taken to determine an optimal solution). The results obtained appear to indicate that the partitioning scheme given by (3.1.17a, b) is more efficient for smaller sized problems, but the SOS branching strategy begins to dominate as the size of the problem increases.

Although the SOS branching strategy uniformly dominates in terms of the number of nodes enumerated, the effort required at each node is greater and hence, for the relatively smaller sized problems, more CPU time is taken to determine an optimal solution. However, as the problem size increases, the number of nodes enumerated is considerably larger for the branching strategy (3.1.17a, b) as compared with the SOS branching method to the extent that the SOS branching scheme begins to dominate. Naturally, the solution of larger sized problems more effectively is of greater concern, and so we recommend the use of the SOS branching strategy.

Data Set Branching Rule	1	2	3	4	Averages
	(250, 4)	(250, 6)	(500, 4)	(500, 6)	
HCP4 with branching strategy (17a, b)	(1068, 113.2)	(1454, 256.0)	(2455, 340.45)	(3580, 462.4)	(2139, 293.0)
HCP4 with SOS branching strategy	(866, 116.81)	(974, 269.1)	(1375, 258.2)	(2421, 369.3)	(1409, 271.0)

Table 3.1.6: Performance of the different branching strategies, measured in terms of the number of nodes enumerated and the CPU time.

Finally, for the purpose of comparison, a computational study was performed to test the efficacy of solving the enhanced model formulation HCP4 directly by the commercial software CPLEX-MIP 8.1.0 using its default settings. Furthermore, as a point of interest, the commercial global optimizer BARON (refer Sahinidis 1996, 1999-2000) was utilized to directly solve the original model HCP augmented by the symmetry-defeating constraints (3.1.14). Let us denote the best objective function value obtained by solving CP4.1 with (3.1.14) via BARON as Z_{BARON}^* . Table 3.1.7 displays the CPU times obtained for each of these cases, and the ratios of the final objective function values. Comparing the results displayed in Table 3.1.7 with those in Table 3.1.6, it can be seen that both HCP4 solved directly by CPLEX-MIP 8.1.0 as well as CP4.1 with (3.1.14) solved via BARON consume a significantly greater CPU time for larger problem instances, as compared with using the proposed branch-and-bound algorithm. Indeed, in those instances where solving the nonlinear program CP4.1 with (3.1.14) via BARON dominated in terms of CPU time, it terminated at a significantly inferior local optimal

solution (as recorded by the Z_{BARON}^*/Z^* values). Assimilating the information given in Tables 3.1.4 through 3.1.7, we conclude that Problem HCP4 including the constraints (3.1.16f, g) along with the symmetry-defeating mechanism given by (3.1.14), and solved via the proposed branch-and-bound algorithm utilizing the SOS branching scheme affords the most viable composition of the tested strategies for solving relatively large instances of the hard clustering problem.

Data Set Problem Type	1	2	3	4	Averages
	(250, 4)	(250, 6)	(500, 4)	(500, 6)	
HCP4 via CPLEX-MIP 8.1.0 default settings	263.44	360.83	577.0	711.25	478.13
CP1 with (14) via the BARON global optimizer	388.45	344.67	649.07	697.7	468.61
$\frac{Z_{\text{BARON}}^*}{Z^*}$	1.0	2.43	1.79	3.45	2.16

Table 3.1.7: The performance of HCP4 via the default strategies of CPLEX-MIP 8.1.0 and HCP3 via BARON, measured in terms of CPU time.

To reinforce this and to establish the robustness of the proposed approach, we solved several additional problems of larger sizes, and also compared the results obtained with those produced by the popular k -means algorithm. The number of data points in these test instances was varied from 250 to 1000 in steps of 250, and the dimension of the space was varied from two to eight, in steps of two, thereby leading to a total of $4 \times 4 = 16$ test problems, with the smallest data set having 250 points in a two-dimensional space, and the largest problem having 1000 points in an eight-dimensional space. The number of clusters (c) for each case was taken to be either three (Table 3.1.8) or five (Table 3.1.9).

From the results displayed in Tables 3.1.8 and 3.1.9, note that the k -means algorithm requires a significantly lesser CPU time as compared with the proposed exact approach, but the best solution produced by the k -means algorithm is also substantially inferior. However, the node zero heuristic solution produced by the proposed approach uniformly dominates the k -means solution with respect to both quality and effort in most of the problem instances, with the exceptions being shaded in the rows of Tables 3.1.8 and 3.1.9. On an average, to obtain a feasible solution to Problem HCP based on the node

zero analysis alone, the CPU time required is on an average 17.2% lesser than for the k -means algorithm, yet the quality of the solution is 13.3% better in terms of the objective function value for the three cluster center case. A similar result holds true for the case of five cluster centers. Using a more sophisticated heuristic than the one advocated in Remark 3.1.4, or improving this solution by appending some steps of a suitable meta-heuristic approach such as the genetic algorithm or simulated annealing, might lead to a

Parameters Data Sets	$\frac{UB_0}{LB_0}$	$\frac{UB_0}{LB^*}$	$\frac{Z_0}{Z^*}$	$\frac{Z_{k\text{-means}}^*}{Z^*}$	CPU* (s)	CPU _{k-means} (s)	$\frac{\text{CPU}_{\text{CP1.4}}}{\text{CPU}_{k\text{-means}}}$	$\frac{\text{CPU}_0}{\text{CPU}_{k\text{-means}}}$
(250, 2)	6.72	3.17	2.45	3.34	35.411	3.314	10.685	0.635
(500, 2)	7.19	1.85	2.61	3.58	74.554	5.335	13.974	0.504
(750, 2)	25.4	4.44	12.9	9.00	150.30	11.027	13.630	0.978
(1000, 2)	13.1	2.29	4.70	6.62	180.87	17.578	10.289	0.844
(250, 4)	15.8	2.76	5.66	8.02	139.19	4.58	30.392	0.751
(500, 4)	26.5	4.64	11.8	11.07	246.60	8.572	28.768	1.779
(750, 4)	8.89	1.55	3.21	4.45	377.67	10.208	36.997	0.615
(1000, 4)	32.2	5.64	11.4	16.4	708.10	17.578	40.283	0.497
(250, 6)	16.5	3.89	5.88	8.34	384.33	23.801	16.147	0.555
(500, 6)	29.9	5.23	13.2	12.54	466.48	18.547	25.151	1.961
(750, 6)	21.5	3.76	7.64	10.9	485.43	14.602	33.244	0.783
(1000, 6)	7.09	2.24	2.58	3.53	508.38	18.790	27.056	0.634
(250, 8)	31.0	5.43	15.5	11.2	392.33	17.578	22.316	0.932
(500, 8)	4.21	2.73	1.57	2.06	945.16	20.316	46.523	0.589
(750, 8)	32.6	5.71	11.5	10.6	1181.1	32.106	36.787	1.089
(1000, 8)	23.7	4.15	8.40	12.0	1425.3	39.925	35.701	0.603
Averages	18.9	3.71	7.56	8.72	481.33	16.491	26.746	0.859

Table 3.1.8: Results for the proposed approach and the k -means algorithm for large problem instances having three cluster centers.

more effective procedure. Moreover, utilizing a better approximation to the convex hull of data points in the model formulation could lead to a further improvement in the performance of both the exact and heuristic routines. We recommend these investigations for future research.

Parameters Data Sets	$\frac{UB_0}{LB_0}$	$\frac{UB_0}{LB^*}$	$\frac{Z_0^*}{Z^*}$	$\frac{Z_{k\text{-means}}^*}{Z^*}$	CPU* (s)	CPU _{<i>k</i>-means} (s)	$\frac{\text{CPU}_{\text{CP1.4}}}{\text{CPU}_{k\text{-means}}}$	$\frac{\text{CPU}_0}{\text{CPU}_{k\text{-means}}}$
(250, 2)	12.4	2.17	2.29	2.77	28.383	1.793	15.829	0.681
(500, 2)	12.1	3.11	5.33	6.43	73.495	2.887	25.457	0.729
(750, 2)	8.26	1.44	8.51	7.85	127.89	5.967	21.433	1.106
(1000, 2)	1.37	1.23	8.07	9.74	191.35	16.092	11.891	0.574
(250, 4)	2.98	1.52	6.80	8.21	116.81	4.638	25.185	0.722
(500, 4)	3.13	1.54	5.13	4.99	258.20	5.524	46.741	0.955
(750, 4)	14.2	3.48	8.21	9.91	349.13	12.092	28.873	0.661
(1000, 4)	1.21	1.21	11.2	13.6	360.20	12.879	27.968	0.935
(250, 6)	1.71	1.29	3.89	4.70	269.10	10.036	26.813	0.997
(500, 6)	25.2	4.41	8.13	7.40	369.33	7.901	46.744	1.716
(750, 6)	6.83	1.19	7.70	9.29	475.24	10.167	46.743	0.736
(1000, 6)	1.23	1.21	5.21	6.29	491.35	26.092	18.831	0.414
(250, 8)	2.47	1.43	6.79	8.19	313.84	10.993	28.549	0.915
(500, 8)	25.0	4.37	4.71	5.69	812.03	17.373	46.740	0.642
(750, 8)	1.05	1.05	10.8	13.1	1009.8	21.604	46.741	0.614
(1000, 8)	11.6	3.03	8.83	9.45	1257.5	26.905	46.741	0.785
Averages	13.1	2.10	6.97	7.97	406.48	12.058	33.708	0.611

Table 3.1.9: Results for the proposed approach and the *k*-means algorithm for large problem instances having five cluster centers.

3.1.3. Summary, Conclusions, and Extensions for Further Research

In Section 3.1, we addressed the problem of determining a global optimum to the hard clustering problem, where the objective function seeks to minimize the total squared (Euclidean) distance from each data point to the center of the cluster to which it is assigned. A series of enhanced reformulations of this problem were presented, augmented by valid inequalities and RLT-based constraints. A specialized branch-and-bound algorithm was designed for the resulting equivalent 0-1 mixed-integer programming problem. Several computational experiments were performed using standard data sets as well as synthetically generated test cases, to explore the efficacy of including the different proposed model enhancement strategies, as well as to study the effectiveness of

the heuristic scheme implemented at the root node. Furthermore, this performance was compared with the k -means algorithm (see Forgy 1966, McQueen, 1967) that is popularly used in the literature on this topic. The results support the robustness of the proposed approach, and exhibit its superiority over the k -means algorithm (even as a heuristic based on the node zero analysis). In particular, the RLT-enhanced model HCP4 coupled with a valid symmetry-defeating strategy, and solved via the proposed branch-and-bound algorithm using an SOS branching mechanism yielded the best combination of the strategies tested, and is recommended for solving the hard clustering problem. Note that in practice, cluster analysis problems often involve very large data sets, and therefore, good heuristic procedures are essential for handling such problem instances. Our research suggests that designing heuristic methods based on constructs that are borrowed from strong effective exact procedures might be a prudent approach.

Finally, note that the number of cluster centers is introduced as a fixed, external parameter into the optimization model, as opposed to finding an optimal number of clusters, given a certain data set. A decision criterion to determine an optimal number of clusters in hierarchical clustering was advocated by Jung *et al.* (2003). Our work could be extended to accommodate this feature as well. As another possible extension, an alternative idea that one could use to address the issue of symmetry (see Remark 3.1.2), as well as to develop an effective heuristic procedure is as follows.

Define a *cluster vector* v_r , for any index r , to have n components, with the i^{th} component being 1 if the point a_i is assigned to the particular cluster, and 0 otherwise. Let

$$V_r = \{i : (v_r)_i = 1\}. \quad (3.1.18)$$

Given a cluster vector v_r with the associated set of assigned points V_r , the optimal center location z has components given via (3.1.2b) as

$$z_k^* = \sum_{i \in V_r} a_{ik} / |V_r|. \quad (3.1.19)$$

The objective cost term associated with this cluster vector, C_r , is given as follows, using (3.1.3) and (3.1.19).

$$C_r \equiv \sum_{i \in V_r} \sum_{k=1}^s (z_k^* - a_{ik})^2 = \sum_{i \in V_r} \sum_{k=1}^s a_{ik}^2 - \frac{1}{|V_r|} \sum_k \left\{ \sum_{i \in V_r} a_{ik} \right\}^2. \quad (3.1.20)$$

Suppose that we have generated several potential cluster vectors indexed by $r = 1, \dots, R$, based on various covers of the scatter of data points. Then we can solve the following set partitioning problem (**SPP**), where e is a vector of n ones.

$$\text{SPP:} \quad \text{Minimize} \quad \sum_{r=1}^R C_r x_r \quad (3.1.21a)$$

$$\text{subject to} \quad \sum_{r=1}^R v_r x_r = e \quad (3.1.21b)$$

$$x \text{ binary.} \quad (3.1.21c)$$

Note that feasibility of (3.1.21) can be assured by including within it a known partition of the data points into suitable cluster vectors. Furthermore, having solved this, we could try generating additional clusters that might yield a negative reduced cost for the LP relaxation $\overline{\text{SPP}}$ (and hence, perhaps lead to an improved IP solution) by monitoring the following reduced cost expression, as points are added to V_r , where $\bar{\pi}$ is an optimal dual solution vector associated with (3.1.21b).

$$C_r - \bar{\pi} v_r = \sum_{k=1}^s \left\{ \sum_{i \in V_r} a_{ik}^2 - \frac{1}{|V_r|} \left[\sum_{i \in V_r} a_{ik} \right]^2 \right\} - \sum_{i \in V_r} \bar{\pi}_i. \quad (3.1.22)$$

Conceivably, some genetic algorithmic concepts could be applied to generate such advantageous members from the population of cluster vectors. Turning this into an effective heuristic scheme is a topic that is recommended for future research. Note that this algorithmic process could also possibly be converted into an exact algorithm by using branch-and-price concepts, although one would need to contend in this context with a nonconvex objective function, which would be problematic.

3.2. Fuzzy Clustering Problem

Amongst the many areas in which optimization has proven to be an invaluable tool, one notable application is that of cluster analysis. As defined in Section 3.1, clustering is the process of partitioning a set of data points, $i = 1, \dots, n$, into subsets, $j = 1, \dots, c$, called *clusters*, such that some distance measure is minimized (Sultan *et al.*, 2002). Specifically, there are two different types of clustering problems that have been addressed in the literature: the *hard clustering* problem, wherein a data point is to be assigned to exactly one cluster (refer Section 3.1 for details), and the *fuzzy clustering* problem, which, in contrast, addresses the issue of assigning a data point to one or more clusters along with a designation of a membership grade for each assignment that represents the likelihood of the data point belonging to that cluster. (Here, the word fuzzy is derived from fuzzy programming, and reflects the fact that the specific cluster to which a data point belongs is only fuzzily known, and is not deterministic.)

A first attempt to solve the fuzzy clustering problem is credited to Dunn (1973). Subsequently, Bezdek (1981) generalized Dunn's algorithm and developed a more comprehensive iterative procedure, popularly known as the *fuzzy c-means algorithm* (FCMA). Given a set of heuristically prescribed initial cluster centers, the FCMA first computes the membership grade for each data point based on the relative distance measures, and then revises the cluster centers using these resulting membership grades as fixed input quantities. This process is iteratively repeated until no further improvement in the objective function is obtained. However, it has been observed that FCMA often produces local minima and/or suboptimal clustering of the given data. Consequently, several recent algorithms that have appeared in literature to solve the fuzzy clustering problem are essentially modifications and improvements of the FCMA (refer Kamel and Selim, 1994). Furthermore, although some of these iterative procedures are guaranteed to converge to optimality under certain assumptions, this convergence can be slow in practice. Other issues such as the validity of the clusters determined by the FCMA (Roubens, 1982, Zahid *et al.*, 1999, Windham, 1982), the geometric shape of the clusters produced (Windham, 1983), and a demonstration that the FCMA can at best produce only local optima (Ismail and Selim, 1986), are also addressed in the literature. As an aside,

note that the FCMA produces cluster regions that are always spherical in shape. Gustafson and Kessel (1979) found that replacing the traditional Euclidean distance objective function criterion by another measure formulated from a symmetric, positive semidefinite matrix, yielded elliptical clusters when solved via a modified version of the FCMA. Gath and Geva (1989) further generalized this concept by taking into account the size and density of the clusters as well.

A comprehensive survey of fuzzy cluster analysis that is specifically aimed at pattern recognition problems is presented by Baraldi and Blonda (1999a, b) and the use of evolutionary algorithms for fuzzy clustering is discussed by Klawon and Keller (1998). However, despite this notable literature dedicated to solving the fuzzy clustering problem, there exist only a limited number of global optimization procedures, such as the algorithms designed via fuzzy set theory as proposed by Ruspini (1973) and Guoyao (1998). This motivates us to consider alternative effective and robust global optimization approaches for solving the fuzzy clustering problem.

In this research effort, similar to Section 3.1, we again apply the RLT to develop an effective global optimization algorithm for solving the fuzzy clustering problem. However, this approach is completely different from that for the hard clustering case because of the modified structure of the present problem. The remainder of Section 3.2 is organized as follows. The fuzzy clustering problem is formulated as a nonlinear program in Section 3.2.1, and a tight linear programming relaxation is derived as prescribed via the RLT methodology. Accordingly, the reformulated problem is then embedded in a specialized branch-and-bound algorithm along with a branching rule that ensures global convergence, in the spirit of Sherali and Tuncbilek (1992). Section 3.2.2 presents computational results using certain standard test problems from the literature as well as using larger synthetically generated data sets, and explores the performance of different formulations. Finally, Section 3.2.3 concludes this chapter with a summary and a discussion on further avenues for research in this area.

3.2.1. Modeling and Reformulation

The fuzzy clustering problem can be defined as follows. Given a set of n data points, each having some s attributes, we are required to assign each of these points to

one or more of some c clusters (where c is given). In this process, we are also required to specify for each assignment a membership grade that represents the likelihood of the data point belonging to that cluster. The objective criterion is to minimize the total weighted squared Euclidean distances of the data points from the centroids of the assigned clusters.

Mathematically, this fuzzy clustering problem (**FCP**) can be stated as follows.

$$\mathbf{FCP:} \quad \text{Minimize} \quad \sum_{i=1}^n \sum_{j=1}^c w_{ij}^2 \|a_i - z_j\|^2 \quad (3.2.1a)$$

$$\text{subject to} \quad \sum_{j=1}^c w_{ij} = 1, \quad \forall i = 1, \dots, n \quad (3.2.1b)$$

$$w_{ij} \geq 0, \quad \forall (i, j), \quad (3.2.1c)$$

where, as aforementioned, $a_i \equiv (a_{ik}, k = 1, \dots, s)^T$ is the location descriptor for the data point i , $z_j \equiv (z_{jk}, k = 1, \dots, s)^T$ is the centroid of the to-be-determined cluster j , w_{ij} is the membership grade associated with a data point i when assigned to a cluster j , and the norm $\| \cdot \|$ in (3.2.1a) represents the Euclidean distance between the two points in its argument in the s -dimensional space under consideration.

Note that, in general, the objective function for the fuzzy clustering problem is sometimes expressed as $\sum_{i=1}^n \sum_{j=1}^c w_{ij}^m \|a_i - z_j\|^2$, where m represents the *degree* of fuzziness, with the notion that m is increased as the desired extent of fuzziness in the problem increases. Given a data set, the choice of m , also called the *fuzzifier*, is largely dependent on the separation between the clusters. For example, if the data set contains clusters that are far apart, then the data points can be crisply divided into various clusters, thereby leading to the hard clustering problem, with $m = 1$, and the associated membership grade for each data point turns out to be either 0 or 1. Conversely, for data sets containing clusters that are indistinguishable, a large value of m ought to be prescribed. Indeed, as $m \rightarrow \infty$, it is observed that the membership grade for each data point approaches $1/c$ (refer Höppner *et al.* (1999) for a general discussion on this subject). In our research, we have adopted the most commonly used value for m , namely $m = 2$. Observe that, unlike as in the case of hard clustering, the w -variables can now

fractionate, thereby reflecting the fuzziness with which each data point i is assigned to different clusters. Also, consistent with the optimization approach adopted in this paper, we note that for solving fuzzy clustering problems having a higher degree, some suitable pseudoglobal optimization approach coupled with factorable programming techniques might be gainfully employed (see Sherali and Wang, 2001, and Sherali and Ganesan, 2003).

Now, note that for a fixed w in Problem FCP, optimality in z requires that

$$\sum_{i=1}^n w_{ij}^2 (z_{jk} - a_{ik}) = 0, \quad \forall (j, k). \quad (3.2.2)$$

This yields,

$$z_{jk} = \frac{\sum_{i=1}^n w_{ij}^2 a_{ik}}{\sum_{i=1}^n w_{ij}^2}, \quad \forall (j, k), \text{ i.e., } z_j = \sum_i \lambda_i a_i \text{ where } \lambda_i = \frac{w_{ij}^2}{\sum_{i=1}^n w_{ij}^2}, \quad \forall i. \quad (3.2.3)$$

Hence, each cluster centroid z_j is a convex combination (since $\sum_{i=1}^n \lambda_i = 1$, $\lambda_i \geq 0$, $\forall i$ in (3.2.3)) of the vectors a_i for which $w_{ij} > 0$. With this motivation, let us define a (conveniently derived) superset approximation to the convex hull of all the data points a_i , $i = 1, \dots, n$, as given by the inequalities

$$\sum_{k=1}^s \gamma_{qk} \xi_k \leq \gamma_{q0}, \quad \forall q = 1, \dots, Q. \quad (3.2.4)$$

Accordingly, we can impose the restrictions

$$\sum_{k=1}^s \gamma_{qk} z_{jk} \leq \gamma_{q0}, \quad \forall q = 1, \dots, Q, \quad \text{for each } j. \quad (3.2.5)$$

Furthermore, given (3.2.2), the quartic objective function (3.2.1a) can be reduced to a cubic polynomial as follows.

$$\begin{aligned} \sum_{i=1}^n \sum_{j=1}^c w_{ij}^2 \|a_i - z_j\|^2 &= \sum_{i=1}^n \sum_{j=1}^c \sum_{k=1}^s w_{ij}^2 (z_{jk} - a_{ik})^2 \\ &= \sum_{i=1}^n \sum_{j=1}^c \sum_{k=1}^s w_{ij}^2 (z_{jk} - a_{ik}) z_{jk} - \sum_{i=1}^n \sum_{j=1}^c \sum_{k=1}^s w_{ij}^2 (z_{jk} - a_{ik}) a_{ik} \\ &= \sum_{j=1}^c \sum_{k=1}^s z_{jk} \left[\sum_{i=1}^n w_{ij}^2 (z_{jk} - a_{ik}) \right] - \sum_{i=1}^n \sum_{j=1}^c \sum_{k=1}^s a_{ik} w_{ij}^2 (z_{jk} - a_{ik}) \end{aligned}$$

$$= - \sum_{i=1}^n \sum_{j=1}^c \sum_{k=1}^s a_{ik} w_{ij}^2 (z_{jk} - a_{ik}). \quad (3.2.6)$$

In addition, a critical factor that can seriously inhibit the solution of FCP via a branch-and-bound (B&B) approach is the *symmetry* in the problem structure. Note that for any given solution, alternative equivalent solutions could be obtained by simply re-indexing each cluster composition, and a B&B algorithm could get mired in sifting through such symmetric reflections. To alleviate the related computational difficulties, we validly impart a somewhat distinctive identity to each cluster set by indexing them in nonincreasing order of their sizes. That is, we impose

$$\sum_{i=1}^n w_{ij} \geq \sum_{i=1}^n w_{i,j+1}, \quad \forall j = 1, \dots, c-1. \quad (3.2.7)$$

Using (3.2.2), (3.2.5), (3.2.6), and (3.2.7), we can rewrite FCP as follows where the bounds on w_{ij} can be initialized at $l_{ij} = 0$, and $u_{ij} = 1$, $\forall (i, j)$, and will be revised subsequently during the algorithmic process.

$$\text{FCP1:} \quad \text{Maximize} \quad \sum_{i=1}^n \sum_{j=1}^c \sum_{k=1}^s a_{ik} w_{ij}^2 (z_{jk} - a_{ik}) \quad (3.2.8a)$$

$$\text{subject to} \quad \sum_{i=1}^n w_{ij}^2 (z_{jk} - a_{ik}) = 0, \quad \forall (j, k) \quad (3.2.8b)$$

$$\sum_{k=1}^s \gamma_{qk} z_{jk} \leq \gamma_{q0}, \quad q = 1, \dots, Q, \quad \forall j \quad (3.2.8c)$$

$$\sum_{i=1}^n w_{ij} \geq \sum_{i=1}^n w_{i,j+1}, \quad \forall j = 1, \dots, c-1 \quad (3.2.8d)$$

$$\sum_{j=1}^c w_{ij} = 1, \quad \forall i = 1, \dots, n \quad (3.2.8e)$$

$$l_{ij} \leq w_{ij} \leq u_{ij}, \quad \forall (i, j). \quad (3.2.8f)$$

We now apply the RLT to FCP1 by generating some special additional valid inequalities. Note that in order to curtail the size of the resulting problem obtained via this process, we will only generate RLT product constraints that contain nonlinear terms of the type that are already present within FCP1. Denoting by $(3.2.8c)_{qj}$, the particular

constraint expression $\gamma_{q0} - \sum_{k=1}^s \gamma_{qk} z_{jk} \geq 0$ that appears in (3.2.8c), $\forall (q, j)$, and denoting

by $[\cdot]_L$ the linearization of an expression $[\cdot]$ under the substitution:

$$W_{ij} = w_{ij}^2, \quad x_{ijk} = w_{ij} z_{jk}, \quad \text{and} \quad y_{ijk} = w_{ij}^2 z_{jk}, \quad \forall (i, j, k), \quad (3.2.9)$$

we will generate the following constraints, $\forall q, \forall (i, j)$:

$$\begin{aligned} [(3.2.8c)_{qj} * (w_{ij} - l_{ij})^2]_L &\geq 0, \quad [(3.2.8c)_{qj} * (u_{ij} - w_{ij})^2]_L \geq 0, \quad \text{and} \\ [(3.2.8c)_{qj} * (u_{ij} - w_{ij}) * (w_{ij} - l_{ij})]_L &\geq 0. \end{aligned} \quad (3.2.10)$$

Incorporating (3.2.10) within FCP1 yields the following enhanced reformulation **FCP2**, where we have now used the substitution (3.2.9) in (3.2.8a, b) as well, and where we have re-written (3.2.8c) in (3.2.10) as (3.2.11c) below for the sake of convenience in referencing. Proposition 3.2.1 below establishes the validity of this model.

$$\text{FCP2: Maximize} \quad \sum_{i=1}^n \sum_{j=1}^c \sum_{k=1}^s a_{ik} y_{ijk} - \sum_{i=1}^n \sum_{j=1}^c \sum_{k=1}^s a_{ik}^2 W_{ij} \quad (3.2.11a)$$

$$\text{subject to} \quad \sum_{i=1}^n y_{ijk} - \sum_{i=1}^n a_{ik} W_{ij} = 0, \quad \forall (j, k) \quad (3.2.11b)$$

$$\sum_{k=1}^s \gamma_{qk} z_{jk} \leq \gamma_{q0}, \quad \forall q = 1, \dots, Q, \quad \forall j \quad (3.2.11c)$$

$$[(3.2.11c)_{qj} * (w_{ij} - l_{ij})^2]_L \geq 0, \quad \forall q, i, j \quad (3.2.11d)$$

$$[(3.2.11c)_{qj} * (u_{ij} - w_{ij})^2]_L \geq 0, \quad \forall q, i, j \quad (3.2.11e)$$

$$[(3.2.11c)_{qj} * (u_{ij} - w_{ij}) * (w_{ij} - l_{ij})]_L \geq 0, \quad \forall q, i, j \quad (3.2.11f)$$

$$\sum_{i=1}^n w_{ij} \geq \sum_{i=1}^n w_{i, j+1}, \quad \forall j = 1, \dots, c-1 \quad (3.2.11g)$$

$$\sum_{j=1}^c w_{ij} = 1, \quad \forall i = 1, \dots, n \quad (3.2.11h)$$

$$l_{ij} \leq w_{ij} \leq u_{ij}, \quad \forall (i, j) \quad (3.2.11i)$$

$$\text{Constraints (3.2.9)}. \quad (3.2.12)$$

Note that the complicating constraints (3.2.12) are a part of FCP2; however, upper bounds will be computed by solving Problem (3.2.11a - i), without constraint (3.2.12). We will refer to this linear programming (LP) relaxation as Problem $\overline{\text{FCP2}}$. Conditions under which these relaxed constraints (3.2.12) would be satisfied by an optimum to $\overline{\text{FCP2}}$, as well as the implication of other plausible RLT constraints that could have been added to this formulation while creating only the product terms of the type (3.2.9), are addressed below.

Proposition 3.2.1.

- (a) The constraints $[(w_{ij} - l_{ij})^2]_{\text{L}} \geq 0$, $[(u_{ij} - w_{ij})^2]_{\text{L}} \geq 0$, and $[(u_{ij} - w_{ij})(w_{ij} - l_{ij})]_{\text{L}} \geq 0$, $\forall (i, j)$ are implied by $\overline{\text{FCP2}}$.
- (b) The constraints $[(3.2.11c)_{qj} * (w_{ij} - l_{ij})]_{\text{L}} \geq 0$ and $[(3.2.11c)_{qj} * (u_{ij} - w_{ij})]_{\text{L}} \geq 0$, $\forall (i, j)$ are implied by $\overline{\text{FCP2}}$.
- (c) For any feasible solution $(\bar{w}, \bar{z}, \bar{W}, \bar{x}, \bar{y})$ to $\overline{\text{FCP2}}$, if $\bar{w}_{ij} = l_{ij}$ or $\bar{w}_{ij} = u_{ij}$, then we must have $\bar{W}_{ij} = \bar{w}_{ij}^2$, $\bar{x}_{ijk} = \bar{w}_{ij} \bar{z}_{jk}$, $\forall k$, and $\bar{y}_{ijk} = \bar{w}_{ij}^2 \bar{z}_{jk}$ $\forall k$, holding true, *i.e.*, the related constraints in (3.2.9) or (3.2.12) are satisfied.

Proof. To begin with, let us define $\alpha_k = \min \{\xi_k : \text{constraints (3.2.4)}\}$, and $\beta_k = \max \{\xi_k : \text{constraints (3.2.4)}\}$, $\forall k$. Note that α_k and β_k exist for all k since (3.2.4) defines a nonempty compact set. Moreover, we can compose surrogates of (3.2.4) composed by using multipliers equal to the optimal dual solutions to these problems to yield the restrictions $\xi_k \geq \alpha_k$, and $\xi_k \leq \beta_k$, $\forall k$. Applying this same surrogation process equivalently to (3.2.5) or (3.2.11c), we get

$$\alpha_k \leq z_{jk} \leq \beta_k, \quad \forall (j, k). \quad (3.2.13)$$

- (a) To prove part (a), consider the RLT constraints $[(w_{ij} - l_{ij})^2]_{\text{L}} \geq 0$. Pick some k for which $\alpha_k < \beta_k$ (this must exist; else FCP is trivial). By surrogating (3.2.11d) using the same Lagrange multipliers with respect to $(3.2.11c)_{qj}$ as those that produced (3.2.13), the algebra readily yields the constraints

$$[(z_{jk} - \alpha_k)(w_{ij} - l_{ij})^2]_{\text{L}} \geq 0 \text{ and } [(\beta_k - z_{jk})(w_{ij} - l_{ij})^2]_{\text{L}} \geq 0. \quad (3.2.14)$$

Summing the constraints in (3.2.14) (in the linearized form) produces

$$[(\beta_k - \alpha_k)(w_{ij} - l_{ij})^2]_{\text{L}} \geq 0,$$

which implies that $[(w_{ij} - l_{ij})^2]_{\text{L}} \geq 0$ because $\alpha_k < \beta_k$. The other constraints in Part (a) are similarly implied by (3.2.11e) and (3.2.11f), respectively.

(b) If $l_{ij} = u_{ij}$, then the stated constraints are null upon fixing $w_{ij} = l_{ij} = u_{ij}$ in $\overline{\text{FCP2}}$.

Hence, suppose that $l_{ij} < u_{ij}$. The constraints of Part (b) can then be obtained by summing the corresponding constraints in (3.2.11d, f) and (3.2.11e, f), respectively, and are hence implied.

(c) Finally, consider Part (c), and assume that $\bar{w}_{ij} = l_{ij}$. (The case of $\bar{w}_{ij} = u_{ij}$ is similar.)

First, let us show that $\overline{W}_{ij} = \bar{w}_{ij}^2$. By Part (a), since the stated constraints are implied by $\overline{\text{FCP2}}$, we have that when $\bar{w}_{ij} = l_{ij}$,

$$[(w_{ij} - l_{ij})^2]_{\text{L}} = [w_{ij}(w_{ij} - l_{ij})]_{\text{L}} - l_{ij}(w_{ij} - l_{ij}) \geq 0 \Rightarrow \overline{W}_{ij} \geq l_{ij}^2,$$

and similarly,

$$[(u_{ij} - w_{ij})(w_{ij} - l_{ij})]_{\text{L}} = u_{ij}(w_{ij} - l_{ij}) - [w_{ij}(w_{ij} - l_{ij})]_{\text{L}} \geq 0 \Rightarrow \overline{W}_{ij} \leq l_{ij}^2.$$

Hence, we have $\overline{W}_{ij} = l_{ij}^2 = \bar{w}_{ij}^2$.

Next, let us show that $\bar{x}_{ijk} = \bar{w}_{ij} \bar{z}_{jk}$, $\forall k$. For any k , noting (3.2.13) and Part (b), we have that the constraints of $\overline{\text{FCP2}}$ imply the restrictions $[(z_{jk} - \alpha_k)(w_{ij} - l_{ij})]_{\text{L}} \geq 0$ and $[(\beta_k - z_{jk})(w_{ij} - l_{ij})]_{\text{L}} \geq 0$. Under the condition $\bar{w}_{ij} = l_{ij}$, these constraints respectively imply that $\bar{x}_{ijk} \geq l_{ij} \bar{z}_{jk}$ and $\bar{x}_{ijk} \leq l_{ij} \bar{z}_{jk}$, which yields $\bar{x}_{ijk} = l_{ij} \bar{z}_{jk} = \bar{w}_{ij} \bar{z}_{jk}$.

Finally, let us establish that $\bar{y}_{ijk} = \bar{w}_{ij}^2 \bar{z}_{jk}$, $\forall k$. Again, for any k , noting (3.2.11d) and (3.2.13), we have that the corresponding surrogates of the former yield

$$[(z_{jk} - \alpha_k)(w_{ij} - l_{ij})^2]_{\text{L}} \geq 0 \text{ and } [(\beta_k - z_{jk})(w_{ij} - l_{ij})^2]_{\text{L}} \geq 0, \text{ i.e.,}$$

$$[z_{jk}(w_{ij} - l_{ij})^2]_{\text{L}} - \alpha_k [(w_{ij} - l_{ij})^2]_{\text{L}} \geq 0 \text{ and } [z_{jk}(w_{ij} - l_{ij})^2]_{\text{L}} - \beta_k [(w_{ij} - l_{ij})^2]_{\text{L}} \leq 0. \quad (3.2.15)$$

But when $\bar{w}_{ij} = l_{ij}$, we have $[(w_{ij} - l_{ij})^2]_L = W_{ij} + l_{ij}^2 - 2l_{ij}w_{ij} = 0$ since $\bar{W}_{ij} = l_{ij}^2$ from above. Hence, (3.2.15) asserts that when $\bar{w}_{ij} = l_{ij}$, we have $[z_{jk}(w_{ij} - l_{ij})^2]_L = 0$, *i.e.*, $\bar{y}_{ijk} + \bar{z}_{jk}l_{ij}^2 - 2l_{ij}\bar{x}_{ijk} = 0$. Using $\bar{x}_{ijk} = l_{ij}\bar{z}_{jk}$ from above, this implies that $\bar{y}_{ijk} = l_{ij}^2\bar{z}_{jk} = \bar{w}_{ij}^2\bar{z}_{jk}$. This completes the proof. \square

We now design a B&B algorithm for solving Problem FCP2, based on partitioning the hyperrectangle (3.2.11i) alone. For any node in this branch-and-bound tree, we compute an upper bound by solving the LP relaxation $\overline{\text{FCP2}}$ for the corresponding subproblem (*i.e.*, $\overline{\text{FCP2}}$ with modified bounds in (3.2.11i), and hence in (3.2.11d) – (3.2.11f)). If the resulting solution $(\bar{w}, \bar{z}, \bar{W}, \bar{x}, \bar{y})$ satisfies (3.2.12), it is optimal to this subproblem. Otherwise, a heuristic solution could be computed by fixing \bar{w} , solving for \bar{z} via (3.2.3), then fixing the resulting z -variables and solving for the w variables in (3.2.1a) – (3.2.1c) (see Remark 3.2.1 below for the relevant formulae), and so on, alternating in this fashion until the objective function value no longer improves. The node selection strategy in this process picks a node that has the greatest upper bound for further exploration. Finally, to select a branching variable, we compute the index

$$\theta_{ij} = \max \left\{ \left| \bar{W}_{ij} - \bar{w}_{ij}^2 \right|, \left| \bar{x}_{ijk} - \bar{w}_{ij}\bar{z}_{jk} \right| \text{ for all } k, \left| \bar{y}_{ijk} - \bar{w}_{ij}^2\bar{z}_{jk} \right| \text{ for all } k \right\}. \quad (3.2.16)$$

Note that by Proposition 3.2.1, if $\bar{w}_{ij} = l_{ij}$ or $\bar{w}_{ij} = u_{ij}$, then we have $\theta_{ij} = 0$. Also, if (3.2.12) is satisfied, then $\theta_{ij} = 0$, $\forall (i, j)$. Else, we select $\theta_{pq} \equiv \arg \max_{(i,j)} \{\theta_{ij}\} > 0$, which

means that $l_{pq} < \bar{w}_{pq} < u_{pq}$. The node subproblem is then split by imposing the dichotomy that

$$l_{ij} \leq w_{ij} \leq \bar{w}_{ij} \quad \text{or} \quad \bar{w}_{ij} \leq w_{ij} \leq u_{ij}. \quad (3.2.17)$$

Infinite convergence to a global optimum (in case finite termination does not occur) follows from Sherali and Tuncbilek (1992), noting Proposition 3.2.1.

Remark 3.2.1. For a fixed $z = \bar{z}$, Problem (3.2.1a) – (3.2.1c) can be solved for an optimal value of w as follows.

Proposition 3.2.2. Let $z_j = \bar{z}_j$ be fixed for all j in Problem FCP ((3.2.1a) – (3.2.1c)). Then an optimal corresponding solution \bar{w} to FCP is obtained as follows.

For each i , if $\bar{z}_r = a_i$ for some r , then set $\bar{w}_{ir} = 1$, and $\bar{w}_{ij} = 0$ for all $j \neq r$; (3.2.18a)

otherwise, let $\bar{w}_{ij} = \frac{\bar{\pi}_i}{\|a_i - \bar{z}_j\|^2}$, $\forall j$, where $\bar{\pi}_i = \frac{1}{\sum_{j=1}^c \left(\frac{1}{\|a_i - \bar{z}_j\|^2} \right)}$. (3.2.18b)

Proof. For $z = \bar{z}$ fixed, Problem FCP is a linearly constrained convex program for which the KKT conditions are both necessary and sufficient. Denoting π_i and μ_{ij} as the Lagrange multipliers associated with (3.2.1b) and (3.2.1c), respectively, $\forall i, j$, these conditions require that (where we have denoted $c_{ij} \equiv \|a_i - \bar{z}_j\|^2$, $\forall i, j$, and equivalently written the objective function as: Minimize $\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^c c_{ij} w_{ij}^2$):

$$\sum_{j=1}^c w_{ij} = 1, \quad \forall i, \quad w \geq 0, \quad (3.2.19a)$$

$$c_{ij} w_{ij} - \pi_i - \mu_{ij} = 0, \quad \mu_{ij} w_{ij} = 0, \quad \mu_{ij} \geq 0, \quad \forall (i, j). \quad (3.2.19b)$$

Consider any $i \in \{1, \dots, n\}$. Let us first show that we must have $\mu_{ij} = 0$, $\forall j$, in (3.2.19b). If any $\mu_{ij} > 0$, then (3.2.19b) implies that $w_{ij} = 0$, and so $\pi_i = -\mu_{ij} < 0$. But (3.2.19a) requires that $w_{ir} > 0$ for some r , and (3.2.19b) asserts that μ_{ir} must be zero for this (i, r) , which means that we should have $c_{ir} w_{ir} = \pi_i$. Since $c_{ir} \geq 0$, this contradicts that $\pi_i < 0$. Hence, $\mu_{ij} = 0$, $\forall j$ in (3.2.19b).

Consequently, if $c_{ir} = 0$ (i.e., $\bar{z}_r = a_i$) for any r , then by (3.2.19b), we will have $\pi_i = 0$ and the KKT conditions (3.2.19a, b) are satisfied by selecting $w_{ij} = \bar{w}_{ij}$ for all j as specified in (3.2.18a). On the other hand, if we have $c_{ij} > 0$, $\forall j$, we have from (3.2.19b) that $w_{ij} = \pi_i / c_{ij}$, $\forall j$, and using (3.2.19a), we obtain $\pi_i = \bar{\pi}_i$ and $w_{ij} = \bar{w}_{ij}$, $\forall j$, as given by (3.2.18b). This completes the proof. \square

Remark 3.2.2. Note that the proof of Proposition 3.2.2 asserts that we could impose the constraints

$$w_{ij} \sum_{k=1}^s (z_{jk} - a_{ik})^2 = \pi_i, \quad \forall (i, j), \quad (3.2.20)$$

within the reformulation of FCP2. However, doing so would produce new nonlinear terms other than those in (3.2.9) that would require additional supporting RLT constraints involving the pairwise products of (3.2.11c), or the pairwise products of the surrogated implied constraints (3.2.13), multiplied by the corresponding bound factors $(w_{ij} - l_{ij})$ and $(u_{ij} - w_{ij})$, $\forall (i, j)$. To avoid this increase in size, we do not include (3.2.20) explicitly, and permit FCP2 itself to implicitly attain these conditions ultimately.

3.2.2. Computational Results

Throughout this section, we will use the following terminology:

- Z_0 : Objective function value of FCP corresponding to the heuristic solution found at node zero.
- Z^* : Optimal objective function value of FCP, evaluated at the optimal solution to FCP2 ($\equiv -v[\text{FCP2}]$).
- Z_{FCMA}^* : Best objective function value obtained via the FCMA procedure of Bezdek (1981).
- CPU*: CPU time required to determine a global optimum for FCP2 via the proposed B&B algorithm.
- CPU_{FCMA}: CPU time required for the FCMA heuristic procedure.
- CPU₀: CPU time required to determine a heuristic solution at node zero via the solution to $\overline{\text{FCP2}}$.

To test our proposed methodology, the standard data sets from Späth (1980), which are described in Section 3.1.3, were used. The proposed B&B algorithm for solving FCP2 was implemented in C++, and the commercial solver CPLEX 8.1.0 was invoked for the purpose of solving the LP relaxations at each node. Furthermore, for

modeling our problem, the constraints (3.2.11c) in FCP2 that represent a superset of the convex hull of the data points were generated by simply constructing a tightest hyperrectangle that encloses the data points. Also, for benchmarking our results, we coded the FCMA procedure in C++, and executed this method with a prescribed termination tolerance of $\varepsilon = 10^{-3}$. Tables 3.2.1 and 3.2.2 present the relative performance of the proposed algorithm versus the FCMA procedure, measured in terms of various statistics, for three and five cluster centers, respectively.

Parameters	$\frac{Z_0}{Z^*}$	$\frac{Z_{FCMA}^*}{Z^*}$	$\frac{Z_{FCMA}^*}{Z_0}$	CPU* (s)	CPU _{FCMA} (s)	$\frac{\text{CPU}^*}{\text{CPU}_{FCMA}}$	$\frac{\text{CPU}_0}{\text{CPU}_{FCMA}}$
Data Sets							
1	2.25	2.48	1.10	0.140	0.551	0.254	0.084
2	2.43	2.99	1.23	0.274	1.161	0.236	0.078
3	2.19	4.03	1.84	0.288	2.614	0.110	0.036
4	2.68	3.55	1.32	0.410	3.585	0.114	0.037
Averages	2.387	3.26	1.37	0.278	1.977	0.140	0.046

Table 3.2.1: Relative performance of the proposed optimization approach versus the FCMA procedure for three cluster centers.

Parameters	$\frac{Z_0}{Z^*}$	$\frac{Z_{FCMA}^*}{Z^*}$	$\frac{Z_{FCMA}^*}{Z_0}$	CPU* (s)	CPU _{FCMA} (s)	$\frac{\text{CPU}^*}{\text{CPU}_{FCMA}}$	$\frac{\text{CPU}_0}{\text{CPU}_{FCMA}}$
Data Sets							
1	2.15	3.22	1.50	0.166	1.092	0.152	0.088
2	2.20	4.43	2.01	0.300	2.113	0.141	0.082
3	3.27	5.73	1.75	0.414	5.658	0.073	0.042
4	3.42	4.68	1.37	0.600	6.029	0.099	0.057
Averages	2.76	4.52	1.64	0.37	3.723	0.116	0.067

Table 3.2.2: Relative performance of the proposed optimization approach versus the FCMA procedure for five cluster centers.

Note that, on an average, the reformulated problem FCP2 required only 14.05% and 9.87% of the time taken by the FCMA, while producing optimal solutions that further improve the FCMA solutions by 69.32% and 77.88%, for the respective cases of three

and five cluster centers. Indeed, from the results in Tables 3.2.1 and 3.2.2, it can be seen that the (heuristic) solution obtained at node zero itself was uniformly better than that prescribed by the FCMA, yielding an average improvement of 26.77% and 38.93%, for three and five cluster centers, respectively. Furthermore, from the column of values CPU_0/CPU_{FCMA} in Tables 3.2.1 and 3.2.2, it can be observed that this node zero heuristic solution process consumed only 4.6% and 6.7% of the CPU time taken by the FCMA at an average, for three and five cluster centers, respectively, while yet producing superior solutions. Moreover, as evident from the results in this table, the global optimum further significantly improved upon the heuristic solution produced at node zero, and was derived within a reasonable computational effort.

Next, to further test the robustness of solving the reformulated problem FCP2 via the proposed approach, a comparative study was conducted by solving the nonlinear programs FCP and FCP1 directly, using the commercial software GAMS/BARON software (version 2.50) (see Sahinidis, 1996). The corresponding results obtained are reported in Tables 3.2.3 and 3.2.4. Assimilating the results obtained in Tables 3.2.1 through 4.2.4, note that the proposed approach required only 49.20% and 54.57% of the CPU time as consumed by BARON for solving FCP, and 84.24% and 67.8% of the CPU time consumed by BARON for solving FCP1, for the case of three and five cluster centers, respectively. Moreover, denoting the optimal objective function values obtained by solving problems FCP and FCP1 using BARON as Z_{FCP}^* and Z_{FCP1}^* , respectively, it is evident that BARON consistently produced relatively inferior solutions that respectively deviate in value from optimality (as detected by our method) by factors of 3.505 and 1.852 when solving FCP and FCP1 for the case of three cluster centers, and by factors of 3.46 and 2.037 when solving FCP and FCP1, for the case of five cluster centers. The observed robustness of our approach in comparison with BARON stems from the fact that we solve linear, rather than general convex programming relaxations, which yields more reliable bounds for fathoming purposes. Nonetheless, at least in comparison with the FCMA, the solution values obtained by BARON when solving Problem FCP1 dominated the FCMA solution values.

		Data Sets				
		1	2	3	4	Averages
FCP	CPU (s)	0.12	0.41	0.81	0.92	0.565
	$\frac{Z_{FCP}^*}{Z^*}$	3.00	3.71	3.40	3.91	3.505
FCP1	CPU (s)	0.248	0.299	0.308	0.48	0.333
	$\frac{Z_{FCP1}^*}{Z^*}$	1.06	1.83	2.10	2.42	1.852

Table 3.2.3: Relative performance of solving problems FCP and FCP1 via BARON versus the proposed approach, for three cluster centers.

		Data Sets				
		1	2	3	4	Averages
FCP	CPU (s)	0.18	0.53	0.98	1.022	0.678
	$\frac{Z_{FCP}^*}{Z^*}$	2.87	3.55	3.4	4.02	3.46
FCP1	CPU (s)	0.24	0.365	0.43	0.60	0.41
	$\frac{Z_{FCP1}^*}{Z^*}$	1.108	1.62	2.55	2.87	2.037

Table 3.2.4: Relative performance of solving problems FCP and FCP1 via BARON versus the proposed approach, for five cluster centers.

To reinforce the efficacy of our proposed approach, we also solved several additional randomly generated problems of larger sizes, and compared the results obtained with those produced by the popular FCMA procedure. The number of data points in these test instances was varied from 250 to 1000 in steps of 250, and the dimension of the space was varied from two to eight, in steps of two, thereby leading to a total of $4 \times 4 = 16$ test problems, with the smallest data set having 250 points in a two-dimensional space, and the largest problem having 1000 points in an eight-dimensional space. The number of clusters (c) for each case was taken to be either three (Table 3.2.5) or five (Table 3.2.6).

From the results displayed in Tables 3.2.5 and 3.2.6, note that the FCMA procedure requires a significantly lesser CPU time as compared with the proposed exact approach, but the best solution produced by the FCMA procedure is also substantially inferior. However, the node zero heuristic solution produced by the proposed approach uniformly dominates the FCMA solution with respect to both quality and effort in most of the problem instances, with three exceptions out of the total of 32 problems, all occurring for three centers as shaded in the rows of Table 3.2.5. On an average, to obtain

Parameters Data Sets	$\frac{Z_0}{Z^*}$	$\frac{Z_{FCMA}^*}{Z^*}$	$\frac{Z_{FCMA}^*}{Z_0}$	CPU* (s)	CPU _{FCMA} (s)	$\frac{CPU^*}{CPU_{FCMA}}$	$\frac{CPU_0}{CPU_{FCMA}}$
(250, 2)	1.23	3.33	2.71	29.52	7.09	4.17	0.55
(500, 2)	1.60	3.85	2.41	62.15	11.41	5.45	0.43
(750, 2)	1.82	4.14	2.27	125.29	23.58	5.31	0.84
(1000, 2)	2.27	4.76	2.10	150.77	37.59	4.01	0.73
(250, 4)	1.31	1.31	1.00	116.03	9.79	11.85	1.65
(500, 4)	2.14	2.09	0.98	205.57	18.33	11.22	1.53
(750, 4)	2.87	5.58	1.94	314.83	21.83	14.42	0.53
(1000, 4)	3.39	6.30	1.86	590.27	37.59	15.70	0.43
(250, 6)	1.66	3.92	2.36	320.38	50.89	6.30	0.48
(500, 6)	2.54	5.13	2.02	388.86	39.66	9.81	0.69
(750, 6)	4.07	7.23	1.78	404.66	31.22	12.96	0.67
(1000, 6)	4.92	8.40	1.71	423.79	40.18	10.55	0.55
(250, 8)	2.12	2.06	0.97	327.05	37.59	8.70	1.80
(500, 8)	3.37	5.58	1.66	787.89	43.44	18.14	0.51
(750, 8)	4.85	6.30	1.30	984.57	68.65	14.34	0.94
(1000, 8)	5.99	3.92	0.65	1188.13	85.37	13.92	0.52
Averages	2.887	5.13	1.78	401.24	35.26	11.38	0.80

Table 3.2.5: Comparative results for the proposed approach versus the FCMA procedure for randomly generated problem instances having three cluster centers.

a feasible solution to Problem FCP2 based on the node zero analysis alone, the CPU time required was 20% lesser than for the FCMA procedure, yet the quality of the solution was 43.2% better in terms of the objective function value for the three cluster center case. A

similar performance was observed for the case of five cluster centers. Note that other meta- heuristic procedures such as the genetic algorithm or simulated annealing could also be combined with the node zero analysis to derive enhanced quality feasible solutions, via $\overline{FCP2}$, either as a stand-alone procedure or within the framework of the proposed B&B algorithm. We recommend such investigations for future research.

Parameters Data Sets	$\frac{Z_0}{Z^*}$	$\frac{Z_{FCMA}^*}{Z^*}$	$\frac{Z_{FCMA}^*}{Z_0}$	CPU* (s)	CPU _{FCMA} (s)	$\frac{CPU^*}{CPU_{FCMA}}$	$\frac{CPU_0}{CPU_{FCMA}}$
(250, 2)	1.51	5.01	3.32	36.23	9.49	3.82	0.70
(500, 2)	1.97	5.79	2.94	72.06	15.91	4.53	0.54
(750, 2)	2.24	6.22	2.78	141.39	33.97	4.16	0.96
(1000, 2)	2.79	7.16	2.57	169.36	54.77	3.09	0.92
(250, 4)	1.61	1.97	1.22	131.22	13.50	9.72	1.09
(500, 4)	2.63	3.14	1.19	229.53	26.18	8.77	1.04
(750, 4)	3.53	8.39	2.38	349.50	31.37	11.14	0.67
(1000, 4)	4.17	9.47	2.27	651.94	54.77	11.90	0.54
(250, 6)	2.04	5.89	2.89	355.60	74.51	4.77	0.61
(500, 6)	3.13	7.71	2.46	430.79	57.84	7.45	0.87
(750, 6)	5.01	10.87	2.17	448.14	45.31	9.89	0.85
(1000, 6)	6.06	12.63	2.08	469.14	58.61	8.00	0.70
(250, 8)	2.61	3.10	1.19	362.92	54.77	6.63	1.28
(500, 8)	4.15	8.39	2.02	868.92	63.45	13.69	0.65
(750, 8)	5.97	9.47	1.59	1084.88	100.87	10.75	0.79
(1000, 8)	7.37	5.89	0.80	1308.39	125.69	10.41	0.66
Averages	3.55	7.71	2.17	444.38	51.31	8.66	0.804

Table 3.2.6: Comparative results for the proposed approach versus the FCMA procedure for large problem instances having five cluster centers.

3.2.3. Summary, Conclusions, and Extensions for Further Research

In Section 3.2, we have addressed the design of a global optimization approach to the fuzzy clustering problem, where the objective function seeks to minimize the total degree-two fuzzifier weighted squared Euclidean distance from each data point to the

centroids of the clusters to which it is assigned, and requires an accompanying membership grade to be assigned to each data point that reflects the possibility of a data point belonging to each particular cluster. A series of enhanced reformulations of this problem were presented, augmented by optimality-induced, symmetry-defeating, and RLT-based inequalities, and a specialized branch-and-bound algorithm was designed for solving the resulting model representation. Several computational experiments were performed using standard data sets as well as synthetically generated test cases to explore the efficacy of the proposed exact solution approach, as well as to study the effectiveness of the heuristic scheme implemented at the root node. This performance was compared with the FCMA procedure (see Bezdek, 1981) that is popularly used in the literature on this topic. The results revealed the viability and robustness of the proposed approach, and exhibited its superiority over the FCMA procedure, even as a heuristic based on the node zero analysis. Note that in practice, cluster analysis problems can involve very large data sets, and therefore, good heuristic procedures can prove to be critically important for handling such problem instances. Our research provides an additional scope and impetus for designing effective heuristic methods based on constructs derived from the proposed exact optimization approach, and offers a rich potential for future advances in the domain of cluster analysis.

4. Risk Management with Equity Considerations

Risk Management is primarily concerned with the allocation of resources to attenuate the probability of a risk occurring or to mitigate hazards that might have already occurred. In recent times, risk management has rapidly developed into a field that now encompasses elements such as quality control, occupational health and safety, individual security, environmental liability, socio-economic issues, data-base and Internet systems, *etc.* Given the obvious economic benefits, it has now become essential to practice risk management in both the government and private sectors.

The issue of mitigating risks using risk management techniques has been addressed in several areas such as finance, electrical power systems, environmental hazard reduction, and increasing the safety of individuals. Quantitative approaches to reduce risks in different settings involving nuclear radiation exposure and social resource management have been considered by Rivard (1971), Weinstein (1979), and Lichtenberg and Zilberman (1988). The monitoring of policy decisions within governmental programs, and especially those that increase the longevity of human life, is discussed in Fisher *et al.* (1988). Mathematical approaches for determining routes that minimize the risk of low probability-high consequence accidents associated with hazardous material (hazmat) transfer, along with issues related to data acquisition and algorithmic computations, are addressed in Sherali *et al.* (1997), and Sivakumar *et al.* (1993). Amendola *et al.* (2000) describe a systems approach to modeling catastrophic risk and insurability using a spatial-dynamic stochastic optimization model. A survey of approaches to assess and manage extremely risky events is compiled in Bier *et al.* (1999).

It is important to note that there exist several other strategic considerations that ought to be taken into account in determining an optimal risk management technique. One such consideration is the issue of *risk acceptability*, and another is that of *risk equity*. An analytical approach to assess risk-benefit tradeoffs that lead to maximizing net social benefits was developed by Starr and Whipple (1982). Young (1994) discusses equity-related concepts, and provides a comprehensive understanding associated with the equitable distribution of goods, progressive taxation, impartiality, and consistency. In a similar vein, Luss (1999) reviewed a variety of resource allocation problems in which an

equitable distribution of limited resources among competing objectives is required. He defined an equitable allocation to be one in which no performance function can be feasibly improved without degrading another activity's performance value that is greater than or equal to this one, and showed that a lexicographic minimax vector of performance ratings (arranged in nonincreasing order) yields an equitable solution. Sherali *et al.* (2003) devised a national airspace planning model for selecting flight plans under air traffic control workload, flight safety, and airline equity considerations. Feldman *et al.* (2002) discussed the various mechanisms available to American universities in managing the commercialization of intellectual property, considering equity as a technology transfer mechanism that offers an advantage for generating revenues while simultaneously aligning the interests of universities, faculty, and industry. In addition, risk equity has been addressed with respect to hazardous waste management and disposal (Atlas, 2001), disparities in government programs with respect to cost and risk reduction, especially in the health and safety sectors (Morgan, 2000), and aspects related to the effects of upper-management stock ownership and firm diversification (Eisenmann, 2002).

Other papers related to risk ceiling and risk reduction also appear in the literature. Mosler (1997) presented a model that develops a priority listing to evaluate various strategies in risk management. These priority indices reflect the decrease in risk of an individual while maintaining the goal of reducing collective risk and simultaneously satisfying risk equity. Sherali *et al.* (1995) developed models and algorithms to determine an optimal mix of available strategies that attempt to attenuate risks and associated costs, subject to budgetary and resource constraints.

Risk management has also made recent forays in the areas of nuclear safety, energy industries, pipeline safety, software design, and space agency issues. Insurance-related industries are typically considered to have little interest in energy issues, unless they are associated with large supply systems (Mills, 2002). However, risk management tools for power systems planning now include a multiple criteria decision-making and risk analysis framework (Linares, 2002) and an optimization approach to purchase options in dual electric power markets (Liu and Guan, 2002). In particular, Kafka (2002) has dealt with methodologies and basic models for risk identification and assessment

policies for risk control measures and for goal settings in a nuclear safety environment. An overview of risk management activities for NASA's Space Shuttle Upgrades Development (SSUD) Program is described in Turner (2002) and the integrated risk management process adopted by the International Space Station (ISS) is discussed in Sebastian (2002). Risk analysis related to the tiles of the space shuttle orbiter was explored by Paté-Cornell and Fischbeck (1994), well in advance of the 2002 Columbia space shuttle disaster, and risk management of future reusable launch vehicle missions with a focus on active health monitoring systems has been considered by Renson (2002).

The safety of pipelines and software design are areas where risk management has proven to be an invaluable tool. Issues related to the protection of pipelines and their operational maintenance have been discussed by Porter and Savigny (2002), Gonzalez *et al.* (2002), and Fenyvesi (2002). A quantitative risk management aid to refinery construction was advocated by Dey (2002). Descriptions of software risk management can be found in Murthi (2002) and Freimut *et al.* (2001).

In the present research effort, we are primarily concerned with assisting agencies that deal with emergency situations by developing the basis of a decision-support system that can help them respond quickly and effectively to a given situation. For example, in the aftermath of a tornado, office buildings might have collapsed, fires might have started, and flooding might have occurred due to a water main getting severed. In such a scenario, the *emergency manager* would like to effectively and equitably deploy the limited resources available at his or her disposal to mitigate the consequences of the disaster.

The remainder of Chapter 4 is organized as follows. In Section 4.1, we formulate an emergency response model and develop a tight linear programming relaxation for this problem through suitable transformations of variables and polyhedral approximations. Section 4.2 describes a specialized branch-and-bound procedure to solve the formulated problem and provides a theoretical proof of convergence of the proposed algorithm. Some computational experience and evaluation of alternative algorithmic strategies are presented in Section 4.3 based on data pertaining to a hypothetical mid-size city. Finally, Section 4.4 concludes the section with a summary and directions for future research.

4.1. Emergency Response Model

Consider a situation in which emergency plans need to be developed to address a potential hazardous scenario (or a set of such scenarios), or in which an emergency is underway and critical response decisions must be made. For example, in the immediate aftermath of an earthquake, emergency responders might have to deal with collapsed structures, fires, traffic accidents, and hazardous material spills. In order to mitigate the hazards, the emergency manager would typically call into play a variety of available resources to perform search and rescue operations, fight fires, respond to medical emergencies, and so on. Let $i \in I$ index such a set of resources (*e.g.* personnel and equipment combinations). In appropriate units, let b_i denote the available level of resource $i \in I$. Also, let us suppose that the affected region has been partitioned into areas (indexed by $j \in J$) based on population density, land-use, critical facility locations, *etc.* Furthermore, let $k \in K_j$ index the set of hazards affecting area j that the emergency manager needs to address for each $j \in J$, and suppose that we have determined ratings r_{jk} , $\forall j \in J, k \in K_j$, to reflect the relative importance of responding to the situation created by hazard k in the affected area j , where $r_{jk} \in [1, 10]$.

Now, in each area $j \in J$, let the unmitigated risk associated with hazard type $k \in K_j$ be denoted by α_{jk} . Following the conventional definition of risk, α_{jk} is the product of two factors: probability of some ill-occurrence and its associated consequence (*e.g.*, monetary penalty). Assume that when an amount x_{ijk} of resource i is allotted to hazard type k in area j , it serves to attenuate the risk α_{jk} by an exponential factor $e^{-\beta_{ik} x_{ijk}}$, for some suitable parameter value $\beta_{ik} \geq 0$, reducing it to $\alpha_{jk} e^{-\beta_{ik} x_{ijk}}$. Also, in practice, note that the attenuation factor β_{ik} is independent of the area j and is dependent only on resource i and hazard type k . This attenuation could either be due to a reduction in the probability factor or a mitigation of the consequences associated with the risk. Hence, the overall attenuation of α_{jk} due to all resource applications is given by

$$\alpha_{jk} \prod_{i \in I} e^{-\beta_{ik} x_{ijk}} = \alpha_{jk} e^{-\sum_{i \in I} \beta_{ik} x_{ijk}},$$

leading to a measure of the *weighted mitigated risk in area j* as given by

$$R_j = \sum_{k \in K_j} r_{jk} \alpha_{jk} e^{-\sum_{i \in I} \beta_{ik} x_{ijk}}, \quad \forall j \in J. \quad (4.1)$$

Summing over all areas yields the *overall system weighted mitigated risk*:

$$R = \sum_{j \in J} R_j \equiv \sum_{j \in J} \sum_{k \in K_j} r_{jk} \alpha_{jk} e^{-\sum_{i \in I} \beta_{ik} x_{ijk}}. \quad (4.2)$$

In addition, the emergency manager might wish to commit a minimal level $L_{ijk} \geq 0$ of resource i to hazard k in area j , *i.e.*, we have $x_{ijk} \geq L_{ijk}$, $\forall i \in I, j \in J, k \in K_j$. Naturally, we have $\sum_{j \in J} \sum_{k \in K_j} L_{ijk} \leq b_i, \forall i \in I$.

In order to accommodate a relative degree of equity among the affected areas while allocating the limited available resources, let us examine the *overall risk attenuation factor for area j* as given by

$$\gamma_j \equiv R_j / \sum_{k \in K_j} r_{jk} \alpha_{jk}, \quad \forall j \in J, \quad (4.3)$$

and also, let us compute the *mean weighted attenuation factor*

$$\bar{\gamma} \equiv \sum_{j \in J} \gamma_j / |J|, \quad (4.4)$$

where $|J|$ denotes the total number of areas under consideration. Then, in addition to minimizing the overall system weighted mitigated risk R as given by (4.2), we would also like to simultaneously minimize the total spread of the γ_j -values from their mean $\bar{\gamma}$, given by $\sum_{j \in J} |\gamma_j - \bar{\gamma}|$, in order to achieve a relative degree of equity in the allocation scheme. Furthermore, note that it is possible to have more than one solution that yields the same value for the total absolute deviation $\sum_{j \in J} |\gamma_j - \bar{\gamma}|$, but the maximum spread might be more in one case than in the other. Hence, in addition to a total deviation term, it is necessary to minimize the maximum *inequity* (refer Sherali *et al.* (2003) for further discussion), denoted as $\text{maximum}_{j \in J} \{\gamma_j - \bar{\gamma}\}$, so as to confine the attenuation factors within a limited range. Accordingly, we prescribe an objective function:

$$\text{Minimize } R + \mu_D \sum_{j \in J} |\gamma_j - \bar{\gamma}| + \mu_R \max_{j \in J} \{\gamma_j - \bar{\gamma}\}, \quad (4.5)$$

where the factors $\mu_D \geq 0$ and $\mu_R \geq 0$ compromise appropriately between the two objectives of reducing the overall system weighted mitigated risk, R , and achieving an acceptable degree of equity in this process. Also, for the purpose of making the various parameter values commensurate, we assume that $\mu_R = |J| \mu_D$ in our implementation. In applying the prescribed model developed below using the objective function (4.5), it is anticipated that the emergency manager would perform various experimental runs using different nonnegative values of the parameter μ_D in order to study the sensitivity and character of the resulting solutions to variations in this parameter. Having generated such a frontier of solutions that compromise between the system efficiency and equity considerations, the emergency manager could then subjectively choose a solution that strikes the best compromise to some desired extent. In our case study application, we will perform such a study to demonstrate the sensitivity with respect to variations in the parameter μ_D .

In order to linearize the terms in the prescribed objective function, let η_j and η represent the absolute difference $|\gamma_j - \bar{\gamma}|$ and the $\max_{j \in J} \{\gamma_j - \bar{\gamma}\}$ terms in (4.5), respectively. Using (4.1) - (4.5), we can state the *emergency response model (ERM)* as follows.

$$\text{ERM: Minimize } \sum_{j \in J} \sum_{k \in K_j} r_{jk} \alpha_{jk} e^{-\sum_{i \in I} \beta_{ik} x_{ijk}} + \mu_D \sum_{j \in J} \eta_j + \mu_R \eta \quad (4.6a)$$

subject to

$$\sum_{j \in J} \sum_{k \in K_j} x_{ijk} \leq b_i, \quad \forall i \in I \quad (4.6b)$$

$$\eta_j \geq \gamma_j - \bar{\gamma}, \quad \forall j \in J \quad (4.6c)$$

$$\eta_j \geq \bar{\gamma} - \gamma_j, \quad \forall j \in J \quad (4.6d)$$

$$\eta \geq \gamma_j - \bar{\gamma}, \quad \forall j \in J \quad (4.6e)$$

$$\gamma_j \left(\sum_{k \in K_j} r_{jk} \alpha_{jk} \right) = \sum_{k \in K_j} r_{jk} \alpha_{jk} e^{-\sum_{i \in I} \beta_{ik} x_{ijk}}, \quad \forall j \in J \quad (4.6f)$$

$$|J| \bar{\gamma} = \sum_{j \in J} \gamma_j \quad (4.6g)$$

$$x_{ijk} \geq L_{ijk}, \quad \forall i \in I, j \in J, k \in K_j. \quad (4.6h)$$

In this formulation, the objective function seeks the aforementioned compromise between system efficiency and equity, constraints (4.6b) enforce the resource availability restrictions, constraints (4.6c, d) along with the second objective term (for $\mu_D > 0$) essentially yield $\eta_j = |\gamma_j - \bar{\gamma}|$, $\forall j \in J$, constraints (4.6e) along with the third term in the objective function (for $\mu_R > 0$) yield $\eta = \max_{j \in J} \{\gamma_j - \bar{\gamma}\}$, constraints (4.6f) and (4.6g) respectively represent the identities (4.3) and (4.4), where R_j is given by (4.1), $\forall j \in J$, and constraints (4.6h) require the principal decision variables x_{ijk} to be at least some nonnegative value L_{ijk} , $\forall i, j, k$, thereby ensuring that all hazards achieve a minimum required level of mitigation. Note that the variables γ_j , $\forall j \in J$, and $\bar{\gamma}$ could be eliminated from the model using (4.6f) and (4.6g), respectively. More importantly, observe that since the second term in the objective function essentially involves an absolute difference of two convex functions of $x \equiv (x_{ijk}, \forall i, j, k)$, ERM is a nonconvex programming problem.

In order to transform ERM into an equivalent formulation that is more amenable to algorithmic manipulations, let us perform the following substitutions. Let

$$y_{jk} = e^{-\sum_{i \in I} \beta_{ik} x_{ijk}} \quad \text{and} \quad z_{jk} = \ln(y_{jk}) = -\sum_{i \in I} \beta_{ik} x_{ijk}, \quad \forall j \in J, k \in K_j. \quad (4.7)$$

For algorithmic purposes, we will also need to incorporate lower and upper bounds on each variable y_{jk} as given by

$$l_{jk} \leq y_{jk} \leq u_{jk}, \quad \forall j \in J, k \in K_j, \quad (4.8a)$$

where initially, noting (4.6b), (4.6g), and (4.7), we can take

$$l_{jk} = l_{jk}^0 \equiv e^{-\sum_{i \in I} \beta_{ik} [b_i - \sum_{(p,q) \neq (j,k)} L_{ipq}]} \quad \text{and} \quad u_{jk} = u_{jk}^0 \equiv e^{-\sum_{i \in I} \beta_{ik} L_{ijk}}, \quad \forall j \in J, k \in K_j. \quad (4.8b)$$

Note that during the process of the algorithm, these bounds in (4.8a) will be revised via a suitable partitioning scheme within a branch-and-bound framework. For any such suitable lower and upper bound vectors, l and u , imposed on the y -variables as in (4.8a), we derive the following formulation **ERM**(l, u) from ERM under (4.7), where ERM is equivalent to $\text{ERM}(l^0, u^0)$, with l^0 and u^0 as given by (4.8b), and where L represents the vector of lower bounds (L_{ijk}).

$$\mathbf{ERM}(l, u): \quad \text{Minimize} \quad \sum_{j \in J} \sum_{k \in K_j} r_{jk} \alpha_{jk} y_{jk} + \mu_D \sum_{j \in J} \eta_j + \mu_R \eta \quad (4.9a)$$

subject to

$$\sum_{j \in J} \sum_{k \in K_j} x_{ijk} \leq b_i, \quad \forall i \in I \quad (4.9b)$$

$$\eta_j \geq \gamma_j - \bar{\gamma}, \quad \forall j \in J \quad (4.9c)$$

$$\eta_j \geq \bar{\gamma} - \gamma_j, \quad \forall j \in J \quad (4.9d)$$

$$\eta \geq \gamma_j - \bar{\gamma}, \quad \forall j \in J \quad (4.9e)$$

$$\gamma_j \left(\sum_{k \in K_j} r_{jk} \alpha_{jk} \right) = \sum_{k \in K_j} r_{jk} \alpha_{jk} y_{jk}, \quad \forall j \in J \quad (4.9f)$$

$$|J| \bar{\gamma} = \sum_{j \in J} \gamma_j \quad (4.9g)$$

$$z_{jk} + \sum_{i \in I} \beta_{ik} x_{ijk} = 0, \quad \forall j \in J, k \in K_j \quad (4.9h)$$

$$z_{jk} = \ln(y_{jk}), \quad \forall j \in J, k \in K_j \quad (4.9i)$$

$$x \geq L, \quad l \leq y \leq u. \quad (4.9j)$$

Observe that $\text{ERM}(l, u)$ is linear except for the complicating side-constraints (4.9i). In order to develop a relaxation **RERM**(l, u) of $\text{ERM}(l, u)$, we replace (4.9i) by a polyhedral outer approximation given by the affine convex envelope of the concave function $\ln(y_{jk})$ over $[l_{jk}, u_{jk}]$, along with some $n_{jk} \geq 2$ affine tangential supports, as shown in Figure 5.1. This yields the following relaxation $\text{RERM}(l, u)$ of $\text{ERM}(l, u)$.

RERM(l, u): Minimize $\{(4.9a):(4.9b)-(4.9h), (4.9j), \text{ along with } (4.10b, c)\}$ (4.10a)

where,

$$z_{jk} \geq \ln(l_{jk}) + \frac{[\ln(u_{jk}) - \ln(l_{jk})]}{(u_{jk} - l_{jk})} (y_{jk} - l_{jk}), \quad \forall j \in J, k \in K_j, \quad (4.10b)$$

$$z_{jk} \leq \ln(\bar{y}_{jk}) + \frac{(y_{jk} - \bar{y}_{jk})}{\bar{y}_{jk}}, \quad \forall \bar{y}_{jk} \in \left\{ l_{jk} + \frac{t}{(n_{jk} - 1)} (u_{jk} - l_{jk}), \text{ for } t = 0, \dots, n_{jk} - 1 \right\}. \quad (4.10c)$$

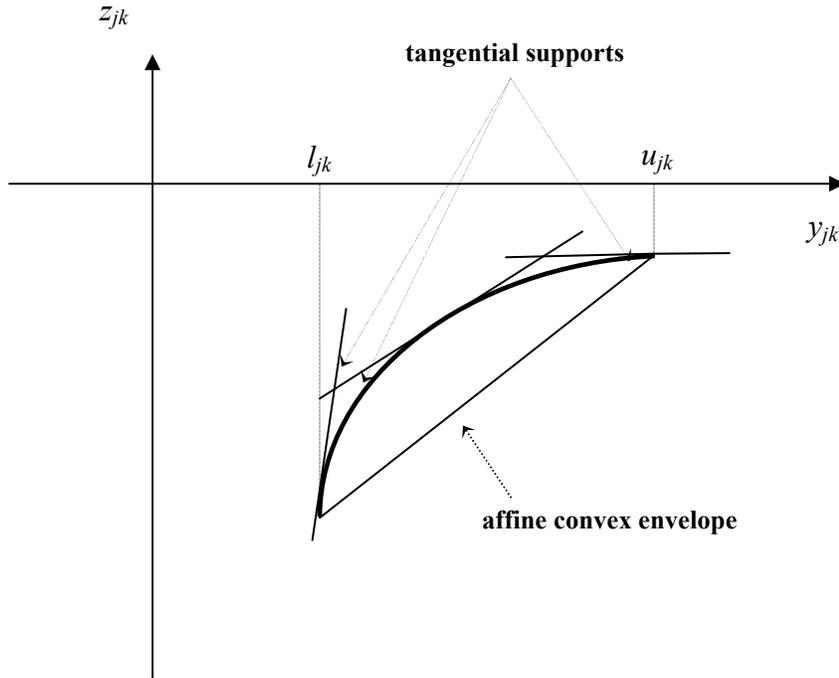


Figure 4.1: Polyhedral Outer Approximation for $z_{jk} = \ln(y_{jk})$ over $0 < l_{jk} \leq y_{jk} \leq u_{jk} \leq 1$.

Remark 4.1. As an alternative to the n_{jk} supports used in (4.10c), we can use three supports, respectively at $\bar{y}_{jk} = l_{jk}$, $\bar{y}_{jk} = u_{jk}$, and $\bar{y}_{jk} = \hat{y}_{jk}$, respectively, where the lattermost point lies in $[l_{jk}, u_{jk}]$ and is such that it minimizes the maximum approximation error (*i.e.*, the approximation errors it induces at l_{jk} and u_{jk} are equal).

This value is readily verified to be given by

$$\hat{y}_{jk} = \frac{(u_{jk} - l_{jk})}{\ln(u_{jk}/l_{jk})}. \quad (4.11)$$

4.2. Branch-And-Bound Algorithm to Solve Problem ERM

We now develop a branch-and-bound procedure to solve Problem ERM. At each stage s of this procedure, $s = 0, 1, 2, \dots$, we will have a set of non-fathomed or *active nodes* Q_s , where each node $q \in Q_s$ is indexed by some lower-upper bounding vector (l^q, u^q) for the y -variables. (To initialize, at $s = 0$, the set $Q_0 = \{0\}$, with (l^0, u^0) being given by (4.8b).) For each node $q \in Q_s$, a lower bound LB_q will be given by $v[\text{RERM}(l^q, u^q)]$, where $v[\text{P}]$ denotes the optimal value for any Problem P. As a result, the *global lower bound* at stage s for problem ERM (equivalently, Problem $\text{ERM}(l^0, u^0)$) is given by

$$LB(s) \equiv \text{minimum } \{LB_q : q \in Q_s\}. \quad (4.12)$$

Whenever any lower bounding node subproblem is solved, we can take the x -part of its solution, which is feasible to (4.9b) and (4.9i), and directly substitute this into the objective function formula given by (4.5), where the different terms in this objective representation are defined in (4.1) - (4.4), in order to derive an upper bound on the overall problem ERM. Accordingly, let x^* be the best such incumbent solution found, having an objective value of v^* . Naturally, whenever $LB_q \geq v^*$, we fathom node q . (Practically, we can fathom node q whenever $LB_q \geq v^*(1-\varepsilon)$, for some percentage optimality tolerance $100\varepsilon \% \geq 0$.) Hence, the active nodes at any stage s would satisfy $LB_q < v^*$, $\forall q \in Q_s$. From this set of active nodes, we now select a node $q(s)$ that yields the least lower bound, *i.e.*, for which $LB_{q(s)} = LB(s)$ as given by (4.12). Note that for the corresponding solution

$$\xi^{q(s)} \equiv \left(x^{q(s)}, y^{q(s)}, z^{q(s)}, \gamma^{q(s)}, \bar{\gamma}^{q(s)}, \eta^{q(s)} \right) \quad (4.13)$$

to $\text{RERM}(l^{q(s)}, u^{q(s)})$, we could not possibly have $z_{jk}^{q(s)} = \ln(y_{jk}^{q(s)})$, $\forall (j, k)$, because then, $\xi^{q(s)}$ would be feasible to $\text{ERM}(l^{q(s)}, u^{q(s)})$, thereby yielding $LB_{q(s)} \geq v^*$, a contradiction. Hence, we find a *branching variable* $y_{j^*k^*}$ according to:

$$\left| z_{j^*k^*}^{q(s)} - \ln(y_{j^*k^*}^{q(s)}) \right| \in \arg \max_{j \in J, k \in K_j} \left| z_{jk}^{q(s)} - \ln(y_{jk}^{q(s)}) \right| \quad (4.14a)$$

and we partition the interval $[l_{j^*k^*}^{q(s)}, u_{j^*k^*}^{q(s)}]$ for $y_{j^*k^*}$ in the subproblem for node $q(s)$ into two subintervals, one for each *child node* or *subnode* generated, as follows:

$$[l_{j^*k^*}^{q(s)}, y_{j^*k^*}^{q(s)}] \quad \text{and} \quad [y_{j^*k^*}^{q(s)}, u_{j^*k^*}^{q(s)}], \quad (4.14b)$$

where $y_{j^*k^*}^{q(s)}$ is as defined in (4.13). Note that since the polyhedral approximation (4.10) is exact at the interval endpoints, by virtue of (4.14a), we must have

$$y_{j^*k^*}^{q(s)} \in (l_{j^*k^*}^{q(s)}, u_{j^*k^*}^{q(s)}). \quad (4.15)$$

We will refer to the branching strategy embodied by (4.14) as **Branching Rule A**, in order to distinguish it from other viable partitioning strategies discussed in the sequel. A formal statement of this proposed algorithm is given below.

Branch-and-Bound Algorithm for Problem ERM

Step 0: Initialization. Set $s = 0$, $Q_s = \{0\}$, $q(s) = 0$, $q = 0$, and let (l^0, u^0) be given by (4.8b). Solve the linear program $\text{RERM}(l^0, u^0)$ and let ξ^0 be the solution obtained (as defined in (4.13)) having an objective value LB_0 . Set the incumbent solution $x^* = x^0$, and let the incumbent objective value be v^* as given via (4.5) for $x = x^*$. If $\text{LB}_0 \geq v^*(1 - \varepsilon)$, for some optimality tolerance $\varepsilon \geq 0$, then stop with the incumbent solution as (ε) -optimal to Problem ERM. Otherwise, find a branching variable $y_{j^*k^*}$ via (4.14), and proceed to Step 1.

Step 1: Partitioning Step. Partition the current selected node $q(s)$ into two subnodes indexed by $q+1$ and $q+2$ according to the Branching Rule A given by (4.14), and replace $Q_s \leftarrow Q_s \cup \{q+1, q+2\} - \{q(s)\}$. Let (l^{q+i}, u^{q+i}) , $i = 1, 2$, be the respective bounds on the y -variables for the corresponding nodes $q+1$ and $q+2$.

Step 2: Bounding Step. Solve $\text{RERM}(l^{q+i}, u^{q+i})$, for each $i = 1, 2$. Update the incumbent solution x^* and its value v^* , if possible, using the corresponding x -variable parts of the resulting solutions along with (4.5), and determine branching variable indices according

to (4.14a) for each of these nodes (provided that their lower bounds are lesser than $v^*(1 - \varepsilon)$) for possible future use. Replace $q \leftarrow q + 2$.

Step 3: Fathoming Step. Fathom any potentially non-improving nodes by setting $Q_{s+1} = Q_s - \{\hat{q} \in Q_s : \text{LB}_{\hat{q}} \geq v^*(1 - \varepsilon)\}$. Increment s by 1.

Step 4: Termination Check and Node Selection. If $Q_s = \emptyset$, then stop with the incumbent solution as (ε) -optimal. Otherwise, select an active node $q(s) \in \arg \min\{\text{LB}_{\hat{q}} : \hat{q} \in Q_s\}$, and return to Step 1.

Theorem 4.1. (Main Convergence Result). *The foregoing algorithm (run with $\varepsilon \equiv 0$) either terminates finitely with the incumbent solution being optimal to Problem ERM, or else an infinite sequence of stages is generated such that along any infinite branch of the branch-and-bound tree, any accumulation point of the x -variable part of the sequence of linear programming relaxation solutions generated for the corresponding node subproblems solves Problem ERM.*

Proof. The case of finite termination is clear. Hence, suppose that an infinite sequence of stages is generated. Consider any infinite branch of the branch-and-bound tree generated via the sequence of nested intervals $\{[l^{q(s)}, u^{q(s)}]\}$ that correspond to a set of stages s in some index set S . Hence, we have

$$\text{LB}(s) = \text{LB}_{q(s)} \equiv v[\text{RERM}(l^{q(s)}, u^{q(s)})], \quad \forall s \in S. \quad (4.16)$$

For each node $q(s)$, $s \in S$, let $\xi^{q(s)}$ be the solution obtained for $\text{RERM}(l^{q(s)}, u^{q(s)})$ as defined in (4.13). By taking any convergent subsequence, if necessary, using the boundedness of the sequence generated (and noting that the feasible region is a compact set), assume without loss of generality that

$$\{\xi^{q(s)}, l^{q(s)}, u^{q(s)}\}_S \rightarrow (\xi^*, l^*, u^*). \quad (4.17)$$

We must show that the x -variable part x^* of the solution ξ^* solves Problem ERM.

First, note that since $\text{LB}_{q(s)}$ is the least lower bound at stage s , we have

$$\text{LB}_{q(s)} \leq v[\text{ERM}], \quad \forall s \in S. \quad (4.18)$$

Second, observe that in the infinite sequence of nodes $q(s)$ for $s \in S$, there exists some variable $y_{j^*k^*}$ that is selected as the branching variable infinitely often via (4.14a). Let $S_1 \subset S$ index the set of nodes where this occurs, so that from (4.14a), we have

$$\left| z_{j^*k^*}^{q(s)} - \ln(y_{j^*k^*}^{q(s)}) \right| \geq \left| z_{jk}^{q(s)} - \ln(y_{jk}^{q(s)}) \right|, \forall j \in J, k \in K_j, \text{ and for each } s \in S_1. \quad (4.19)$$

Now, from (4.17), by the continuity of the linear programming relaxations we have that ξ^* is feasible to $\text{RERM}(l^*, u^*)$. Moreover, by the partitioning scheme (4.14b), we know that for each $s \in S_1$, we have $y_{j^*k^*}^{q(s)} \notin (l_{j^*k^*}^{q(s')}, u_{j^*k^*}^{q(s')})$, $\forall s' \in S_1, s' > s$, while in the limit as $s \rightarrow \infty, s \in S_1$, we have that $y_{j^*k^*}^* \in [l_{j^*k^*}^*, u_{j^*k^*}^*]$. Hence, we must have

$$y_{j^*k^*}^* = l_{j^*k^*}^* \text{ or } y_{j^*k^*}^* = u_{j^*k^*}^*. \quad (4.20)$$

Since the polyhedral approximation (4.10) is exact at the interval endpoints, this implies that

$$z_{j^*k^*}^* = \ln(y_{j^*k^*}^*). \quad (4.21)$$

Hence, taking limits in (4.19) as $s \rightarrow \infty$ for $s \in S_1$, we get using (4.17) and (4.21) that $z_{jk}^* = \ln(y_{jk}^*), \forall j \in J, k \in K_j$, i.e., ξ^* is feasible to $\text{ERM}(l^*, u^*)$, yielding an objective value v^* that coincides with the value that would be obtained by substituting $x = x^*$ into (4.5). Consequently, from (4.16)

$$\lim_{\substack{s \rightarrow \infty \\ s \in S_1}} \text{LB}_{q(s)} = \lim_{\substack{s \rightarrow \infty \\ s \in S_1}} v[\text{RERM}(l^{q(s)}, u^{q(s)})] = v^* \geq v[\text{ERM}]. \quad (4.22)$$

But (4.18) then asserts that

$$\lim_{\substack{s \rightarrow \infty \\ s \in S_1}} \text{LB}_{q(s)} \leq v[\text{ERM}], \quad (4.23)$$

which, together with (4.22), yields $v^* = v[\text{ERM}]$, or that x^* solves Problem ERM. This completes the proof. \square

Corollary 4.1. The convergence Theorem 4.1 holds true under either of the following Branching Rules B, C, or D, as alternatives to Branching Rule A:

i) **Branching Rule B:** For a selected node $q(s)$, find a branching variable $y_{j^*k^*}$ as in (4.14a), and partition this node's subproblem by bisecting the corresponding interval $[l_{j^*k^*}^{q(s)}, u_{j^*k^*}^{q(s)}]$ in lieu of splitting this interval at the value $y_{j^*k^*}^{q(s)}$ as in (4.14b).

ii) **Branching Rule C:** For a selected node $q(s)$, find a branching variable $y_{j^*k^*}$ according to

$$\left(u_{j^*k^*}^{q(s)} - l_{j^*k^*}^{q(s)}\right) = \text{maximum} \left\{ \left(u_{jk}^{q(s)} - l_{jk}^{q(s)}\right) : j \in J, k \in K_j \right\}, \quad (4.24)$$

and bisect the current interval for $y_{j^*k^*}$ as in Branching Rule B.

iii) **Branching Rule D:** For a selected node $q(s)$, find a branching variable $y_{j^*k^*}$ as in (4.14a) and partition this node's subproblem by splitting the corresponding interval $[l_{j^*k^*}^{q(s)}, u_{j^*k^*}^{q(s)}]$ for $y_{j^*k^*}$ at the value $\tilde{y}_{j^*k^*}^{q(s)}$, where

$$\tilde{y}_{j^*k^*}^{q(s)} = \begin{cases} \hat{y}_{j^*k^*}^{q(s)} \text{ as given by (4.11), with } (l_{jk}, u_{jk}) \leftarrow (l_{j^*k^*}^{q(s)}, u_{j^*k^*}^{q(s)}), & \text{if} \\ \max \left\{ \hat{y}_{j^*k^*}^{q(s)} - l_{j^*k^*}^{q(s)}, u_{j^*k^*}^{q(s)} - \hat{y}_{j^*k^*}^{q(s)} \right\} \leq 0.9 \left(u_{j^*k^*}^{q(s)} - l_{j^*k^*}^{q(s)} \right) & (4.25) \\ y_{j^*k^*}^{q(s)} \text{ as given via (4.13), otherwise.} & \end{cases}$$

Proof. The proof of Theorem 4.1 for Branching Rule B holds true identically in this case, noting that we get $l_{j^*k^*}^* = u_{j^*k^*}^*$, so that (4.20) is again satisfied. Likewise, for Branching Rule C, the bisection of the largest interval leads to all interval lengths approaching zero, thereby yielding $z_{jk}^* = \ln(y_{jk}^*)$, $\forall j \in J, k \in K_j$, so that the remainder of the proof of convergence continues to hold true. For Branching Rule D, observe that in the proof of Theorem 4.1, if the first case of (4.25) occurs infinitely often, then the corresponding interval length for $y_{j^*k^*}$ approaches zero, *i.e.*, $l_{j^*k^*}^* = u_{j^*k^*}^*$, so that (4.20) is satisfied. Otherwise, the second case in (4.25) occurs infinitely often, again leading to (4.20) being satisfied as in the proof of Theorem 4.1 itself. In either case, the remainder of the proof holds true identically. This completes the proof. \square

Remark 4.2. In our computations, we will investigate the relative merits of each of the foregoing four proposed branching rules A, B, C, and D. Other similar partitioning rules

that support (4.20) or, more generally, which imply in the proof of Theorem 4.1 that $z_{jk}^* = \ln(y_{jk}^*)$, $\forall j \in J, k \in K_j$, would also yield a theoretically convergent algorithmic procedure.

4.3. Computational Case Study

To illustrate the proposed approach, consider the following hypothetical case study. Assume that a mid-size city has been struck by a major tornado, affecting the residential and commercial areas. Specifically, say that an office building has collapsed, major fires have broken out in the downtown area, flooding has occurred due to a water main getting severed, and there is a power loss in the city. The emergency manager needs to quickly deploy the fire-fighting service, the police, the rescue squads, and the emergency medical units to mitigate the hazards that have occurred. Tables 4.1 and 4.2 show the hazard incidence matrix, and the response relevance matrix that assimilate the above information. Specifically, Table 4.1 provides information about the type of hazard faced in each area, while Table 4.2 identifies the emergency response units required to mitigate each hazard in the two areas under consideration.

Hazard Area	Collapse	Fire	Flood	Power Loss
Residential Area	-	✓	✓	✓
Commercial Area	✓	✓	-	-

Table 4.1: Hazard incidence matrix.

Hazard Resource	Collapse	Fire	Flood	Power Loss
Police	✓	✓	✓	✓
Firefighting	✓	✓	-	✓
Rescue	✓	✓	✓	-
Medical	✓	✓	✓	-

Table 4.2: Response relevance matrix.

Based on the above information, the emergency manager has decided on the values for the ratings, r_{jk} , which reflect the relative importance of addressing a specific hazard for each hazard-area combination. The factors that affect these ratings are the extent of damage, the risk to life and limb, and the potential of a situation to get more aggravated. Table 4.3 quantitatively denotes these ratings, which lie between 1 (least critical) and 10 (most critical) and are based on the severity of the situation.

Hazard Area	Collapse	Fire	Flood	Power Loss
Residential Area	-	10	6	3
Commercial Area	10	8	-	-

Table 4.3: Hazard-area ratings matrix (r_{jk} -values).

Also, the effectiveness of each considered emergency response resource in dealing with each type of hazard is displayed in Table 4.4. These effectiveness measures are based on a three level scale as low, medium, and high (L, M, and H). In our proposed model, these prescriptions correspond numerically to the β_{ik} values of $1.0 \lambda_i$, $1.5 \lambda_i$, and $2.0 \lambda_i$, respectively. Here, λ_i is such that with a full allocation of $x_{ijk} = b_i$ resource units, and an H-evaluation ($\beta_{ik} = 2.0 \lambda_i$), the attenuation $e^{-\beta_{ik} x_{ijk}} = (0.01)^{1/|I|}$, *i.e.*, $e^{-2\lambda_i b_i} = (0.01)^{1/4}$. This yields the values of λ_i , $\forall i$, and consequently the values of β_{ik} , $\forall (i, k)$. Hence, if all the resources were ascribed to a particular hazard in some area, if at all possible, and assuming the greatest effectiveness, we would achieve an overall extent of attenuation equal to $e^{-\sum_{i \in I} \beta_{ik} x_{ijk}} = 0.01$, *i.e.*, a reduction to 1% of the unmitigated risk level. Table 4.5 presents the total availability of emergency response teams, b_i (in appropriate units), for each of the four categories. Furthermore, Table 4.6 displays the minimum value of each emergency response resource that needs to be allocated for each hazard in each area as a proportion of the total availability, given the hazard incidence matrix. Combining this information with that given in Table 4.5, the values for the lower bounds

on the decision variables (L_{ijk} – values) can be obtained by multiplying the proportions in Table 4.6 by the corresponding resource availabilities (b_i -values) in Table 4.5, as specified within parentheses in Table 4.6. Finally, the value of the unmitigated risk, α_{jk} , was taken to be equal to 1000 for each hazard-area combination, *i.e.*, $\alpha_{jk} = 1000, \forall (j, k)$.

Hazard Resource	Collapse	Fire	Flood	Power Loss
Police	M	M	L	L
Firefighting	M	H	-	L
Rescue	H	M	M	-
Medical	H	H	L	-

Table 4.4: Measures relating to attenuation factors (β_{ik} -values) of $1.0 \lambda_i$, $1.5 \lambda_i$, and $2.0 \lambda_i$ for L, M, and H, respectively.

Resource	# of Units
Police Officers	110
Fire Companies	12
Rescue Teams	7
Medical Teams	20

Table 4.5: Total resource unit availability (b_i -values).

Hazard-Area Resource	Collapse	Fire	Fire	Flood	Power Loss	Total
	C	R	C	R	R	
Police	0.20 (22)	0.10 (11)	0.10 (11)	0.05 (4.5)	0.05 (4.5)	0.50 (45)
Firefighting	0.30 (3.6)	0.15 (1.8)	0.15 (1.8)	-	0.05 (0.6)	0.65 (7.8)
Rescue	0.30 (2.1)	0.10 (0.7)	0.15 (1.05)	0.10 (0.7)	-	0.65 (4.55)
Medical	0.25 (4)	0.05 (1)	0.05 (1)	0.10 (2)	-	0.45 (9)

C = Commercial Area; R = Residential Area

Table 4.6: Minimum resource assignments as a proportion of total availability

Tables 4.1 through 4.6, therefore provide the input data for the associated problem ERM. For the purpose of illustration, Problem ERM(l^0, u^0), given by (4.9a - 4.9j) and (4.8b), was first solved using the GAMS/BARON software (version 2.5, refer Sahinidis, 1996, Sahinidis, 1999-2000), for two different instances pertaining to $\mu_D = 0$ and $\mu_D = 100$, respectively. Obviously, the case of $\mu_D = 0$ corresponds to the instance where there is no emphasis on equity, and the objective function is dominated by minimizing the overall system weighted mitigated risk. The other extreme of $\mu_D = 100$ represents a strong emphasis on achieving equity among the respective hazard affected areas. Tables 4.7 and 4.8 display the resource assignments (x_{ijk} -values) obtained by solving Problem ERM(l^0, u^0), when μ_D takes on the values of 0 and 100, respectively.

Hazard-Area Resource	Collapse	Fire	Fire	Flood	Power Loss	Total
	C	R	C	R	R	
Police	33.66	42.422	22.88	5.5	5.538	110
Firefighting	3.6	3.508	4.292	-	0.6	12
Rescue	2.51	0.7	1.05	2.74	-	7
Medical	5.0	8.141	4.859	2.0	-	20
Objective term values	$R = 15590.9725, \sum_{j \in J} \gamma_j - \bar{\gamma} = 0.184, \max_{j \in J} \{\gamma_j - \bar{\gamma}\} = 0.092$					

Table 4.7: Resource assignments for each hazard in each area (x_{ijk} -values) corresponding to $\mu_D = 0$ obtained by solving ERM(l^0, u^0) using BARON.

Hazard-Area Resource	Collapse	Fire	Fire	Flood	Power Loss	Total
	C	R	C	R	R	
Police	33.874	42.377	22.711	5.5	5.538	110
Firefighting	3.6	3.524	4.276	-	0.6	12
Rescue	2.429	0.7	1.05	2.821	-	7
Medical	5.0	8.295	4.705	2.0	-	20
Objective term values	$R = 15590.7206, \sum_{j \in J} \gamma_j - \bar{\gamma} = 0.176, \max_{j \in J} \{\gamma_j - \bar{\gamma}\} = 0.088$					

Table 4.8: Resource assignments for each hazard in each area (x_{ijk} -values) corresponding to $\mu_D = 100$ obtained by solving ERM(l^0, u^0) using BARON.

For the purpose of comparison, the above two problem instances corresponding to $\mu_D = 0$ and $\mu_D = 100$ were also solved by applying BARON to the model ERM as directly given by (4.6a)-(4.6h). Tables 4.9 and 4.10 respectively display the resource assignments pertaining to $\mu_D = 0$ and $\mu_D = 100$.

Hazard-Area Resource	Collapse	Fire	Fire	Flood	Power Loss	Total
	C	R	C	R	R	
Police	22	51.651	25.349	5.5	5.5	110
Firefighting	3.6	1.943	5.857	-	0.6	12
Rescue	2.509	0.7	1.05	2.741	-	7
Medical	6.582	9.499	1.919	2.0	-	20
Objective term values	$R = 15590.6556, \sum_{j \in J} \gamma_j - \bar{\gamma} = 0.063, \max_{j \in J} \{\gamma_j - \bar{\gamma}\} = 0.032$					

Table 4.9: Resource assignments for each hazard in each area (x_{ijk} -values) corresponding to $\mu_D = 0$ obtained by solving ERM using BARON.

Hazard-Area Resource	Collapse	Fire	Fire	Flood	Power Loss	Total
	C	R	C	R	R	
Police	22	66.0	11	5.5	5.5	110
Firefighting	3.6	5.889	1.911	-	0.6	12
Rescue	2.497	0.7	1.05	2.753	-	7
Medical	6.595	1.0	10.405	2.0	-	20
Objective term values	$R = 15590.8071, \sum_{j \in J} \gamma_j - \bar{\gamma} = 0.062, \max_{j \in J} \{\gamma_j - \bar{\gamma}\} = 0.031$					

Table 4.10: Resource assignments for each hazard in each area (x_{ijk} -values) corresponding to $\mu_D = 100$ obtained by solving ERM using BARON.

To test the computational efficiency of our proposed methodology in obtaining the global optimum, and to establish the robustness of our algorithmic procedure as compared with the optimization strategy utilized by BARON, the data obtained from Tables 4.1 through 4.6 was further used to solve problem ERM via the branch-and-bound algorithm of Section 4.3. In this context, the branching rules A, B, C, and D were

implemented using a combination of CPLEX 8.1.0 and a code developed in C++. The (global) optimal resource allocations obtained by our algorithmic strategy, corresponding to $\mu_D = 0$ and $\mu_D = 100$, are displayed in Tables 4.11 and 4.12. The objective function value (with $\mu_D = 0$) pertaining to the global optimum is given by 13587.8717. The

Hazard-Area \ Resource	Collapse	Fire	Fire	Flood	Power Loss	Total
	C	R	C	R	R	
Police	22	30.481	46.487	5.5	5.532	110
Firefighting	3.6	1.8	6.0	-	0.6	12
Rescue	2.10	0.7	1.05	3.15	-	7
Medical	7.803	9.197	1.0	2.0	-	20
Objective term values	$R = 13587.8717, \sum_{j \in J} \gamma_j - \bar{\gamma} = 0.204, \max_{j \in J} \{\gamma_j - \bar{\gamma}\} = 0.102$					

Table 4.11: Global optimal resource assignments for each hazard in each area (x_{ijk} -values) corresponding to $\mu_D = 0$ obtained by solving ERM using the proposed algorithm.

Hazard-Area \ Resource	Collapse	Fire	Fire	Flood	Power Loss	Total
	C	R	C	R	R	
Police	22	18.232	46.487	17.75	5.532	110
Firefighting	3.6	1.8	6.0	-	0.6	12
Rescue	2.10	0.7	1.05	3.15	-	7
Medical	6.144	10.586	1.0	2.0	-	20
Objective term values	$R = 13594.84, \sum_{j \in J} \gamma_j - \bar{\gamma} = 0.170, \max_{j \in J} \{\gamma_j - \bar{\gamma}\} = 0.085$					

Table 4.12: Global optimal resource assignments for each hazard in each area (x_{ijk} -values) corresponding to $\mu_D = 100$ obtained by solving ERM using the proposed algorithm.

tightness of the LP relaxation, $RERM(l, u)$, can be gauged by comparing this optimal value with the lower and upper bounds obtained at the initial node. The lower bound at node 0 was 11468.6020, and the corresponding upper bound (obtained by substituting the x -part of the solution to Problem $RERM(l^0, u^0)$ into (4.5)) was found to be 22855.7789. Similarly, for the case of $\mu_D = 100$, the optimal value is 13628.8397, and the lower and

upper bounds obtained at node 0 were 11501.2130 and 22916.9531, respectively. Furthermore, note that the equity terms in Table 4.12 (corresponding to $\mu_D = 100$) are smaller as compared to those in Table 4.11 (corresponding to $\mu_D = 0$). Obviously, as the emphasis on equity increases, the overall risk attenuation factors for all the areas under consideration tend towards the mean weighted attenuation factor, thereby decreasing the total inequity as well the maximum inequity spread.

The number of nodes enumerated along with the CPU times taken to solve the case study example for $\mu_D = 0$ with each of the branching strategies, implemented using optimality tolerance values (ε) of 0.05, 0.01, and 10^{-6} , are displayed in Table 4.13. These computational results show that the branching rules, ordered in decreasing level of performance, are given by A, D, C, B. Since the convergence of the proposed algorithm is essentially involved with the process of driving the variables to achieve their bounds, the most successful branching strategy would be that which guides this propensity in the most efficient way by quickly creating partitions at optimal values. Obviously, branching rule A is based mainly on this construct, whereas branching rule C is the most oblivious to it. Likewise, branching rule B does not make use of the information regarding the tendency of the optimal solution while splitting the intervals, although it selects the branching variable index similar to rule A. Note that, Rule D generalizes Rule A in the sense that if the parameter 0.9 in (4.25) is reduced below 0.5, then this rule coincides with Rule A. Table 4.13 also presents results for Rule D, where the parameter 0.9 is replaced by 0.8, 0.7, and 0.6, as noted respectively in the last three rows of the table. Observe that Rule D improves in performance as it becomes closer to Rule A with a decrease in the stated parameter value. On average, branching rule A was able to converge to an optimal solution 14.57% faster than branching rule D(0.6), which was the next best rule. Similar results for the CPU times were obtained for the case corresponding to $\mu_D = 100$, and hence these results are not shown here.

For the purpose of comparison, the relative percentage deviations of the total objective function values obtained by GAMS/BARON from the optimal values derived by our algorithm (with $\varepsilon = 10^{-6}$) are presented in Table 4.14. On average, the objective

Branching Strategy	Optimality Tolerance						CPU Time Averages
	$\varepsilon = 0.05$		$\varepsilon = 0.01$		$\varepsilon = 10^{-6}$		
	# Nodes	CPU Time(s)	# Nodes	CPU Time(s)	# Nodes	CPU Time(s)	
A	9	0.0225	14	0.0551	24	0.1240	0.0672
B	15	0.0558	19	0.0655	39	0.1655	0.0956
C	13	0.0551	18	0.0655	39	0.1655	0.0953
D	14	0.0600	18	0.0655	28	0.1350	0.0868
D (0.8)	12	0.0551	19	0.0655	28	0.1350	0.0852
D (0.7)	12	0.0551	15	0.0559	25	0.1288	0.0799
D (0.6)	12	0.0551	15	0.0559	24	0.1250	0.0786

Table 4.13: Computational results obtained for comparing the various branching strategies.

function values obtained from the BARON solutions exhibit a 14.66% deviation from the global optimal value obtained via the proposed algorithm. From Tables 4.7 - 4.10, observe that the solution of problems ERM and $ERM(l^0, u^0)$ using BARON results in considerably different resource allocations, and these solutions and their objective values differ significantly from the global optimal solutions as displayed in Tables 4.11 and 4.12, for the cases corresponding to $\mu_D = 0$ and $\mu_D = 100$, respectively.

Problem	Value of parameter μ_D	
	$\mu_D = 0$	$\mu_D = 100$
ERM	15590.6556 (14.744%)	15603.2071 (14.497%)
ERM(l, u)	15590.9725 (14.744%)	15626.9206 (14.670%)

Table 4.14: Optimal objective function values obtained by solving problems ERM and ERM(l, u) using BARON along with the percentage deviations from the optimal value found using the

The ability of the proposed algorithm to consistently yield better solutions than BARON is due to the robustness of the LP-based bounds as prescribed by the proposed algorithm. In contrast, BARON employs convex bounding problems that are solved by

the commercial nonlinear programming software, CONOPT. Any inaccuracies in computing the resulting lower bounds could lead to a false fathoming of nodes in the enumeration tree, as appears to be the case in the present context.

In order to further support the utility of the proposed methodology, we compared its results against that obtained by applying an ad-hoc intuitive procedure that might be used by an emergency manager in such a context. For this purpose, because judgments with respect to equity can be difficult to prescribe without considerable trial-and-error attempts, we considered the simple case of $\mu_D = 0$. The intuitively appealing resource allocation scheme is as follows. First, we initialized $x_{ijk} = L_{ijk}, \forall i \in I, j \in J, k \in K_j$, and computed the residual resources $\bar{b}_i = b_i - \sum_{j \in J} \sum_{k \in K_j} L_{ijk}, \forall i \in I$, and the current value

of the attenuated risk $R_{jk} \equiv r_{jk} \alpha_{jk} e^{-\sum_{i \in I} \beta_{ik} x_{ijk}}, \forall j \in J, k \in K_j$, for $x_{ijk} = L_{ijk}, \forall (i, j, k)$.

Next, considering the resources in increasing order of $\bar{b}_i, i \in I$, we allocated these resources 0.1 units at a time until depleted (the final allocation being possibly fractional), each time allocating a unit of the particular resource to that combination (j, k) , $j \in J, k \in K_j$, for which the corresponding decrease in the R_{jk} -value would be a maximum. This methodology was also implemented in C++, and the resulting resource allocations are given in Table 4.15, along with the final R -value and corresponding equity characteristics obtained. Comparing this solution with the optimal solution given in Table 4.11, we see that the optimal solution is 17.4% superior (lower) in terms of the objective function value, and moreover, the CPU time required by the ad-hoc method was 0.0571 seconds, which is not substantially lower than that required by the optimization scheme. Thus, for a relatively small increase in computational effort, we can achieve a far better solution via the prescribed optimization methodology. Indeed, note that the solution obtained via the ad-hoc method is relatively inferior when compared to the sub-optimal BARON solution itself.

Another possible intuitive alternative would be to allocate, at each step, 0.1 extra units of a resource (until all are depleted), which decreases an R_{jk} -value by a maximum amount, instead of doing this one resource at a time, as in the above scheme. The

resulting resource allocation obtained via this second ad-hoc intuitive scheme was found to be similar to the one perceived for the first ad-hoc method, and is therefore not included here.

Hazard-Area Resource	Collapse	Fire	Fire	Flood	Power Loss	Total
	C	R	C	R	R	
Police	30.9	20.2	20.5	32.9	5.5	110
Firefighting	3.6	4.3	3.5	-	0.6	12
Rescue	2.10	2.7	1.5	0.7	-	7
Medical	6.7	5.7	5.6	2.0	-	20
Objective term values	$R = 15951.74, \sum_{j \in J} \gamma_j - \bar{\gamma} = 0.2089, \max_{j \in J} \{\gamma_j - \bar{\gamma}\} = 0.1045$					

Table 4.15: Resource assignments for each hazard in each area (x_{ijk} -values) corresponding to $\mu_D = 0$ obtained by solving ERM using an ad-hoc intuitive algorithm.

The sensitivity of the objective function value to the parameter μ_D was examined next. Figure 4.2 depicts the variation in the objective function value corresponding to the overall system mitigated risk (R), as well as the equity terms ($\sum_{j \in J} \eta_j + |J| \eta$) without being multiplied by μ_D (as shown atop the bar graphs corresponding to each parameter value), for various values of the parameter μ_D . Observing the trend, it can be seen that the R -value tends to be nearly a constant for $\mu_D \in [0, 250]$ or so, and then rapidly increases as the value of μ_D increases further, *i.e.*, when a relatively larger emphasis is placed on equity, leading to inefficient overall resource assignments for the sake of achieving a greater degree of fairness.

Considering problem $ERM(l^0, u^0)$, the second and third terms in the objective function can be viewed as constraints that have been dualized by the Lagrange multipliers, μ_D and μ_R , respectively, *i.e.*, problem $ERM(l^0, u^0)$ can be equivalently viewed as: Minimize $\sum_{j \in J} \sum_{k \in K_j} r_{jk} \alpha_{jk} y_{jk}$, subject to (4.9a-5.9j) along with $\sum_{j \in J} \eta_j \leq \theta_1$ and $\eta \leq \theta_2$, where θ_1 and θ_2 are respectively the values of $\sum_{j \in J} \eta_j$ and η for the

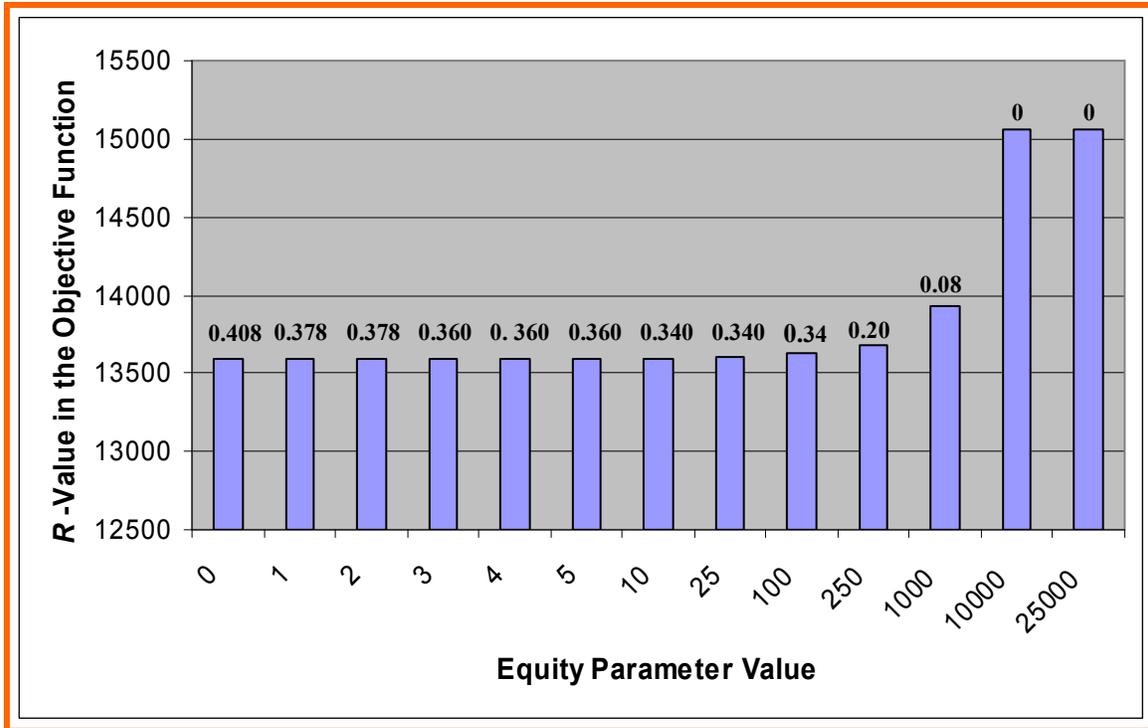


Figure 4.2: Sensitivity of the R -term and the equity terms in the objective function to variations in the equity parameter μ_D .

optimal solution obtained from problem $ERM(l^0, u^0)$. (In this case, $-\mu_D$ and $-\mu_R$ are the respective Lagrange multipliers associated with these constraints at optimality.) Thus, requiring a greater degree of equity can be viewed as imposing a corresponding tighter constraint, resulting in an increase in the overall system weighted mitigated risk R as μ_D increases. However, from Figure 4.2, it can be inferred that these added constraints affect the overall system risk R significantly only for relatively larger values of μ_D exceeding 1000. The decision-maker can view plots similar to that of Figure 4.2 to achieve the desired compromise between efficiency and equity.

4.4. Summary, Conclusions, and Extensions for Further Research

This chapter has focused on the problem of employing the available emergency response resources to mitigate risks that arise in the aftermath of a natural disaster, terrorist attack, or any other unforeseen calamity. Given several areas that might be affected, the emergency manager is faced with the issue of not only mitigating the

hazards that might have occurred, but to also ensure that equity among all the affected regions is achieved. Following the conventional definition of risk, this problem scenario was modeled to achieve any desired level of compromise between the overall system weighted mitigated risk, and equity with respect to deviations from the mean of the risk attenuation factors attained for the different affected areas. The resulting nonconvex program was solved through a suitable transformation and polyhedral outer approximation process that was used in concert with a specialized branch-and-bound procedure. The developed algorithm was proven to converge to a global optimal solution. Various alternative branching strategies that preserve the convergence characteristics of the algorithm were also proposed. Computational results obtained by solving a hypothetical case scenario were presented, and variations in the solution and algorithmic performance with respect to the objective equity parameter and the alternative branching strategies were investigated. In particular, the proposed algorithm was demonstrated to more robustly yield global optimal solutions in comparison with the commercial global optimizer BARON, as well as an ad-hoc intuitive method.

There are several variations and extensions of this work that could be considered for future research. For example, following Sherali and Subramanian (1999), we could accommodate risks associated with potential hazards that might yet occur with some probabilities while dispatching emergency response resources to address the presently existing hazards. The model could also be extended to scenarios such as emergencies arising in hospitals or medical/refugee units in war-devastated regions or homeland security scenarios. In some of these contexts, one might need to contend with allocating resources to a cascading sequence of catastrophic events. The scope of the problem of allocating emergency response resources to minimize risk is broad and offers a rich area for modeling and analytical research.

5. Cascading Risk Management Using an Event Tree Optimization Approach

One of the primary challenges of risk management relates to the issue of minimizing the loss that might occur as a result of a series of hazardous events. Specifically, system safety is the area of risk management that deals with identifying the hazards that can potentially infect a given system and assessing the risk that these hazards can inflict. Among the methods most commonly used to assess the risk of system failure is *decision tree* analysis. In particular, two types of decision trees, namely, *fault trees* and *event trees* are widely used in practice. A significant difference in the two is that the basis for fault trees lies in *deductive* (backward) logic, whereas event trees usually conform to *inductive* (forward) logic.

Event tree analysis deals with identifying the consequences resulting from a causative event via a forward logic routine. An event tree begins with an *initiating event*, could either be a component failure within the system under study or the result of an external cause. This event is represented as node zero in the event tree. Beginning with the initiating event, the safety features inherent within the system, which are triggered in a cascading fashion, define the nodes of the event tree. Note that a particular safety feature can be associated with possibly several nodes, and these features are represented in the order in which they are activated to counteract the hazard that has occurred. Each one of these nodes is generated by some success or failure state, which is associated with a unique link in the event tree diagram. Once enumerated, these success and failure states (links) give rise to various event sequences that can possibly occur due to the initiating event. Every alternating action-state sequence corresponds to a unique chain in the event tree diagram. Given an event tree with v safety features, and a two-state mechanism (representing success and failure links) which produces binary event trees (or Bernoulli event trees), the total number of outcomes is of the order $O(2^v)$ (or $O(m^v)$ for a generic event tree with an m -state mechanism). Figure 5.1 displays an event tree for a gas-line rupture situation, where, at each node, the decision taken to interpose a safety feature or ameliorating action can lead to one of two immediate scenarios (based on the success or failure of this action), each of which then continues to cascade through the event tree.

Each event chain culminates in some end node of the tree, which entails a specific *consequence* (loss). As an aside, note that while we focus on a single initiating event for the sake of simplicity in presentation, the case of multiple initiating events can be handled in a fashion analogous to that described below.

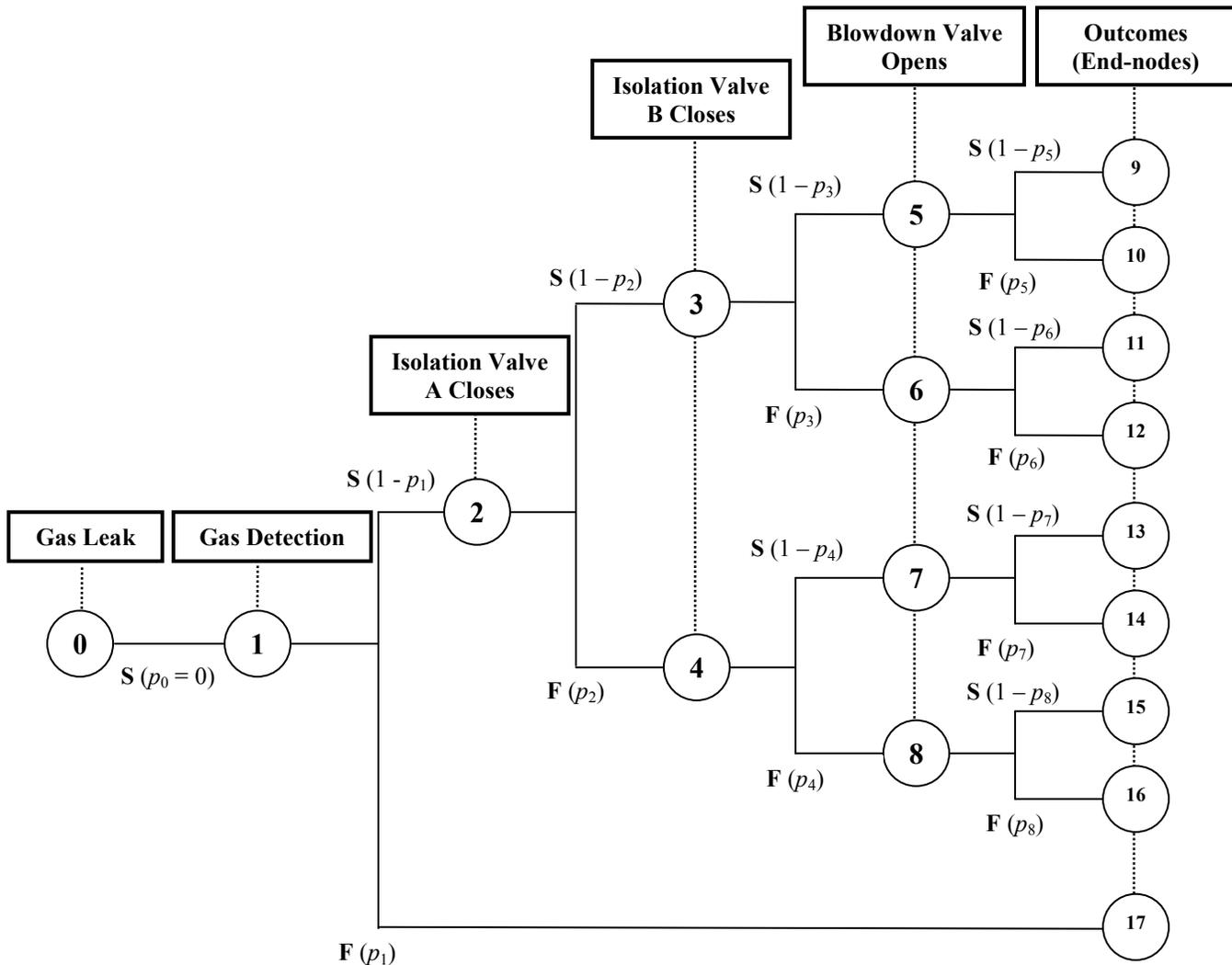


Figure 5.1: Illustration of a binary event tree depicting the occurrence of cascading risk events initiated by a gas-line rupture (refer Andrews and Dunnett, 2000).

The use of event trees to analyze system failures began in the 1970s, when the US Nuclear Regulatory Commission performed risk assessment tests in nuclear power plants (see Rasmussen, 1975). Since then, event tree analysis has been used to study system risk in various contexts arising in both the public and private sectors. These studies include steam generator tube ruptures (Zhang and Yan, 1999), water resource planning (Beim and

Hobbs, 1997), fusion-fission hybrid reactor failures (Yang and Qiu, 1993), electrical accident counter-measure systems for mines (Collins and Cooley, 1983), failure of temporary structures (Hadipriono, *et al.*, 1986), reliability analysis of high voltage transmission systems (Ohba *et al.*, 1984), and emergency response in the event of chemical hazards or spills (Raghu, 2004, and Zhang *et al.*, 2004).

Once an entire event tree is constructed, as illustrated in Figure 5.1, the principal task lies in computing the *risk* (probability of occurrence times the consequence) associated with each outcome (end node) of the event tree. For independent events, the probability of occurrence of a particular outcome is the product of the probabilities associated with the links that lie on the unique chain connecting the initiating event to the corresponding end node. Most existing case studies, including the ones mentioned above, deal mainly with the development of the event tree and the derivation of the associated probabilities and consequences as prompted by the specific application, along with the computation of the risk or expected consequence. Some notable quantitative approaches that specifically address the composition of event trees and their associated data include the work of Takaragi *et al.* (1983), wherein by using minimum cut/prime implicant sets, a few basic events are eliminated from the event tree and an upper-bounding approximate computation of the failure probabilities is prescribed. Based on this work, several modifications and extensions have appeared in the literature. Using mostly binary decision diagrams, Sinnamon and Andrews (1996, 1997a, b) and Andrews and Dunnett (2000) have proposed several quantitative and qualitative approaches for calculating the (conditional) link probabilities and the total expected loss in fault trees and event trees, particularly for those cases when the events are not independent. Some related research is also discussed in Rauzy (1993, 1996). By viewing event trees in terms of transition matrices that evolve from an entry state to exit states and following certain logical arguments, Kaplan (1982) initiated matrix theory formalisms for event tree analysis and provided related conceptual and computational insights. Unwin (1984) provided a compact numerical representation of the different scenarios that occur in an event tree in order to reduce computer memory requirements. In addition, certain fuzzy set-based event tree analyses have been reported in Huang *et al.* (2001), Kenarangui (1991), Jin *et al.* (2003), and Patra *et al.* (1995). However, none of these papers deal with the *strategic*

planning idea of allocating a given set of available resources to control the event's success and failure probabilities and to mitigate the possible final consequences as a way of reducing the ensuing risk. The present paper fills this void by providing a novel modeling and algorithmic approach for such a strategic planning decision problem.

More specifically, in this research effort our primary concern is to develop the basis for a strategic planning decision-support system that can help the parties who prepare for and manage emergency safety situations reduce the risk associated with a given scenario. For example, in planning to reduce the risk associated with a gas leak, the available preventive resources could be used to reduce the failure likelihood of the different safety features, such as leak detection, closing of critical valves, *etc.* (see Figure 5.1). Likewise, the available mitigation resources could be brought to bear to ameliorate the potential consequences associated with the end nodes of the decision tree. In such a situation, the goal would be to achieve the most effective deployment of the limited available resources by manipulating the event probabilities and the resultant consequences associated with the event tree in order to minimize the overall risk.

Accordingly, let $i \in \{1, \dots, I\}$ index the set of nodes (*decision points*) at which some safety feature is deployed in a binary event tree, and let p_i and $(1 - p_i)$ respectively denote the probabilities of failure and success associated with the outcome resulting from decision i . (Although we model this problem for binary event trees, this approach can easily be extended to generic event trees as well.) Now, suppose that we are given a set of preventive resources $m \in \{1, \dots, M\}$ that can be applied to control the outcome probabilities. Specifically, let q_{im} be the quantity of preventive resource m that is allocated to reduce the associated (failure) probability p_i of decision i . Using traditional logit-model theory, we assume that the logit function for p_i is linearly related to the allocations of the preventive resources, *i.e.*,

$$\ln \left[\frac{p_i}{1 - p_i} \right] = a_{i0} - \sum_{m=1}^M a_{im} q_{im}, \quad \forall i = 1, \dots, I,$$

where, a_{i0}, \dots, a_{iM} are nonnegative constants associated with the logit choice model. Moreover, let the set of final consequences (end-nodes of the event tree) be indexed by

the set $j \in \tau$, and let l_j denote the loss magnitude associated with end-node j . As before, suppose that we are given a set of mitigation resources $n \in \{1, \dots, N\}$, and let r_{jn} be the quantity of mitigation resource n that is allocated to reduce the loss magnitude l_j of consequence j . We assume that the magnitude of consequence j is linearly related to the allocation of all mitigation resources, *i.e.*,

$$l_j = b_{j0} - \sum_{n=1}^N b_{jn} r_{jn}, \quad \forall j \in \tau,$$

where, b_{j0}, \dots, b_{jN} are given nonnegative constants. Furthermore, let s_m and t_n denote the magnitudes of the total available units of preventive resource m and mitigation resource n , respectively. Finally, let c_{im} , d_{jn} , and β represent the per-unit cost of allocating preventive resource m to node i in order to reduce p_i , the per-unit cost of allocating mitigating resource n to end-node j in order to reduce l_j , and the total available budget, respectively. The *Event Tree Optimization* problem is then concerned with determining an optimal allocation of preventive and mitigation resources at the different decision points of the event tree so as to minimize the total expected loss, subject to the resource availability and budgetary restrictions.

The remainder of this chapter is organized as follows. In Section 5.1, we formulate a mathematical model for the event tree optimization problem and develop a tight representation for this problem through suitable transformations and polyhedral approximations. Section 5.1 describes two specialized branch-and-bound procedures to solve the formulated problem and provides theoretical proofs of convergence for these proposed algorithms. Some computational experience and evaluation of the alternative proposed algorithmic strategies are presented in Section 5.2 based on data pertaining to a hypothetical case study. Finally, Section 5.3 concludes the chapter with a summary and directions for future research.

5.1. Event Tree Optimization Model

Consider an event tree T rooted at the initiating event node 0, and let τ denote the set of terminal or end-nodes j , each associated with the corresponding loss l_j . For each

$j \in \tau$, let \mathcal{L}_j denote the set of nodes k lying on the chain from node j to the root node 0, excluding nodes 0 and j . Note that for each $k \in \mathcal{L}_j$, there exists an associated link, L_{jk} , say, which lies on the chain from node 0 to node j and that connects node k to one of its immediate successor nodes in T that belongs to this chain. Accordingly, let

$$S_{1j} = \{k \in \mathcal{L}_j : \text{the probability associated with link } L_{jk} \text{ is } p_k\}, \forall j \in \tau, \text{ and}$$

$$S_{2j} = \{k \in \mathcal{L}_j : \text{the probability associated with link } L_{jk} \text{ is } (1 - p_k)\}, \forall j \in \tau.$$

The Event Tree Optimization (**ETO**) problem can then be formulated as follows, where the bounds specified for defining Ω in (1g) are assumed to be given (or implied by the scenario considerations, *e.g.*, $l_j^u = b_{j0}$, $\forall j$, and $l_j^l = \max\left\{\varepsilon, b_{j0} - \sum_{n=1}^N b_{jn} t_n\right\}$, $\forall j$, where $\varepsilon > 0$).

$$\mathbf{ETO:} \quad \text{Minimize} \quad \sum_{j \in \tau} l_j \prod_{k \in S_{1j}} p_k \prod_{k \in S_{2j}} (1 - p_k) \quad (5.1a)$$

$$\text{subject to} \quad \sum_{i=1}^I q_{im} \leq s_m, \quad \forall m = 1, \dots, M \quad (5.1b)$$

$$\sum_{j \in \tau} r_{jn} \leq t_n, \quad \forall n = 1, \dots, N \quad (5.1c)$$

$$\sum_{i=1}^I \sum_{m=1}^M c_{im} q_{im} + \sum_{j \in \tau} \sum_{n=1}^N d_{jn} r_{jn} \leq \beta, \quad (5.1d)$$

$$\ln\left[\frac{p_i}{1 - p_i}\right] = a_{i0} - \sum_{m=1}^M a_{im} q_{im}, \quad \forall i = 1, \dots, I \quad (5.1e)$$

$$l_j = b_{j0} - \sum_{n=1}^N b_{jn} r_{jn}, \quad \forall j \in \tau \quad (5.1f)$$

$$(p, l) \in \Omega \equiv \left\{ \begin{array}{l} (p, l) : 0 < p_i^l \leq p_i \leq p_i^u < 1, \quad \forall i = 1, \dots, I \\ 0 < l_j^l \leq l_j \leq l_j^u < \infty, \quad \forall j \in \tau \end{array} \right\}, \quad (5.1g)$$

$$(r, q) \geq 0. \quad (5.1h)$$

In this formulation, the objective function seeks to minimize the total risk, constraints (5.1b) and (5.1c) impose the resource availability restrictions, constraint (5.1d) enforces

the budgetary limitation, constraints (5.1e) and (5.1f) follow from the assumptions based on the logit-choice model, and finally (5.1g) and (5.1h) require that the variables satisfy some specified bounding restrictions. Observe that the nonconvexity in Problem ETO arises due to the polynomial objective function (5.1a) and the logarithmic (factorable) term in constraint (5.1e).

Remark 5.1. Note that, in Problem ETO, we have assumed that the quantities of preventive and mitigation resources allocated, namely the q_{im} - and r_{jn} -values, are simply restricted by (5.1b, c, d, and h). However, in general, the effective preventive or resource allocations at each node might be functionally related to certain capital or manpower investments made in a set of pertinent improvement alternatives, and accordingly, the resource availability and budget restrictions would then apply to such investment decisions. The constraints (5.1b, c, d, and h) could then be more generally represented as $(r, q) \in \mathcal{P}$, where \mathcal{P} is a specified polytope. In such a case, nonetheless, the problem manipulations and algorithmic theory remain identical to that described in the sequel. For the sake of simplicity in presentation and for illustrative purposes, we will continue to use (5.1b, c, d, and h) below, with the understanding that these can be replaced by the more general relationship $(r, q) \in \mathcal{P}$ in our proposed methodology.

Let us now define the following auxiliary variables, along with their implied bounds, in order to conveniently reformulate problem ETO. To begin with, let us transform the objective function by denoting

$$\theta_j = l_j \prod_{k \in S_{1j}} p_k \prod_{k \in S_{2j}} (1 - p_k), \quad \forall j \in \tau. \quad (5.2a)$$

Then, we have,

$$\theta_j^l \leq \theta_j \leq \theta_j^u,$$

where,

$$\left\{ \begin{array}{l} \theta_j^l = l_j^l \prod_{k \in S_{1j}} p_k^l \prod_{k \in S_{2j}} (1 - p_k^u) \\ \theta_j^u = l_j^u \prod_{k \in S_{1j}} p_k^u \prod_{k \in S_{2j}} (1 - p_k^l) \end{array} \right\}, \quad \forall j \in \tau. \quad (5.2b)$$

Similarly, to linearize (5.1e), let us introduce the variables

$$y_{1i} = \ln(p_i) \text{ and } y_{2i} = \ln(1-p_i), \forall i=1, \dots, I. \quad (5.3a)$$

Note that,

$$y_{1i}^l \leq y_{1i} \leq y_{1i}^u \text{ and } y_{2i}^l \leq y_{2i} \leq y_{2i}^u, \quad \text{where,}$$

$$y_{1i}^l = \ln(p_i^l), y_{1i}^u = \ln(p_i^u), y_{2i}^l = \ln(1-p_i^l), \text{ and } y_{2i}^u = \ln(1-p_i^u), \forall i=1, \dots, I. \quad (5.3b)$$

Next, to linearize (5.2a) itself, let us denote

$$z_j = \ln(\theta_j), \quad \forall j \in \tau \quad (5.4a)$$

where, based on (5.2b), we can impose,

$$z_j^l \leq z_j \leq z_j^u, \text{ where, } z_j^l = \ln(\theta_j^l) \text{ and } z_j^u = \ln(\theta_j^u), \forall j \in \tau. \quad (5.4b)$$

Likewise, to accommodate the term $\ln(l_j)$ generated by taking logarithms in (5.2a), define,

$$\xi_j = \ln(l_j), \quad \forall j \in \tau \quad (5.5a)$$

and impose the related bounds,

$$\xi_j^l \leq \xi_j \leq \xi_j^u, \text{ where, } \xi_j^l = \ln(l_j^l) \text{ and } \xi_j^u = \ln(l_j^u), \forall j \in \tau. \quad (5.5b)$$

This yields the following equivalently reformulated problem **ETO**(Ω), which is predicated on the hyperrectangle Ω .

$$\mathbf{ETO}(\Omega): \quad \text{Minimize} \quad \sum_{j \in \tau} \theta_j \quad (5.6a)$$

$$\text{subject to} \quad \sum_{i=1}^I q_{im} \leq s_m, \quad \forall m=1, \dots, M \quad (5.6b)$$

$$\sum_{j \in \tau} r_{jn} \leq t_n, \quad \forall n=1, \dots, N \quad (5.6c)$$

$$\sum_{i=1}^I \sum_{m=1}^M c_{im} q_{im} + \sum_{j \in \tau} \sum_{n=1}^N d_{jn} r_{jn} \leq \beta, \quad (5.6d)$$

$$y_{1i} - y_{2i} = a_{i0} - \sum_{m=1}^M a_{im} q_{im}, \quad \forall i=1, \dots, I \quad (5.6e)$$

$$l_j = b_{j0} - \sum_{n=1}^N b_{jn} r_{jn}, \quad \forall j \in \tau \quad (5.6f)$$

$$z_j = \xi_j + \sum_{k \in S_{1j}} y_{1k} + \sum_{k \in S_{2j}} y_{2k}, \quad \forall j \in \tau \quad (5.6g)$$

$$y_{1i} = \ln(p_i), \quad \forall i = 1, \dots, I \quad (5.6h)$$

$$y_{2i} = \ln(1 - p_i), \quad \forall i = 1, \dots, I \quad (5.6i)$$

$$z_j = \ln(\theta_j), \quad \forall j \in \tau \quad (5.6j)$$

$$\xi_j = \ln(l_j), \quad \forall j \in \tau \quad (5.6k)$$

$$(p, l) \in \Omega, \quad (5.6l)$$

$$(r, q) \geq 0, \quad (5.6m)$$

$$\left\{ \begin{array}{l} \theta_j^l \leq \theta_j \leq \theta_j^u, \quad z_j^l \leq z_j \leq z_j^u, \quad \text{and} \quad \xi_j^l \leq \xi_j \leq \xi_j^u, \quad \forall j \in \tau, \\ y_{1i}^l \leq y_{1i} \leq y_{1i}^u \quad \text{and} \quad y_{2i}^l \leq y_{2i} \leq y_{2i}^u, \quad \forall i = 1, \dots, I \end{array} \right\}, \quad (5.6n)$$

where Ω is as specified in (5.1g), and where the bounds in (5.6n) depend on Ω (even as Ω will be modified via a partitioning process in the sequel) and are given by (5.2b), (5.3b), (5.4b), and (5.5b), respectively. For the sake of convenience in the sequel, we shall denote the set of variables in Problem ETO(Ω), with obvious vector notation, as

$$x \equiv (p, l, r, q, \theta, y_1, y_2, z, \xi),$$

where, in particular, $y_1 = (y_{1i}, \forall i)$ and $y_2 = (y_{2i}, \forall i)$. Observe that ETO(Ω) is linear except for the complicating identities (5.6h) - (5.6k). In order to generate a linear programming relaxation (LP(Ω)), we shall construct polyhedral outer-approximations for each of these identities as follows.

In generic notation, which can then be applied to each of the identities in (5.6h) - (5.6k), consider the following relationship:

$$\lambda = \ln(\gamma), \quad \text{where} \quad 0 < \gamma^l \leq \gamma \leq \gamma^u < \infty. \quad (5.7)$$

Figure 5.2 illustrates this functional relationship. Here, the associated polyhedral approximation is constructed via the affine convex envelope of this concave function, along with four tangential supports (equispaced for the sake of simplicity), including, in particular, supports at the endpoints γ^l and γ^u . The resulting outer-approximation in the two-dimensional functional space for (5.7) is then given as follows.

$$\lambda \geq \ln(\gamma^l) + \frac{(\gamma - \gamma^l)}{(\gamma^u - \gamma^l)} [\ln(\gamma^u) - \ln(\gamma^l)], \quad (5.8a)$$

along with

$$\lambda \leq \ln(\gamma_w) + \frac{(\gamma - \gamma_w)}{\gamma_w}, \text{ for } \gamma_w = \gamma^l + \frac{(\gamma^u - \gamma^l)}{3}(w-1), w=1, \dots, 4. \quad (5.8b)$$

Hence, let $LP(\Omega)$ denote the linear program obtained from $ETO(\Omega)$ by replacing (5.6h) - (5.6k) with the approximating constraints of the type (5.8a, b). Then, the following results are evident, where $\nu[P]$ denotes the optimal value for any optimization problem P.

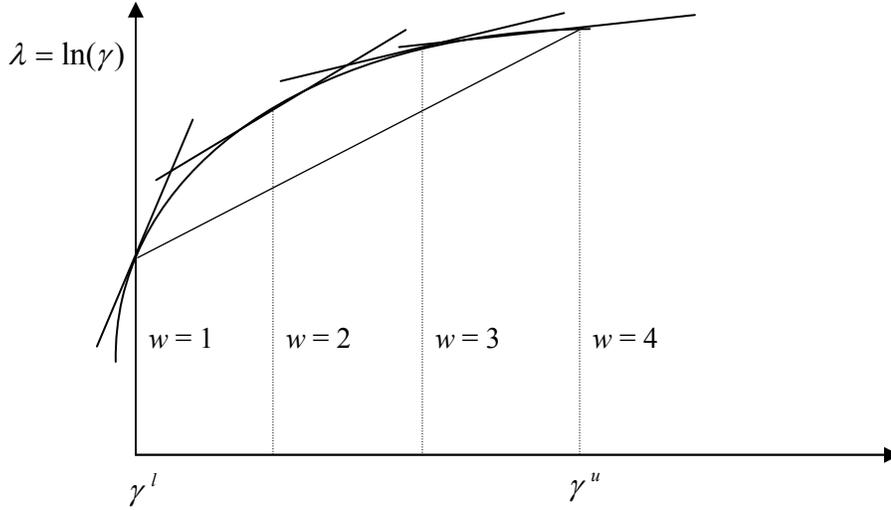


Figure 5.2: Illustration of the polyhedral outer-approximation strategy.

Lemma 5.1. $\nu[LP(\Omega)]$ provides a lower bound on $\nu[ETO(\Omega)]$. Moreover, if $\bar{x} = (\bar{p}, \bar{l}, \bar{r}, \bar{q}, \bar{\theta}, \bar{y}_1, \bar{y}_2, \bar{z}, \bar{\xi})$ solves $LP(\Omega)$ and if this solution satisfies the constraints (5.6h) - (5.6k), then it is also optimal to Problem $ETO(\Omega)$ with the same objective value.

Proof. Obvious from construction. □

Lemma 5.2. Let $(\bar{l}, \bar{r}, \bar{q})$ be part of an optimal solution to $LP(\Omega)$. Compute \hat{p} via

$$\hat{p}_i = \frac{\bar{g}_i}{1 + \bar{g}_i}, \forall i, \text{ where } \bar{g}_i \equiv e^{\left[a_{i0} - \sum_{m=1}^M a_{im} \bar{q}_{im} \right]}, \forall i. \quad (5.9)$$

Then, $(\hat{p}, \bar{l}, \bar{r}, \bar{q})$ is a feasible solution for Problem ETO and provides an upper bound \overline{UB} for this problem as given by

$$\overline{UB} = \sum_{j \in \tau} \bar{l}_j \prod_{k \in \mathcal{S}_{1j}} \hat{p}_k \prod_{k \in \mathcal{S}_{2j}} (1 - \hat{p}_k). \quad (5.10)$$

Proof. Note that given \bar{q} , the computation of \bar{p} via (5.9) satisfies (5.1e). All the other constraints of Problem ETO are satisfied by the solution $(\bar{p}, \bar{l}, \bar{r}, \bar{q})$ by virtue of being directly included within Problem LP(Ω). This completes the proof. \square

A key result that enables us to design the branch-and-bound algorithm proposed below and induces convergence to (global) optimality of this procedure is presented next.

Lemma 5.3. Let $\bar{x} = (\bar{p}, \bar{l}, \bar{r}, \bar{q}, \bar{\theta}, \bar{y}_1, \bar{y}_2, \bar{z}, \bar{\xi})$ be an optimal solution to Problem LP(Ω) having an objective function value $\nu[\text{LP}(\Omega)]$. Furthermore, suppose that each of \bar{p}_i equals one of its bounds p_i^l or p_i^u , $\forall i$, and similarly, each \bar{l}_j equals either l_j^l or l_j^u , $\forall j$, and each $\bar{\theta}_j$ equals either θ_j^l or θ_j^u , $\forall j$. Then \bar{x} also solves Problem ETO(Ω) with the same objective value.

Proof. Noting the construction of Problem LP(Ω) from Problem ETO(Ω), it is sufficient to show that the given conditions of the lemma imply that \bar{x} satisfies (5.6h) - (5.6k). Toward this end, consider the generic case of the identity (5.7) that has been approximated by the relationships (5.8a, b). It is easy to verify that if γ equals γ^l or γ^u , then (5.8a, b) together imply that $\lambda = \ln(\gamma)$. For instance, consider the case of $\gamma = \gamma^l$ (the case of $\gamma = \gamma^u$ is similar). Then, (5.8a) yields $\lambda \geq \ln(\gamma^l)$ and (5.8b) yields $\lambda \leq \ln(\gamma^l)$ for the inequality corresponding to $w = 1$, whence $\gamma_w = \gamma^l$, thereby leading to $\lambda = \ln(\gamma^l)$. Consequently, it directly follows that \bar{x} satisfies (5.6h) - (5.6k). This completes the proof. \square

Corollary 5.1. Let Ω be such that $p_i^l = p_i^u$, $\forall i$, and $l_j^l = l_j^u$, $\forall j$. Then, if \bar{x} solves LP(Ω), it also solves ETO(Ω) with the same objective value.

Proof. Note from (5.2b) that under the given condition of the corollary, we also have that $\theta_j^l = \theta_j^u$, $\forall j$. Hence, \bar{x} satisfies the hypothesis of Lemma 5.3, and this completes the proof. \square

Corollary 5.2. Let \bar{x} be an optimal solution to $\text{LP}(\Omega)$. If $\bar{p}_i = p_i^l$ or $\bar{p}_i = p_i^u$, for any $i \in \{1, \dots, I\}$, then $\bar{y}_{1i} = \ln(\bar{p}_i)$ and $\bar{y}_{2i} = \ln(1 - \bar{p}_i)$. Likewise, for any $j \in \tau$, if $\bar{\theta}_j = \theta_j^l$ or $\bar{\theta}_j = \theta_j^u$, then $\bar{z}_j = \ln(\bar{\theta}_j)$, and if $\bar{l}_j = l_j^l$ or $\bar{l}_j = l_j^u$, then $\bar{\xi}_j = \ln(\bar{l}_j)$.

Proof. Evident from the proof of Lemma 5.3. \square

Lemma 5.3 and its corollaries prompt two alternative approaches to solve Problem ETO via $\text{ETO}(\Omega)$ and its relaxation $\text{LP}(\Omega)$. In the first approach, referred to as **Algorithm A**, we adopt a branch-and-bound process based on partitioning the hyperrectangle Ω . Given any node sub-problem $\text{ETO}(\Omega)$ associated with a particular Ω , where the bounds on the variables $(\theta, y_1, y_2, z, \xi)$ are correspondingly given by (5.2b), (5.3b), (5.4b), and (5.5b), respectively, we construct the relaxation $\text{LP}(\Omega)$, and solve this linear program. Let $\bar{x} = (\bar{p}, \bar{l}, \bar{r}, \bar{q}, \bar{\theta}, \bar{y}_1, \bar{y}_2, \bar{z}, \bar{\xi})$ be the optimal solution obtained for $\text{LP}(\Omega)$. If the condition of Lemma 5.1 holds true, then we will also have solved the node subproblem $\text{ETO}(\Omega)$. Otherwise, we apply Lemma 5.2 to possibly update the incumbent solution, and as necessary, we branch at this node by partitioning Ω as follows.

Branching Rule A. Find $\max\{(p_i^u - p_i^l), \forall i, (l_j^u - l_j^l), \forall j\}$, and bisect the interval of the corresponding variable that achieves this maximum value, breaking ties in favor of the variable that yields the largest value of the discrepancy index

$$\max\left\{\left|y_{1i} - \ln(\bar{p}_i)\right|, \left|y_{2i} - \ln(1 - \bar{p}_i)\right|, \forall i = 1, \dots, I, \text{ and } \left|\bar{\xi}_j - \ln(\bar{l}_j)\right|, \forall j \in \tau\right\}.$$

In an alternative approach, referred to as **Algorithm B**, we also include the variables θ_j in the partitioning scheme. Here, let us define

$$\Omega' \equiv \{(p, l, \theta) : p_i^l \leq p_i \leq p_i^u, \forall i, l_j^l \leq l_j \leq l_j^u, \forall j, \theta_j^l \leq \theta_j \leq \theta_j^u, \forall j\}, \quad (5.11a)$$

where, given any bounds on (p, l) , we adjust the specified bounds on θ based on (5.2b) according to

$$\left[\begin{array}{l} \theta_j^l \leftarrow \max \left\{ \theta_j^l, l_j^l \prod_{k \in S_{1j}} p_k^l \prod_{k \in S_{2j}} (1 - p_k^u) \right\}, \forall j \\ \theta_j^u \leftarrow \min \left\{ \theta_j^u, l_j^u \prod_{k \in S_{1j}} p_k^u \prod_{k \in S_{2j}} (1 - p_k^l) \right\}, \forall j \end{array} \right]. \quad (5.11b)$$

Initially, the bounds on θ are taken as given by (5.2b). Accordingly, let Problem ETO(Ω') be identical to Problem ETO(Ω), except that we now write $(p, l, \theta) \in \Omega'$ in (5.6l), and include just the bounds on the variables (y_1, y_2, z, ξ) in (5.6n), as respectively given by (5.3b), (5.4b), and (5.5b). The corresponding relaxation LP(Ω') is obtained from ETO(Ω') identically by replacing (5.6h) - (5.6k) with the associated approximating constraints of the type (5.8a, b).

Again, in a branch-and-bound scheme, when analyzing any node subproblem ETO(Ω') for a specified Ω' (conforming with (5.11a, b)), we solve the corresponding relaxation LP(Ω') to obtain an optimal solution \bar{x} . If the condition of Lemma 5.1 holds true, then we will have also solved the node subproblem ETO(Ω'). Otherwise, we apply Lemma 5.2 to possibly update the incumbent solution, and as necessary, we branch on this node by partitioning Ω' as follows.

Branching Rule B. Find the most discrepant identity among (5.6h) - (5.6k) at the current relaxation solution \bar{x} according to

$$\max \left\{ \left| \bar{y}_{1i} - \ln(\bar{p}_i) \right|, \forall i, \left| \bar{y}_{2i} - \ln(1 - \bar{p}_i) \right|, \forall i, \left| \bar{z}_j - \ln(\bar{\theta}_j) \right|, \forall j, \left| \bar{\xi}_j - \ln(\bar{l}_j) \right|, \forall j \right\}, \quad (5.12)$$

where ties can be broken by favoring the term involving the log of the variable having the largest bounding interval length. If this identified maximum term is one of the first two types of terms, then we partition the interval of the corresponding variable p_i into two subintervals: $[p_i^l, \bar{p}_i]$ and $[\bar{p}_i, p_i^u]$. Likewise, if the identified maximum is given by the

third or the fourth type of term, we partition the interval of the corresponding variable θ_j or l_j by cutting this interval at the respective value $\bar{\theta}_j$ or \bar{l}_j .

We are now ready to present a formal statement of the foregoing proposed algorithms and establish their convergence.

5.2. Global Optimization Branch-and-Bound Algorithms for Solving Problem ETO

We first describe the branch-and-bound procedure, namely Algorithm A, for solving Problem ETO. At each stage s of this procedure, $s = 0, 1, 2, \dots$, we will have a set of non-fathomed or *active nodes* A_s , where each node $a \in A_s$ is indexed by some hyperrectangle Ω^a . (To initialize, at $s = 0$, the set $A_0 = \{0\}$, with Ω^0 being given by (1g).) For each node $a \in A_s$, a lower bound LB_a will be given by $v[LP(\Omega^a)]$, where $v[P]$ denotes the optimal value for any Problem P. As a result, the *global lower bound* at stage s for problem ETO (equivalently, Problem ETO(Ω^0)) is given by

$$LB(s) \equiv \text{minimum } \{LB_a : a \in A_s\}. \quad (5.13)$$

Whenever any lower bounding node subproblem is solved, we can apply Lemma 5.2, in order to derive an upper bound on the overall problem ETO, where the corresponding p -value is given by (5.9). Accordingly, let p^* be the best such incumbent solution found, having an objective value of ν^* . Naturally, whenever $LB_a \geq \nu^*$, we fathom node a . (Practically, we can fathom node a whenever $LB_a \geq \nu^*(1 - \varepsilon)$, for some percentage optimality tolerance $100\varepsilon\% \geq 0$.) Hence, the active nodes at any stage s would satisfy $LB_a < \nu^*$, $\forall a \in A_s$. From this set of active nodes, we now select a node $a(s)$ that yields the least lower bound, *i.e.*, for which $LB_{a(s)} = LB(s)$ as given by (5.13). Note that for the corresponding solution

$$x^{a(s)} \equiv \left(p^{a(s)}, l^{a(s)}, r^{a(s)}, q^{a(s)}, \theta^{a(s)}, y_1^{a(s)}, y_2^{a(s)}, z^{a(s)}, \xi^{a(s)} \right) \quad (5.14)$$

to $\text{LP}(\Omega^a)$, we could not possibly have $p_i^l = p_i^u, \forall i$, and $l_j^l = l_j^u, \forall j$, because then, from Corollary 5.1, $x^{a(s)}$ would be feasible to $\text{LP}(\Omega^a)$, thereby yielding $\text{LB}_{a(s)} \geq v^*$, which is a contradiction. Hence, we find a branching variable and partition the node subproblem based on Branching Rule A, and proceed to the next stage. A formal statement of this proposed algorithm is given below.

Branch-and-Bound Algorithm A for Problem ETO

Step 0: Initialization. Set $s = 0, A_s = \{0\}, a(s) = 0, a = 0$, and let Ω^0 be given by (5.1g). Solve the linear program $\text{LP}(\Omega^0)$ and let x^0 be the solution obtained (as defined in Lemma 5.1) having an objective value LB_0 . Set the incumbent solution $p^* = \hat{p}$ as given by (5.9) of Lemma 5.2, where $(\bar{l}, \bar{r}, \bar{q}) = (l^0, r^0, q^0)$ and let the incumbent objective value be v^* as given via (5.10). If $\text{LB}_0 \geq v^*(1 - \varepsilon)$, for some optimality tolerance $\varepsilon \geq 0$, then stop with the incumbent solution as (ε) -optimal to Problem ETO. Otherwise, find a branching variable via Branching Rule A, and proceed to Step 1.

Step 1: Partitioning Step. Partition the current selected node $a(s)$ into two subnodes indexed by $a+1$ and $a+2$ according to Branching Rule A, and replace $A_s \leftarrow A_s \cup \{a+1, a+2\} - \{a(s)\}$. Let $\Omega^{a+h}, h=1, 2$, be the hyperrectangles corresponding to the nodes $a+1$ and $a+2$, respectively.

Step 2: Bounding Step. Solve $\text{LP}(\Omega^{a+h})$ for each $h = 1, 2$. Apply Lemma 5.2 to update the incumbent solution p^* and its value v^* , if possible, and determine a branching variable index according to Branching Rule A for each of these nodes (provided that its lower bound is lesser than $v^*(1 - \varepsilon)$) for possible future use. Replace $a \leftarrow a + 2$.

Step 3: Fathoming Step. Fathom any potentially non-improving nodes by setting $A_{s+1} = A_s - \{\hat{a} \in A_s : \text{LB}_{\hat{a}} \geq v^*(1 - \varepsilon)\}$. Increment s by 1.

Step 4: Termination Check and Node Selection. If $A_s = \emptyset$, then stop with the incumbent solution as (ε) -optimal. Otherwise, select an active node $a(s) \in \arg \min\{\text{LB}_{\hat{a}} : \hat{a} \in A_s\}$, and return to Step 1.

Theorem 5.1. (Main Convergence Result). *Algorithm A (run with $\varepsilon \equiv 0$) either terminates finitely with the incumbent solution being optimal to Problem ETO, or else an infinite sequence of stages is generated such that along any infinite branch of the branch-and-bound tree, any accumulation point of the (p, l, r, q) -variable part of the sequence of linear programming relaxation solutions generated for the corresponding node subproblems solves Problem ETO.*

Proof. The case of finite termination is clear. Hence, suppose that an infinite sequence of stages is generated. Consider any infinite branch of the branch-and-bound tree generated via the sequence of nested hyperrectangles $\Omega^{a(s)}$ that correspond to a set of stages s in some index set S . Hence, we have

$$\text{LB}(s) = \text{LB}_{a(s)} = v[\text{LP}(\Omega^{a(s)})], \quad \forall s \in S. \quad (5.15)$$

For each node $a(s)$, $s \in S$, let $x^{a(s)}$ be the solution obtained for $\text{LP}(\Omega^{a(s)})$. By taking any convergent subsequence, if necessary, using the boundedness of the sequence generated (and noting that the feasible region is a compact set), assume without loss of generality that

$$\{x^{a(s)}, \Omega^{a(s)}\}_S \rightarrow (x^*, \Omega^*). \quad (5.16)$$

We must show that the (p^*, l^*, q^*, r^*) -variable part of the solution x^* solves Problem ETO. First, note that since $\text{LB}_{a(s)}$ is the least lower bound at stage s , we have

$$\text{LB}_{a(s)} \leq v[\text{ETO}], \quad \forall s \in S. \quad (5.17)$$

Second, since Branching Rule A bisects the largest interval, we have that in the limit as $s \rightarrow \infty$, $s \in S$, $p_i^{*l} = p_i^{*u}$, $\forall i$, and $l_j^{*l} = l_j^{*u}$, $\forall j$. Hence, by Corollary 5.1, x^* solves $\text{ETO}(\Omega^*)$ with objective value $V^* \equiv \lim_{s \rightarrow \infty, s \in S} \text{LB}_{a(s)}$. Since the corresponding part

(p^*, l^*, q^*, r^*) is feasible to Problem ETO because $\Omega^* \subseteq \Omega$, we get, using (5.17), that

$$V^* = \lim_{s \rightarrow \infty, s \in S} \text{LB}_{a(s)} \leq v[\text{ETO}] \leq V^*, \quad (5.18)$$

or that equality holds throughout (5.18). This completes the proof. \square

Algorithm B

This alternative branch-and-bound procedure is identical to Algorithm A, except that at each stage s , every active node $a(s) \in A_s$ is associated with a hyperrectangle $\Omega^{(a(s))}$, where Ω' is defined by (5.11a, b), and at Steps 1 and 2, we adopt Branching Rule B. The following result establishes the convergence of Algorithm B.

Theorem 5.2. *Similar to the main convergence result, Algorithm B (run with $\varepsilon \equiv 0$) solves Problem ETO.*

Proof. The case of finite termination is clear. Hence, suppose that an infinite sequence of stages is generated. Consider any infinite branch of the branch-and-bound tree generated via the sequence of nested hyperrectangles $\{\Omega^{(a(s))}\}_S$ that correspond to a set of stages s in some index set S . Similar to the proof of Theorem 5.1, suppose that $\{x^{a(s)}, \Omega^{a(s)}\}_S \rightarrow (x^*, \Omega^*)$, and note that

$$V^* \equiv \lim_{s \rightarrow \infty, s \in S} \text{LB}_{a(s)} \leq \nu[\text{ETO}]. \quad (5.19)$$

Now, along this infinite branch of the enumeration tree, some p_i , θ_j , or l_j variable is partitioned infinitely often over nodes $s \in S_1 \subseteq S$, say. Moreover, in the limit, by virtue of the partitioning scheme and following the proof of Theorem 1 in Sherali and Tuncbilek (1992), we have that this variable equals one of its bounds. Hence, from Corollary 5.2, its discrepancy in (5.12) approaches zero. However, since the discrepancy of this variable was the maximum in (5.12), for all $s \in S_1$, we have by taking limits as $s \rightarrow \infty$, $s \in S_1$, that x^* satisfies (5.6h) - (5.6k). By Lemma 5.1, x^* solves $\text{ETO}(\Omega^*)$ with objective value V^* . Using (5.19), we therefore obtain

$$V^* \leq \nu[\text{ETO}] \leq V^*,$$

and this completes the proof. \square

Remark 5.2. In our computations, we will investigate the relative merits of each of the foregoing proposed algorithms, A and B. Other similar partitioning rules that support either Corollary 5.1 or Corollary 5.2, or more generally, which imply in the limit that the

discrepancies in the polyhedral outer-approximations tend towards zero, would also yield theoretically convergent algorithmic procedures.

Remark 5.3. Note that in the approach of Algorithm B, we could perform a partitioning on the original set Ω , in lieu of partitioning Ω' , by eliminating θ_j from $\text{ETO}(\Omega)$ and replacing its objective function equivalently by the convex function:

$$\text{Minimize} \quad \sum_{j \in \tau} e^{\bar{z}_j}. \quad (5.20)$$

In this case, the lower bounding problem would be a nonlinear program (say, $\text{NLP}(\Omega)$) in lieu of the linear program $\text{LP}(\Omega)$, based on the objective function (5.20). By partitioning Ω based on Branching Rule B (ignoring the θ -constraints and variables), we would again obtain convergence to a global optimum to Problem ETO according to Theorem 5.2. However, in this method, we would now need to contend with solving nonlinear (convex) lower bounding problems, which might increase the computational effort and inhibit robustness. Some related computational results are presented in Section 5.3.

Remark 5.4. In the proposed ETO model, note that the loss magnitude l_j , associated with an end-node j , is assumed to reduce *linearly* with respect to the allocated quantities of mitigation resources $r_{jn}, \forall n=1, \dots, N$. Alternatively, suppose that we consider an *exponential* reduction in the loss magnitude l_j , i.e.,

$$l_j = b_{j0} e^{-h_j \sum_{n=1}^N b_{jn} r_{jn}}, \quad \forall j \in \tau, \quad (5.21)$$

where, b_{j0}, \dots, b_{jN} , and h_j are given nonnegative constants.

In this case of nonlinear diminishing marginal reductions in the loss magnitudes, the problem can be further simplified by eliminating the l_j terms from the ETO model, as follows. As before, taking logarithms in (5.21), we get,

$$\xi_j = \ln(l_j) = \ln(b_{j0}) - h_j \sum_{n=1}^N b_{jn} r_{jn}, \quad \forall j \in \tau. \quad (5.22)$$

Using (5.22), we can now propose an alternative event tree optimization model, which we

refer to as **ETO2**(Ω), where the objective function and constraints are identical to **ETO**(Ω), except that constraint (5.6f) is now replaced by the corresponding equation,

$$\xi_j + h_j \sum_{n=1}^N b_{jn} r_{jn} = \ln(b_{j0}), \forall j \in \tau. \quad (5.23)$$

Thus, Problem **ETO2**(Ω) can be formally represented as: (5.6a) - (5.6e), (5.6g) - (5.6n), along with (5.23). Due to the similarity of this model with **ETO**(Ω), we provide computational results in Section 5.3 below only for Problem **ETO**(Ω), in concert with the proposed branch-and-bound algorithms A and B, and we leave a more detailed study of **ETO2**(Ω) for future research.

5.3. Computational Case Study

To illustrate the proposed approach, consider the following hypothetical case study. Assume that a gas-line rupture has led to a gas leak, where the cascading risk scenarios that occur due to this hazardous event are as illustrated in Figure 5.1. Furthermore, suppose that some five preventive and five mitigation resources are available for deployment to counter this particular hazard. Tables 5.1 and 5.2 display the logit model coefficients corresponding to the preventive and mitigation resources, namely, the a_{im} - and b_{jn} -values, respectively. Note that these coefficients reflect the appropriate units of the entities that they represent. For example, if the quantities of

Node (i) \ Resource (m)	0	1	2	3	4	5
1	0	0.74	0.74	0.94	0.73	0.51
2	0	0.62	0.76	0.55	0.71	0.62
3	0	0.59	0.62	0.57	0.89	0.92
4	0	0.65	0.78	0.79	0.50	0.91
5	0	0.73	0.56	0.93	0.51	0.82
6	0	0.93	0.99	0.60	0.91	0.52
7	0	0.94	0.60	0.67	0.50	0.53
8	0	0.53	0.66	0.97	0.61	0.55

Table 5.1: Logit-model coefficients (a_{im}) corresponding to preventive resources.

Resource (n) \ Node $j \in \tau$	0	1	2	3	4	5
9	136720	4635.16	1005.49	4050.58	2434.85	1545.94
10	138460	2735.52	2442.04	4773.24	1311.69	2583.55
11	181940	3270.69	3664.84	4271.86	4805.50	2181.20
12	132271	1071.06	1491.60	4975.41	2059.75	3629.28
13	170787	3969.56	1877.52	4797.54	4706.84	1727.29
14	105984	1385.09	2382.63	3955.48	1135.69	1739.53
15	143064	2797.74	1972.11	1495.41	3115.32	4925.86
16	190149	3398.19	4881.00	4766.68	1086.71	4882.35
17	123229	4440.44	2958.58	2226.26	1545.62	1152.08

Table 5.2: Logit-model coefficients (b_{jn}) corresponding to mitigation resources.

preventive resources, q_{im} , are measured in 10^3 dollars, then the corresponding logit model coefficients are measured per 10^3 dollars. Similarly, if the total loss for each consequence is measured in dollars, say, and the mitigation resources are measured in 10^3 dollars, then the corresponding coefficients in Table 5.2 are expressed in dollars saved per 10^3 dollars invested.

The per-unit-costs of allocating the preventive and mitigation resources, namely, the c_{im} - and d_{jn} -values, are displayed in Tables 5.3 and 5.4, respectively. Moreover, the total available budget, and the upper bounds on the preventive and mitigation resources are assumed to be $\beta = 7500$, $s_m = 10, \forall m$, and $t_n = 50, \forall n$, respectively. Furthermore, we assume that the failure probability at any node is at most 1%, and correspondingly, any preventive action taken at the node cannot reduce the failure probability to less than 0.01%, *i.e.*, $p_i^l = 0.0001$ and $p_i^u = 0.01, \forall i$. Finally, the lower bounds corresponding to the loss, l_j , for each end node j , are displayed in Table 5.5, and the loss upper bounds are assumed to be $l_j^u = 5 \times 10^5, \forall j$. Clearly, end nodes 9 and 17 have the minimum and maximum loss values, respectively.

Tables 5.1 through 5.5, therefore provide the input data for the associated problem $ETO(\Omega)$, given by (5.6a - 5.6n), which was solved via the branch-and-bound algorithms A and B described in Section 5.2. These algorithms were implemented using a combination of CPLEX 9.0.0 and a code developed in C++. The (global) optimal

Node (i) \ Resource (m)	1	2	3	4	5
1	30	33	22	31	24
2	26	30	29	23	26
3	24	40	33	36	25
4	22	24	34	34	26
5	23	23	39	23	21
6	35	21	26	22	33
7	32	40	37	39	24
8	37	31	34	26	25

Table 5.3: Per-unit-costs (c_{im}) corresponding to allocating preventive resources.

Node $j \in \tau$ \ Resource (n)	1	2	3	4	5
9	21	36	25	21	30
10	21	23	37	23	29
11	31	36	29	23	38
12	33	23	31	35	40
13	29	32	31	36	36
14	22	27	37	39	31
15	29	38	24	39	23
16	35	28	33	32	21
17	31	40	22	31	34

Table 5.4: Per-unit costs (d_{jn}) corresponding to allocating mitigation resources.

Node $j \in \tau$	9	10	11	12	13	14	15	16	17
Loss (l_j)	0	5000	5000	10000	10000	20000	20000	30000	50000

Table 5.5: Lower bounds corresponding to the loss accrued at each end-node j .

preventive and mitigation resource assignments (q_{im} and r_{jn} -values) along with the associated probabilities of failure (p_i) and losses (l_j), obtained by solving Problem ETO(Ω) via Algorithm A are displayed in Tables 5.6 and 5.7. (Note that Algorithm B obtained identical resource allocations to those displayed in Tables 5.6 and 5.7.) From the optimal allocations, observe that using the available quantities of resources, the failure probabilities for four of the eight nodes are at their upper bounds, but the others are

Resource (m) Node (i)	1	2	3	4	5	Probability (p_i)
1	5.112	-	3.766	0.625	-	4×10^{-3}
2	-	5.358	-	3.001	-	0.002
3	-	-	-	6.373	-	0.003
4	-	-	-	-	5.050	0.010
5	-	-	1.497	-	4.950	0.004
6	-	4.642	-	-	-	0.010
7	4.888	-	-	-	-	0.010
8	-	-	4.737	-	-	0.010
Total preventive resource available	10	10	10	10	10	

Table 5.6: Global optimal preventive resource assignments for each node (q_{im} -values) obtained by solving Problem ETO using the proposed algorithms.

Resource (n) Node $j \in \tau$	1	2	3	4	5	Loss (l_j)
9	29.496	-	-	-	-	0
10	-	34.260	8.161	-	4.197	5000.00
11	-	-	-	36.82	-	5000.00
12	-	-	24.575	-	-	10000.00
13	4.012	-	17.264	13.180	-	10000.00
14	-	15.740			-	68480.77
15	-	-	-	-	24.983	20000.00
16	-	-	-	-	-	190149.0
17	16.491	-	-	-	-	50000.00
Total mitigation resource available	50	50	50	50	50	

Table 5.7: Global optimal mitigation resource assignments for each node (r_{jn} -values) obtained by solving Problem ETO using the proposed algorithms.

considerably reduced in a strategic manner based on the loss consequences that are prompted by these features. A similar observation holds true for the case of mitigating the end node loss values in coordination with the foregoing failure probability manipulations. The objective function value pertaining to the global optimum is given by 80.9983. Using $\varepsilon = 10^{-6}$, Algorithm A enumerated a total of 60 nodes over a CPU time of 0.12 seconds

for solving Problem ETO(Ω) to global optimality. Similarly, Algorithm B achieved the global optimum while enumerating 51 nodes and a total CPU time of 0.13 seconds.

For the purpose of comparison, Problem ETO, given by (5.1a) - (5.1h), was also directly solved by using the commercial global optimizer BARON (see Sahinidis, 1996). In addition, we solved Problem ETO using the proposed nonlinear relaxation approach (refer to Remark 5.3), with the help of BARON. Both of these computational experiments yielded identical results, and the optimal resource assignments for these two cases are displayed in Tables 5.8 and 5.9. The best objective function value obtained was 92.9335, which is greater (worse) than the global optimum by 14.73%. From Tables 5.8 and 5.9, observe that this significantly inferior solution computed for Problem ETO using BARON yields considerably different resource allocations, especially for the mitigation alternatives, than those displayed in Tables 5.6 and 5.7.

Next, we examined the number of nodes enumerated along with the CPU times when solving the case study example using each of the branch-and-bound strategies, for optimality tolerance (ε) values of 0.05, 0.01, and 10^{-6} . The results obtained are displayed in Table 5.10. Since the convergence of the proposed algorithms essentially involves the process of driving the variables to achieve their bounds, the most successful branching strategy would be one that guides this propensity in the most efficient way by quickly creating partitions at optimal values via the linear programming relaxations. Evidently,

Resource (m) Node (i)	1	2	3	4	5	Probability (p_i)
1	5.112	-	3.914	0.747	-	3×10^{-3}
2	-	5.358	-	2.796	-	0.002
3	-	-	-	6.457	-	0.003
4	-	-	-	-	5.050	0.010
5	-	-	1.349	-	4.950	0.005
6	-	4.642	-	-	-	0.010
7	4.888	-	-	-	-	0.010
8	-	-	4.737	-	-	0.010
Total preventive resource available	10	10	10	10	10	

Table 5.8: Global optimal preventive resource assignments for each node (q_{im} -values) obtained by solving Problem ETO using BARON.

Resource (n) Node $j \in \tau$	1	2	3	4	5	Loss (l_j)
9	29.496	-	-	-	-	0
10	-	-	27.96	-	-	5000.00
11	-	-	-	36.82	-	5000.00
12	-	-	0.422	0.150	0.299	128779.67
13	-	-	21.618	12.126	-	10000.00
14	6.09	23.273			-	42098.517
15	5.205	-	-	-	17.767	40983.218
16	-	2.292	-	-	28.520	39714.551
17	9.208	2.606	-	0.904	1.074	71994.605
Total mitigation resource available	50	50	50	50	50	

Table 5.9: Global optimal mitigation resource assignments for each node (r_{jn} -values) obtained by solving Problem ETO using BARON.

Algorithm B is more strongly geared towards this construct, because it directly involves the additional partitioning on the θ -variables. Hence, Algorithm A uniformly enumerated a greater number of nodes to achieve global optimality than did Algorithm B. However, since the effort per node is somewhat lesser for Algorithm A, the overall CPU time for the two methods turned out to be comparable.

Algorithm	Optimality Tolerance						Averages (#Nodes, CPU Time)
	$\varepsilon = 0.05$		$\varepsilon = 0.01$		$\varepsilon = 10^{-6}$		
	#Nodes	CPU Time (s)	#Nodes	CPU Time (s)	#Nodes	CPU Time (s)	
A	15	0.04	36	0.05	60	0.12	(37, 0.07)
B	9	0.04	25	0.06	51	0.13	(28.33, 0.076)

Table 5.10: Computational results obtained for comparing the two proposed algorithms.

Finally, we examined the sensitivity of the model with respect to the parameters t and β . (A preliminary computational experiment revealed that these two parameters had the most significant impact on the optimal objective function value.) Figure 5.3 displays the variation in the objective function when the parameter $t \in \{10, 30, 50, 70\}$ and

$\beta \in \{2500, 5000, 7500\}$. As expected, the objective function value follows the law of diminishing marginal returns, decreasing rapidly initially as the quantity of mitigation resource increases, but flattening out for larger values of the parameters. Moreover, for values of β greater than 7500, the variation in the objective function was identical to the case of $\beta = 7500$, and no further advantage is gained by increasing β any further.

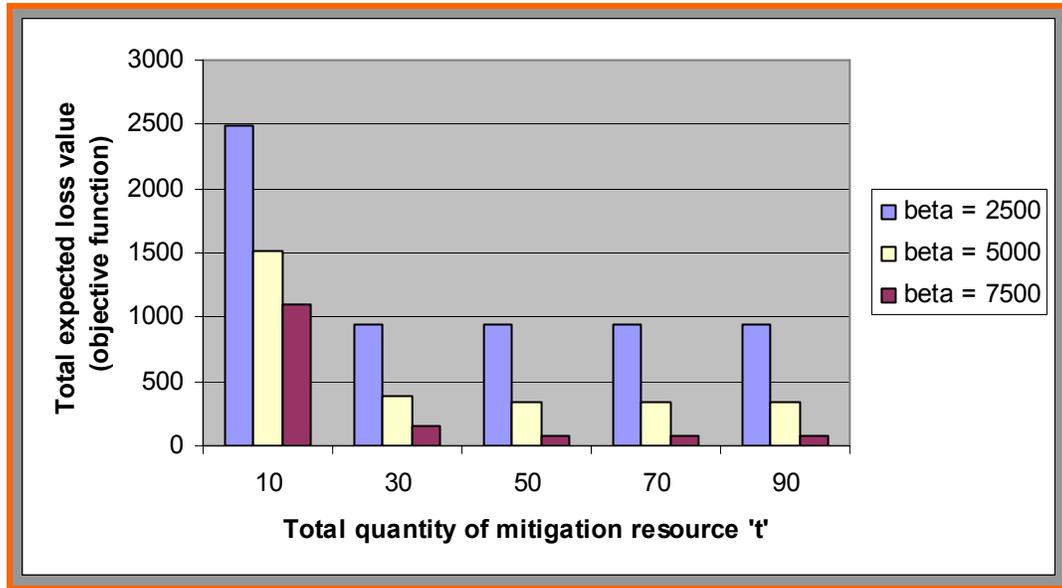


Figure 5.3: Sensitivity of the objective function with respect to the parameters t and β .

5.4. Summary and Conclusions

This research effort introduces a novel strategic planning problem of employing certain available preventive and mitigation resources to respectively curtail the failure probabilities of system safety features and the consequences (cost) of end-effects, in order to minimize the total risk (expected loss) in the aftermath of a hazardous event. Following an event tree optimization approach, this cascading risk scenario problem was modeled as a nonconvex factorable program, which was solved through a suitable variable transformation and polyhedral outer-approximation technique, applied in concert with two specialized branch-and-bound procedures. These developed algorithms were proven to converge to a global optimal solution. Computational results obtained by solving a hypothetical case study were presented, and variations in the solution and

algorithmic performance with respect to the available resource parameters and the alternative algorithmic strategies were investigated. In particular, the proposed algorithm was demonstrated to more robustly yield global optimal solutions in comparison with the commercial global optimizer BARON.

The contribution in this chapter has focused on presenting a first analytical approach toward optimizing an event tree through a novel modeling and algorithmic approach. There are several variations and extensions of this work that could be considered for future research. For example, instead of minimizing the total expected loss, we could alternatively minimize the *maximum expected loss* that occurs in the system under consideration. In such a case, the objective function could be modeled as:

Minimize η , where $\eta \geq l_j \prod_{k \in S_{1j}} p_k \prod_{k \in S_{2j}} (1 - p_k)$, $\forall j \in \tau$. Variable transformation

strategies identical to those adopted in reformulating Problem ETO could be employed to solve this problem. Also, the alternative model described in Remark 5.4 concerning exponential loss mitigation functions for end-nodes could be investigated for both the expected and minimax objective functions. The performance of the proposed branch-and-bound algorithms for other scenarios where the event tree is generic in nature, as opposed to binary, can also be tested, along with a more extensive computational investigation involving several other case studies.

6. Control of Linear Systems

The most commonly used systems found in any industrial setting are dynamic systems. A primary component that deals with the operation of a dynamic system is referred to as the controller. The field of *control systems* deals with developing a control strategy based on which an appropriate dynamic model can be generated for the system under consideration. For the purpose of illustration, consider a dynamic system depicted in Figure 6.1. Here, the controller receives an input, and generates the necessary command, which leads to the process being executed. After the process is completed, a feedback mechanism relays the result of either the successful or the unsuccessful process back to the controller.

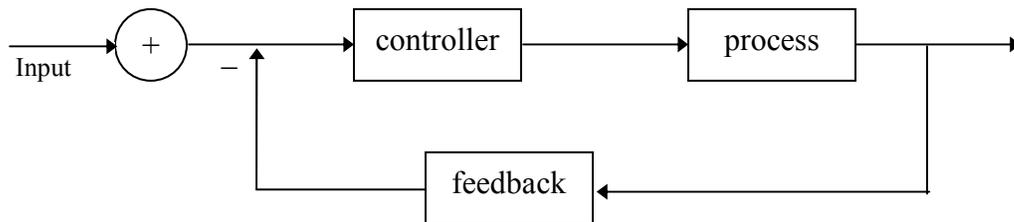


Figure 6.1: Basic structure of a control system.

The concept of *stability* arises in such a context when it is expected that for a bounded input, the output of the system must also be bounded. Systems that exhibit (do not exhibit) such expected behavior are categorized as *stable (unstable)* systems. Even more critical than stability is the issue of *relative stability*, wherein we are interested in determining by how much a stable system can be perturbed without losing stability. For specially structured problems, whose dynamic behavior can be described by a set of linear differential equations with constant (certain) parameters, *i.e.*, *linear time invariant* systems, we can provide exact answers to stability related issues.

The control of linear systems with *uncertain* physical parameters has been a main subject of research in the field of control engineering in the last two decades (see Barmish, 1994, Bhattacharya *et al.*, 1995, and Ackermann, 2002). One approach to deal with this problem is to study the behavior of the polynomial characteristic equations of these systems, and so a good deal of research effort has been invested on exploring the

properties of these polynomials. In particular, the effects of uncertain parameters on the location of the roots of the polynomials, and thence on the stability and the performance of these systems, has been widely investigated.

For the cases when the coefficients of the polynomials are linear functions of the system's uncertain parameters, many powerful tools have been developed (see Barmish, 1994, and Bhattacharya *et al.*, 1995). However, in many engineering applications, these coefficients often turn out to be multilinear, polynomial, or a general nonlinear function of the system's uncertain physical parameters (Ackermann, 2002), and in these cases, the research has been less fruitful. Several representative contributions in these problem areas, mainly in the stability analysis context, are given in Chapellat *et al.* (1993), Polyak and Kogan (1995), and Djaferis (1995).

A fundamental problem that often arises in robust control is the calculation of *stability margins* for parameter perturbations, *i.e.*, determining the maximum allowable perturbation in the uncertain physical parameters of a stable system without losing stability. Here, stability is defined with respect to an arbitrary region D in the complex plane, and is referred to as " D -stability". The ability to compute such stability margins is significant in the design of robust controllers (Bozorg and Nebot, 1999). For the case of linear dependency of coefficients with respect to these uncertain parameters, several algorithms have been developed for determining the D -stability margins (Hinrichsen and Pritchard, 1989, Qiu and Davison, 1989, and Teboulle and Kogan, 1994).

However, for cases where the coefficients are polynomial functions of uncertain parameters, only a handful of results are available in the literature to calculate the system's stability margins. In Ackermann *et al.* (1990), a graphical method is presented to visualize the stability domains of polynomials in the parameter space, but this method is suitable for instances when the number of uncertain parameters is relatively small. In Sideris and Sánchez Peña (1989), it has been demonstrated that a polynomial with a polynomial uncertainty structure and a polytopic domain of parameter uncertainty can be transformed into a polynomial having a multilinear uncertainty structure in conjunction with a new polytopic uncertainty domain. Thus, multilinear results are applicable to these cases. For the case of multilinear dependency in the parameters, the well-known Mapping Theorem (Zadeh and Desoer, 1963) is one of the few tools available for checking robust

stability; however, as recognized in Polyak and Kogan (1995), the sufficiency conditions of this theorem can lead to very conservative results. Despite this limitation, the Mapping Theorem has been extensively used in the development of algorithms for the calculation of stability margins for the multilinear case (De Gaston and Safonov, 1988, and Keel and Bhattacharya, 1993).

However, we note that for general l_p -norm perturbations with an arbitrary p , where $1 < p < \infty$, this problem has not been especially investigated, and to the best of our knowledge, there does not exist any tractable algorithm for computing the exact D -stability margins corresponding to the cases of either multilinear or polynomial uncertainty structures. Moreover, due to the nonconvexity of the stability domains in the parameter space, some researchers have tried to approximate the stability domains using a convex inner-approximation approach (e.g., Henrion *et al.*, 2003). Likewise, other convexification schemes using linear matrix inequalities (LMIs) have also been proposed in the literature (see El Ghaoui and Niculescu, 2000, Parrilo, 2003, and Lasserre, 2001). In this research effort, we present several algorithms, predicated on the *Reformulation-Linearization Technique* (RLT) (see Sherali and Tuncbilek, 1992, 1997, and Sherali and Wang, 2001), for the computation of l_p -norm stability margins for the case of polynomial uncertainty structures corresponding to different values of p .

The remainder of this chapter is organized as follows. First, in Sections 6.1 and 6.2, we state the familiar zero-exclusion results (see Barmish, 1994) that are used to formulate the problem of computing the stability margins as a polynomial optimization problem, and the existence of a solution to this optimization problem is established. Then in Section 6.3, several tailor-made algorithms, based on the RLT, are developed to compute lower bounds for the D -stability margins for cases corresponding to different l_p -norms. Moreover, we show how the structure of the system under study and the objective function (for various l_p -norms) can be used to tighten the lower bounds obtained via the RLT relaxations, thereby facilitating an effective application of this methodology. Several illustrative examples that demonstrate the algorithmic steps are provided in Section 6.4, accompanied by related computational experience. Finally, Section 6.5 concludes the chapter with a summary.

6.1. Formulation of the D -stability Margin Problem

Consider the polynomial

$$Q(q, s) = a_n(q)s^n + \cdots + a_1(q)s + a_0(q); \quad a_n(q) \neq 0, \quad (6.1)$$

where the coefficients $a_n(q)$, \dots , $a_1(q)$, and $a_0(q)$ are polynomial functions of the parameters $q = [q_1, q_2, \dots, q_m] \in \mathbb{R}^m$. Let D be an open subset of the complex plane, \mathbb{C} , and denote its contour by C_D . Below, the well-known theorem on the robust stability of linear systems is cited.

Theorem 6.1. (Zero Exclusion Theorem). *Consider the family of invariant-degree polynomials (6.1) with $q \in Q$, where Q is an uncertainty set that is path-wise connected, and that has at least one D -stable member. Then, the family of polynomials is D -stable if and only if*

$$0 \notin Q(q, u), \quad \forall q \in Q, \quad \forall u \in C_D. \quad (6.2)$$

Proof. See Barmish (1994). \square

To check the necessary and sufficient conditions of Theorem 6.1 for D -stability, we sweep the contour of the complex region C_D , using a *sweeping variable*, say z . Any point u on the contour C_D can be expressed as a function of this sweeping variable z , where $z \in Z \subset \mathbb{R}$, i.e., $u = u(z)$, $\forall u \in C_D \subset \mathbb{C}$. (For instance, the contour of the unit circle can be represented by $u(z) = e^{iz}$, $z \in [0, 2\pi]$, where $i = \sqrt{-1}$.) Substituting $s = u(z)$ in (6.1) yields

$$Q(q, u(z)) = Q_R(q, z) + i Q_I(q, z), \quad (6.3)$$

where,

$$Q_R(q, z) \equiv \operatorname{Re}[Q(q, u(z))] = \sum_{j=0}^n \alpha_j(z) a_j(q), \quad (6.4)$$

$$Q_I(q, z) \equiv \operatorname{Im}[Q(q, u(z))] = \sum_{j=0}^n \beta_j(z) a_j(q), \quad (6.5)$$

and where,

$$\alpha_j(z) \equiv \operatorname{Re}[u(z)]^j, \beta_j(z) \equiv \operatorname{Im}[u(z)]^j, j = 0, \dots, n.$$

Define the weighted l_p -distance (norm) of two arbitrary points $q, q' \in \mathbb{R}^m$ in the parameter space as

$$\delta_p(q, q') \equiv \left[\sum_{k=1}^m (|q_k - q'_k| / w_k)^p \right]^{1/p}, \quad (6.6)$$

where $w_k > 0, \forall k = 1, \dots, m$, are assigned weights and $p \in (1, \infty)$ is a constant.

Now, consider the family of polynomials

$$P \equiv \{Q(q, s) : q \in B_p(\bar{q}, b_p)\}, \quad (6.7)$$

where the *parameter uncertainty domain* given by

$$B_p(\bar{q}, b_p) \equiv \{\tilde{q} : \delta_p(\bar{q}, \tilde{q}) < b_p\}, \quad (6.8)$$

is an l_p -hypersolid in the parameter space, b_p is the size of the hypersolid, and \bar{q} is the nominal value of the parameter vector q , and where the family is assumed to be D -stable. Note that the general hypersolid turns out to be a hyperellipsoid for $p = 2$, a hyperrectangle for $p = \infty$, and a hyperdiamond for $p = 1$.

6.2. Computing D -stability margins

Given the above, the problem of computing the D -stability margins is equivalent to determining the *maximum size* of the hypersolid, namely b_p (defined in (6.8)), so as to preserve D -stability.

To calculate the D -stability margins, define the following optimization problem (see Tesi and Vicino, 1990, and Desages *et al.*, 1991): Given $\delta_p(q, q')$ as defined in (6.6), and Q_R and Q_I as defined in (6.4) and (6.5), respectively, and the nominal value of $q = \bar{q}$, let the sweeping variable $z \in Z$ be fixed, and define

$$\rho_p(z) = \min_q \{\delta_p(\bar{q}, q)\}, \quad (6.9a)$$

$$\text{subject to } Q_R(q, z) = 0, \quad (6.9b)$$

$$Q_l(q, z) = 0, \quad (6.9c)$$

where the function $\rho_p(z)$ in (6.9a) is called the *Minimum l_p -Distance Function (MDF)*.

Moreover, a necessary condition of Theorem 6.1 for D -stability is the invariance of the degree of the perturbed polynomials (see Barmish, 1994). This implies that the parameter perturbations must not result in the nullification of $a_n(q)$. To satisfy this condition, the perturbation $\delta_p(\bar{q}, \tilde{q})$, where \tilde{q} represents a perturbed value of q , must be less than the following optimal value.

$$\eta_p = \min_q \{ \delta_p(\bar{q}, q) \}, \quad (6.10a)$$

$$\text{subject to} \quad a_n(q) = 0. \quad (6.10b)$$

Note that the optimization (6.10) is independent of the sweeping variable z , and thus, it is not required to search the contour to evaluate η_p .

Now, assuming that b_p is as specified in (6.8), we state the following theorem, which proves the necessary and sufficient conditions for D -stability.

Theorem 6.2. *The family of polynomials P , given by (6.7), is D -stable if and only if*

$$b_p < \gamma_p, \quad (6.11)$$

where

$$\gamma_p \equiv \min \left\{ \eta_p, \min_z \rho_p(z) : z \in Z \right\}. \quad (6.12)$$

Proof. By assumption, note that a member of the family (6.7) is stable at $q = \bar{q}$. If the size of the perturbation of the hypersolid, namely b_p , is smaller than η_p , then the condition $a_n(q) \neq 0$ is satisfied for any $q \in B_p(\bar{q}, b_p)$. Hence, the pre-conditions for Theorem 6.1 are satisfied. Moreover, if b_p is smaller than $\min_{z \in Z} \{ \rho_p(z) \}$, this implies that for any $q \in B_p(\bar{q}, b_p)$ and $z \in Z$, constraints (6.9) will not be satisfied, and so, the necessary and sufficient conditions of Theorem 6.1 for D -stability are met. This completes the proof. \square

Remark 6.1. For any z , if (6.9) is infeasible, then evidently $\rho_p(z) = \infty$. Else, given a feasible solution \hat{q} to (6.9), the problem is equivalent to one that further restricts $\delta_p(\bar{q}, q) \leq \delta_p(\bar{q}, \hat{q})$. Then, because the resulting problem (6.9) reduces to minimizing a continuous function over a nonempty compact set, by Weierstrass' Theorem (see Royden, 2001), a minimum exists that defines $\rho_p(z)$.

6.3. Global Optimization of the D -stability Margin Problem

We are now ready to embed the optimization problem given by (6.9) for computing the required MDFs into an RLT-based branch-and-bound scheme (see Sherali and Tuncbilek, 1992). A similar procedure can be followed for solving (6.10). For given values of the sweeping parameter z and the nominal parameter vector \bar{q} , the following *Minimum Distance Estimation (MDE)* problem is defined:

$$\text{MDE:} \quad \text{Minimize } f(q) \equiv [\delta_p(q, \bar{q})]^p = \sum_{k=1}^m c_k |q_k - \bar{q}_k|^p \quad (6.13a)$$

$$\text{subject to} \quad Q_R(q, z) = \sum_{j=0}^n \alpha_j(z) a_j(q) = 0, \quad (6.13b)$$

$$Q_I(q, z) = \sum_{j=0}^n \beta_j(z) a_j(q) = 0, \quad (6.13c)$$

where $c_k = 1/(w_k)^p$ is a weighting parameter associated with a performance index. Let us denote the highest order of any polynomial term appearing in Problem MDE to be given by Δ , $\Delta \geq 2$.

Since the objective function is dependent on the parameter p , the following RLT-based algorithm is designed for different values of p . We begin by describing certain steps that are common for all l_p -norm objective functions.

Step 1: Heuristic Solution Step. Assume that by starting at some suitable solution, e.g., $q = \bar{q}$, we apply a local search procedure to find a feasible solution \hat{q} (see Bazaraa *et al.*, 1993, for example), having a corresponding objective function value \hat{v} .

Step 2: Bounding Step. Based on the solution value \hat{v} obtained at Step 1, we can impose the objective function cut,

$$f(q) \leq \hat{v}. \quad (6.14)$$

By the nature of $f(q)$, which comprises nonnegative, separable terms, we can assert that

$$c_k |q_k - \bar{q}_k|^p \leq \hat{v}, \quad \forall k = 1, \dots, m. \quad (6.15)$$

This yields lower and upper bounds on the q -variables, given by

$$q_{kl} \leq q_k \leq q_{ku}, \quad \forall k = 1, \dots, m, \quad (6.16)$$

where,

$$q_{kl} = \bar{q}_k - (\hat{v}/c_k)^{1/p}, \quad q_{ku} = \bar{q}_k + (\hat{v}/c_k)^{1/p}, \quad \forall k = 1, \dots, m.$$

Remark 6.2. Note that the tightness of the lower and upper bounds derived at Step 2, given by (6.16), depends upon \hat{v} , which measures the quality of the feasible solution obtained at Step 1. These bounds can significantly enhance the convergence performance of the RLT-based algorithm described in the sequel. Alternatively, failing the availability of a feasible solution, we can either impose some known practical bounds on the q -variables, or we can estimate a suitable ad-hoc upper bound, \hat{v} , and thereby derive (6.16) predicated on (6.15). If the optimum objective function value to Problem (6.13) turns out to be less than or equal to the ad-hoc value \hat{v} , then we are done. Otherwise, an appropriate adjustment in \hat{v} could be made, and the process re-iterated.

Step 3: Reformulation Phase. Let \bar{M} denote the set comprised of Δ replicates of M , where $M \equiv \{1, \dots, m\}$. Then, the set of bound-factor product constraints as defined in Sherali and Tuncbilek (1992) is given by

$$\prod_{k \in K_1} (q_k - q_{kt}) \prod_{k \in K_2} (q_{ku} - q_k) \geq 0, \quad (6.17)$$

where, $K_1 \subseteq \bar{M}$, $K_2 \subseteq \bar{M}$, and $|K_1| + |K_2| = \Delta$.

Note that there are $\binom{2m + \Delta - 1}{\Delta}$ such bound-factor constraints. These constraints are appended to Problem MDE to yield the reformulated problem.

Step 4: Linearization Phase. Define the RLT product variables

$$Q_K \equiv \prod_{k \in K} q_k, \quad \forall K \subseteq \bar{M}, \quad 2 \leq |K| \leq \Delta, \quad (6.18)$$

where, the indices in K are assumed to be ordered in nondecreasing order. Note that there are $\binom{m+\Delta}{\Delta} - (m+1)$ such distinct Q -variables. For notational convenience, let $Q_k \equiv q_k$, $\forall k = 1, \dots, m$, and $Q_{\{\emptyset\}} \equiv 1$. Furthermore, for any polynomial function $h(q)$, let $[h(q)]_L$ denote the linearized function expressed in terms of the variables q and Q , which is obtained by substituting (6.18) for each distinct polynomial product term. Accordingly, let us define,

$$\Omega = \{q : q_{kl} \leq q_k \leq q_{ku}\}, \quad \forall k = 1, \dots, m. \quad (6.19)$$

Recognizing that the constraints (6.17) are contingent upon these bounds, let

$$X_L(\Omega) = \left\{ \begin{array}{l} (q, Q) : \sum_{j=0}^n \alpha_j(z) [a_j(q)]_L = 0, \quad \sum_{j=0}^n \beta_j(z) [a_j(q)]_L = 0, \\ \left[\prod_{k \in K_1} (q_k - q_{kl}) \prod_{k \in K_2} (q_{ku} - q_k) \right]_L \geq 0, \quad \forall K_1 \subseteq \bar{M}, K_2 \subseteq \bar{M}, |K_1| + |K_2| = \Delta, q \in \Omega \end{array} \right\}.$$

Note that this set represents an RLT-based linearization of the constraints of Problem MDE given in (6.13), augmented with the linearized bound-factor product constraints (6.17).

We are now ready to pursue different strategies to tackle Problem MDE depending on different values of p .

Case of $p = 2$ (Weighted Least Squares Minimization.)

The case of $p = 2$ corresponds to a weighted least squares minimization MDE problem.

Step 5: Optional Tightening of Bounds in Ω : Note that $f(q)$, as defined in (6.13a), is a quadratic function. Moreover, we can also impose (6.14) as an additional constraint in the problem. Accordingly, define

$$X_L(\Omega, \hat{\nu}) \equiv X_L(\Omega) \cap \{(q, Q) : [f(q)]_L \leq \hat{\nu}\}. \quad (6.20)$$

In lieu of deriving the bounds (6.16) based solely on (6.14), we can now sequentially minimize and maximize each q_k in turn, subject to the linear constraints (6.20), in order to respectively tighten the lower and upper bounds on $q_k, \forall k = 1, \dots, m$. Each time a bound actually improves, we update the hyperrectangle Ω and the corresponding set $X_L(\Omega)$ used in (6.20) for the next minimization in turn. This bound tightening process is useful for improving the fidelity of the polyhedral approximation $X_L(\Omega)$ with respect to the underlying polynomial constrained region (see Sherali and Tuncbilek, 1992). Having performed one complete pass through all the variables $q_k, \forall k = 1, \dots, m$, if the volume of the resulting hyperrectangle Ω is less than, say, 90% of the volume of the original Ω (this percentage value is arbitrarily chosen to ensure that a reasonable extent of the tightening of bounds is achieved), we repeat this process. (This procedure can be iteratively performed for a maximum of, say, three such loops in order to conserve computational effort, but can be terminated before this limit is reached whenever the required percentage reduction in volume does not occur.) Note that the linear programs that sequentially minimize or maximize any q_k subject to $(q, Q) \in X_L(\Omega)$ are typically easy to solve in terms of computational time, and moreover, frequently provide a beneficial tightening of the bounds.

Step 6: Branch-and-Bound Procedure. Given Ω , we prescribe an RLT-based branch-and-bound algorithm based on concepts derived from Sherali and Tuncbilek (1992, 1997) for solving Problem MDE. In this methodology, for any constraining hyperrectangle Ω , a lower bound for Problem MDE over this set Ω is computed via the following linear program, and then we proceed as outlined in steps (6a) - (6e).

$$\mathbf{LP}(\Omega): \underset{(q, Q)}{\text{Minimize}} \{[f(q)]_L : (q, Q) \in X_L(\Omega)\}. \quad (6.21)$$

Step 6a: Initialization. Let $q^* = \hat{q}$ be the incumbent solution with objective function value $\nu^* = \hat{\nu}$. (If no feasible solution is available, let q^* be null and put $\nu^* = \infty$.) Set the iteration counter $r = 1$, and let the set of active nodes (*i.e.*, nodes that yet remain to be

analyzed) be $T_r = \{1\}$, with $t(r) = 1$, and $\Omega^{t(r)} \equiv \Omega$. Solve $\text{LP}(\Omega^{t(r)})$ and let (\tilde{q}, \tilde{Q}) be the optimal solution having an objective function value $LB_{t(r)}$. If \tilde{q} is an improving, feasible solution for Problem MDE, we can update the incumbent solution q^* and the corresponding objective function value ν^* . If $LB_{t(r)} + \varepsilon \geq \nu^*$, for some optimality tolerance $\varepsilon \geq 0$, then stop; q^* is ε -optimal to Problem MDE. Otherwise, select a branching index $w \in \{1, \dots, m\}$ for partitioning the current active node as follows. First, find $K^* \subseteq \bar{M}$, such that

$$K^* \in \arg \max_{\substack{K \subseteq \bar{M} \\ 2 \leq |K| \leq \Delta}} \left| \tilde{Q}_K - \prod_{k \in K} \tilde{q}_k \right|. \quad (6.22a)$$

Then, for each $k \in K^*$, let n_k be the number of times k appears in K^* , and accordingly, select

$$w \in \arg \max_{k \in K^*} \left| \tilde{Q}_{K^*} - \tilde{q}_k^{n_k} \tilde{Q}_{K^* - \{n_k \text{ occurrences of } k\}} \right|. \quad (6.22b)$$

Proceed to Step 6b.

Step 6b: Partitioning Step. Partition the selected active node $t(r)$ (where $t(r) \in T_r$), corresponding to the hyperrectangle $\Omega^{t(r)}$, into two subnodes, indexed by $2r$ and $(6r + 1)$, corresponding to the subhyperrectangles Ω^{2r} and Ω^{2r+1} , respectively, by splitting the current bounds on q_w at the value \tilde{q}_w . Update T_r by adding these two subnodes and deleting the parent node $t(r)$.

Step 6c: Bounding Step. Solve $\text{LP}(\Omega^{2r})$ and $\text{LP}(\Omega^{2r+1})$ corresponding to each of the two new nodes generated, in order to derive respective lower bounds LB_{2r} and LB_{2r+1} . Update the incumbent solution if an improving feasible solution to Problem MDE as detected in this process, and determine a branching variable for each node as necessary, as described in Step 6a.

Step 6d: Fathoming Step. Update $T_{r+1} = T_r - \{t \in T_r : (LB_t + \varepsilon) \geq \nu^*\}$. If $T_{r+1} = \emptyset$, then stop; the incumbent solution is (ε) -optimal. Otherwise, increment r by one and proceed to Step 6e.

Step 6e: Node Selection Step. Select an active node $t(r) \in \arg \min \{LB_t : t \in T_r\}$, and return to Step 6b.

As proven in Sherali and Tuncbilek (1992), this algorithm (when run with $\varepsilon = 0$) will either terminate finitely with a global optimal solution to Problem MDE, or else, will generate an infinite branch-and-bound tree such that any accumulation point of the solutions to the linear lower bounding problems solved along any infinite branch will solve Problem MDE. The proof exhibits that any such branch generates a sequence of lower bounds for Problem MDE that converges to the best known upper bounding solution value, which is therefore a global optimum. (Refer Theorem 6.3, stated below, for a more formal statement of this convergence process.)

Case of $p = \infty$

For the case when $p = \infty$, the objective function (6.13a) turns out to be the weighted l_∞ -norm given by,

$$\text{Minimize}_q \left[\text{maximum}_{k=1, \dots, m} \{c_k |q_k - \bar{q}_k|\} \right].$$

In this case, we can reformulate Problem MDE as follows:

$$\text{Minimize } \xi, \tag{6.23a}$$

$$\text{subject to } \xi \geq c_k (q_k - \bar{q}_k), \forall k = 1, \dots, m, \tag{6.23b}$$

$$\xi \geq c_k (\bar{q}_k - q_k), \forall k = 1, \dots, m, \tag{6.23c}$$

$$\text{Constraints (6.13b) and (6.13c),} \tag{6.23d}$$

where $c_k \equiv 1/w_k$ in the present context.

Note that the procedure for computing tighter lower and upper bounds for the q variables, as described in Step 5 of the algorithm presented for the case of $p = 2$, can similarly be executed here by minimizing and maximizing each $q_k, \forall k = 1, \dots, m$, in turn, over the set:

$$X_L(\Omega) \cap \{(q, Q) : c_k (q_k - \bar{q}_k) \leq \hat{v} \text{ and } c_k (\bar{q}_k - q_k) \leq \hat{v}, \forall k = 1, \dots, m\}$$

in lieu of (6.20). Likewise, the same branch-and-bound algorithm prescribed in Step 6 for the case of $p = 2$ can be adopted, where the lower bounding problem (6.21) is now replaced by

$$\mathbf{LP}(\Omega): \underset{(\xi, q, Q)}{\text{Minimize}} \left\{ \begin{array}{l} \xi : \xi \geq c_k (q_k - \bar{q}_k), \quad \forall k = 1, \dots, m \\ \xi \geq c_k (\bar{q}_k - q_k), \quad \forall k = 1, \dots, m \\ (q, Q) \in X_L(\Omega) \end{array} \right\}. \quad (6.24)$$

Case of $p \geq 4, p$ even

In this case, we can adopt exactly the same procedure as for the case of $p = 2$, except that we consider the order of the polynomial terms in the objective function as well, *i.e.*, we replace Δ by $\max\{\Delta, p\}$. However, if p happens to be appreciably larger than the original Δ (by, say, four or more), then we can adopt the procedure outlined below for the case of a general p , where $1 < p < \infty$.

Case of general $p, 1 < p < \infty$

For the case of a general p , where $p \in (1, \infty)$ is any real number, we can adopt the following strategy, where the objective function for Problem MDE is given by:

$$\text{Minimize} \quad \sum_{k=1}^m c_k |q_k - \bar{q}_k|^p.$$

Here, Steps 1 - 4 corresponding to the case of $p = 2$ remain the same, but we skip Step 5. Furthermore, in the branch-and-bound algorithm of Step 6, in lieu of (6.21) we use a lower bounding problem that is derived as described below.

Note that for this case, to begin with, we can equivalently rewrite Problem MDE as follows, where $y_k \equiv |q_k - \bar{q}_k|, \forall k$.

$$\mathbf{MDE}: \quad \text{Minimize} \quad \sum_{k=1}^m x_k \quad (6.25a)$$

$$\text{subject to} \quad x_k = c_k y_k^p, \quad \forall k = 1, \dots, m, \quad (6.25b)$$

$$y_k \geq q_k - \bar{q}_k, \quad \forall k = 1, \dots, m, \quad (6.25c)$$

$$y_k \geq \bar{q}_k - q_k, \quad \forall k=1, \dots, m, \quad (6.25d)$$

$$\text{Constraints (6.13b, 2.13c),} \quad (6.25e)$$

$$y_{kl} \leq y_k \leq y_{ku}, \quad (6.25f)$$

where,

$$(y_{kl}, y_{ku}) = \left\{ \begin{array}{ll} (0, \max\{\bar{q}_k - q_{kl}, q_{ku} - \bar{q}_k\}), & \text{if } q_{kl} \leq \bar{q}_k \leq q_{ku} \\ (q_{kl} - \bar{q}_k, q_{ku} - \bar{q}_k), & \text{if } \bar{q}_k < q_{kl} \\ (\bar{q}_k - q_{ku}, \bar{q}_k - q_{kl}), & \text{if } \bar{q}_k > q_{ku} \end{array} \right\}, \quad \forall k=1, \dots, m. \quad (6.25g)$$

Referring to Figure 6.2, note that we can derive a polyhedral outer-approximation to the convex function $x_k = c_k y_k^p$ over the interval $[y_{kl}, y_{ku}]$ by constructing its concave envelope over this interval, along with several (typically, four) tangential supports as shown in the figure.

Furthermore, since $h_{kl} = c_k y_{kl}^p$ and $h_{ku} = c_k y_{ku}^p$, this yields,

$$x_k \leq h_{kl} + \frac{(y_k - y_{kl})}{(y_{ku} - y_{kl})} (h_{ku} - h_{kl}), \quad (6.26a)$$

$$x_k \geq c_k \bar{y}_k^p + (y_k - \bar{y}_k) (c_k p \bar{y}_k^{p-1}), \quad \text{for } \bar{y}_k = y_{kl} + \lambda [y_{ku} - y_{kl}], \quad (6.26b)$$

where $\lambda = 0, 1/3, 2/3$, and 1.

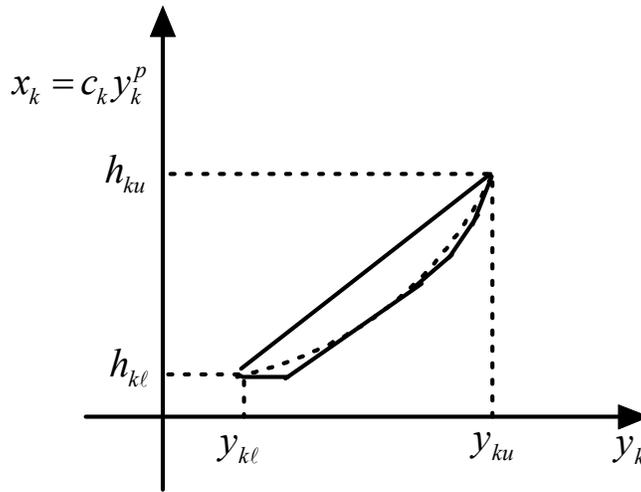


Figure 6.2: Polyhedral Outer-approximation for $x_k = c_k y_k^p$

The lower bounding problem $LP(\Omega)$ can now be defined as follows.

$$\mathbf{LP}(\Omega): \text{ Minimize } \sum_{k=1}^m x_k, \quad (6.27a)$$

$$\text{subject to } (6.26a, 2.26b, 2.25c, 2.25d, 2.25f), \quad (6.27b)$$

$$(q, Q) \in X_L(\Omega). \quad (6.27c)$$

In order to guarantee convergence, taking a leaf from Sherali and Wang (2001), we modify the branching rule (6.22) by selecting a branching variable according to the following rule, where \tilde{q} is defined in Step 6a, with $LP(\Omega)$ given by (6.27):

$$w \text{ is selected as in (6.22) if } \frac{\min\{\tilde{q}_w - q_{wl}, q_{wu} - \tilde{q}_w\}}{(q_{wu} - q_{wl})} \geq 0.1, \quad (6.28a)$$

and otherwise, we select

$$w \in \arg \max_{k=1, \dots, m} \{w_{ku} - w_{kl}\}. \quad (6.28b)$$

Then, for the partitioning step, if w is as given by (6.22), according to the case (6.28a), we split the interval for q_w at the value \tilde{q}_w to create the two children subnodes. However, if w is given by (6.28b), then we split the current interval for q_w at its midpoint. The remainder of the algorithm is as aforementioned for the case of $p = 2$.

Theorem 6.3. (Main Convergence Result). *The above algorithm (run with $\varepsilon \equiv 0$ for any of the discussed cases of p) either terminates finitely with the incumbent solution being (globally) optimal to Problem MDE, or else an infinite sequence of iterations is generated such that along any infinite branch of the branch-and-bound tree, any accumulation point of the q -variable part of the sequence of linear programming relaxation solutions obtained for the node subproblems solves Problem MDE. Moreover if the algorithm is executed with $\varepsilon > 0$, then it will terminate finitely with an ε -optimal solution.*

Proof: See Sherali and Tuncbilek (1992) and Sherali and Wang (2001). \square

6.4. Computational Experience

For the purpose of illustration, consider solving Problem MDE corresponding to a least squares minimization process, given the polynomial function specified in Example 1 below.

Example 1. The characteristic equation of the uncertain feedback control system presented in Example 5.1 of Bhattacharya *et al.* (1995) is obtained as follows:

$$Q(q, s) = \sum_{j=0}^6 a_j(q) s^j, \text{ where } q = [q_1, q_2, q_3, q_4],$$

where, in expanded form, we have,

$$\begin{aligned} Q(q, s) = & s^6 + (5 + q_2)s^5 + (5q_2 + 10 + q_4)s^4 + (21.1 + 6q_2 + 2q_4 + q_2q_4)s^3 \\ & + (25.2 + q_1 + q_3 + 2q_2q_4 + 4q_4 + 0.1q_2)s^2 \\ & + (q_1 + q_3 + q_1q_3 + 8q_4 + 0.2q_2 + 0.4)s + (q_1q_3 + 0.8). \end{aligned} \quad (6.29)$$

The other parameters defining the objective function (6.13a) are the constant nominal vector $\bar{q} = [\bar{q}_1, \bar{q}_2, \bar{q}_3, \bar{q}_4] = [3, 2, 5, 2]$, and the weights $w_k = 1$, $\forall k = 1, \dots, 4$. Since $p = 2$, this yields $c_k = 1, \forall k = 1, \dots, 4$. Furthermore, let the sweeping function $u(z) = zi$, where $z \in [0, 10]$, and $i = \sqrt{-1}$.

As described in the derivation of Problem MDE, substituting the sweeping function $s = u(z)$ in (6.29), and separating the real and imaginary parts of the equation, we get

$$Q(q, u(z)) = Q_R(q, z) + i Q_I(q, z) = \sum_{j=0}^3 \alpha_{2j}(z) a_{2j}(q) + i \sum_{j=0}^2 \alpha_{2j+1}(z) a_{2j+1}(q),$$

where

$$\alpha_{2j}(z) \equiv (iz)^{2j} = (-1)^j z^{2j}, \quad \forall j = 0, \dots, 3, \text{ and where}$$

$$i\alpha_{2j+1}(z) \equiv (iz)^{2j+1} = i(-1)^j z^{2j+1}, \quad \forall j = 0, \dots, 2.$$

This yields,

$$Q(q, u(z)) = \left\{ \begin{array}{l} [-z^6 + (5q_2 + 10 + q_4)z^4 - (25.2 + q_1 + q_3 + 2q_2q_4 + 4q_4 + 0.1q_2)z^2 + (q_1q_3 + 0.8)] \\ + i [(5 + q_2)z^5 - (21.1 + 6q_2 + 2q_4 + q_2q_4)z^3 + (q_1 + q_3 + q_1q_3 + 8q_4 + 0.2q_2 + 0.4)z] \end{array} \right\}. \quad (6.30)$$

Problem MDE given by (6.13) can then be stated as follows:

$$\text{Minimize } f(q) = \sum_{k=1}^4 (q_k - \bar{q}_k)^2 \quad (6.31a)$$

subject to

$$[-z^6 + (5q_2 + 10 + q_4)z^4 - (25.2 + q_1 + q_3 + 2q_2q_4 + 4q_4 + 0.1q_2)z^2 + (q_1q_3 + 0.8)] = 0 \quad (6.31b)$$

$$[(5 + q_2)z^5 - (21.1 + 6q_2 + 2q_4 + q_2q_4)z^3 + (q_1 + q_3 + q_1q_3 + 8q_4 + 0.2q_2 + 0.4)z] = 0 \quad (6.31c)$$

$$(q_1, q_2, q_3, q_4) \in \mathbb{R}^4.$$

We now solve (6.31) corresponding to different values of the sweeping variable z , where $z \in [0, 10]$. Since, the location of the global minimum depends upon the discretization of the sweeping variable z , a finer discretization leads to a more accurate location of the global minimum. For this purpose, we varied z from 0.5 to 2.0 in steps of 0.01, and for each case, the optimal objective function value and corresponding optimal q -values were recorded. Table 6.1 displays these results pertaining to a few select values of z . A global minimum to Example 1, for the case of $p = 2$, occurs at (about) $z = 1.71$ and a local minimum occurs at (about) $z = 1.18$, with objective function values of 1.5682 and 1.6549, respectively. These are indicated as shaded rows in Table 6.1.

Figure 6.3 displays the objective function value as a function of the sweeping variable z . Moreover, since $a_6(q) = 1$ in Equation (6.29), from the optimization problem defined by (6.10), we get $\eta_p = \infty$. Hence, from Theorem 6.2, we get that the optimum occurs at $z = 1.71$, as gleaned from the parametric plot. As an alternative strategy, in lieu of obtaining the optimal z value from a parametric plot, namely Figure 6.3 we could instead consider z to be an additional variable in Problem (6.31), and obtain the optimal value of z directly by solving the corresponding optimization problem. Note, however,

z	q_1	q_2	q_3	q_4	Objective Value
0.5	1.116	2.392	4.251	-0.375	9.9049
0.6	1.605	2.605	4.263	-0.219	7.7825
0.7	2.095	2.785	4.403	-0.004	5.8070
0.8	2.563	2.863	4.635	0.251	4.1281
0.9	2.989	2.812	4.905	0.515	2.8720
1.0	3.350	2.645	5.167	0.767	2.0884
1.1	3.633	2.402	5.388	0.993	1.7257
1.18	3.798	2.176	5.520	1.154	1.6549
1.2	3.830	2.119	5.550	1.193	1.6575
1.3	3.942	1.830	5.647	1.370	1.7313
1.4	3.966	1.553	5.675	1.530	1.8099
1.5	3.900	1.299	5.632	1.681	1.8020
1.6	3.734	1.073	5.514	1.834	1.6897
1.7	3.445	0.872	5.317	2.002	1.5714
1.71	3.408	0.853	5.294	2.021	1.5682
1.8	2.997	0.694	5.042	2.209	1.7517
1.9	2.304	0.533	4.739	2.498	2.9535
2.0	1.189	0.389	4.710	2.875	6.7253
2.1	-0.229	0.289	5.406	3.127	14.7898
2.2	-1.510	0.235	6.362	3.327	27.0710
2.3	-2.666	0.202	7.339	3.587	43.3234
2.4	-3.762	0.180	8.319	3.908	63.6965
2.5	-4.834	0.163	9.307	4.279	88.4878

Table 6.1: Optimal q -values and objective values for different (selected) values of z .

that the complexity of the problem naturally increases, where Problem (6.31) now becomes a polynomial program of order six, instead of being a quadratic program. Solving this expanded problem via the same algorithm, the resulting optimal z -value was found to be $z^* = 1.715$, with the corresponding objective value being 1.5679. The optimal q -vector was obtained as $q^* = [3.391, 0.844, 5.281, 2.029]$. The stability margin is then computed, via Equation (6.9a), as $\rho(z^*) = (1.5679)^{1/2} = 1.252$.

To provide a further validation for our approach, we performed the following experiment. We generated 5000 random points in the hypersphere (6.8), centered at $q = \bar{q}$, having a radius of $b = 1.252 = \rho(z^*)$. The roots of the characteristic polynomial at these points are plotted as shown in Figure 6.4. It is evident that the root locus lies in

the left-half of the plane, and marginally touches the border of the region (the imaginary axis), which shows the tightness of the computed bound.

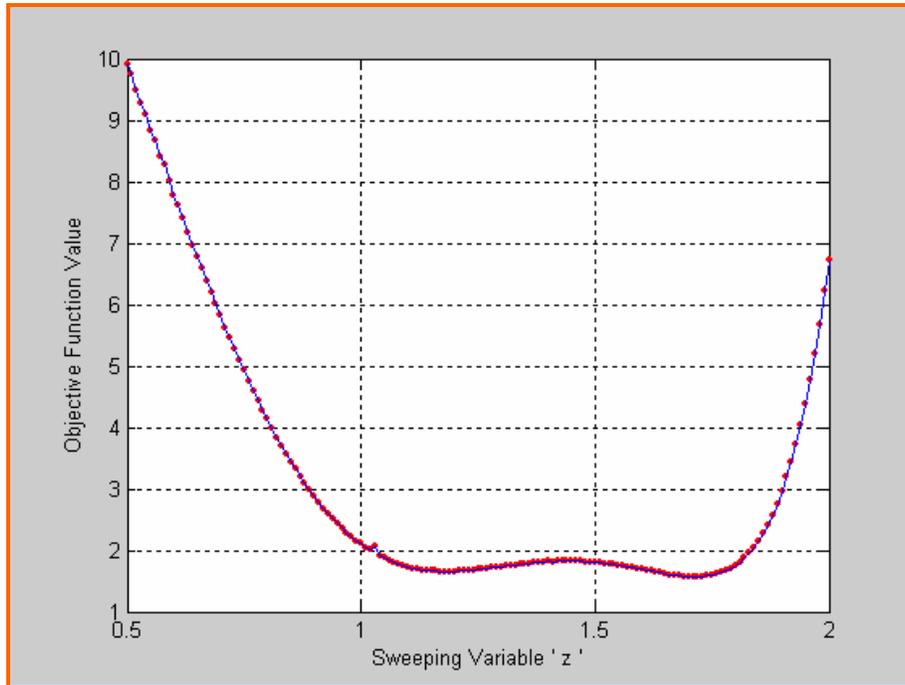


Figure 6.3: Optimal objective value as a function of z for Example 1.

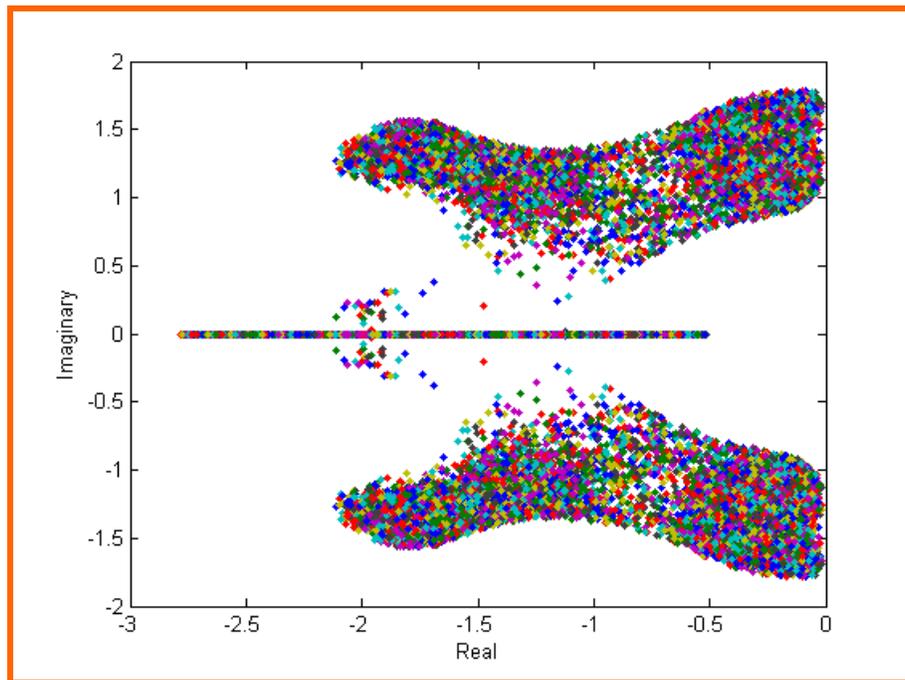


Figure 6.4: Graph displaying the root-locus of the 5000 randomly generated points for Example 1.

Having demonstrated the efficacy of the proposed optimization approach in determining global optimal solutions to the D -stability margin problem, we now present computational experience with respect to producing D -stability margins corresponding to different values of p . Using three different examples, we demonstrate the efficiency of the RLT-based branch-and-bound methodology towards determining global optimal solutions, even in cases when the commercial global optimizer BARON (see Sahinidis, 1996) fails to do so. (Note that BARON adopts a similar branch-and-bound process algorithmic framework, but computes lower bounds differently via generally nonlinear convex programming relaxations.) The related results and associated insights are presented next.

First, we solved Example 1 for several values of p that were chosen to reflect all the different formulations and algorithms derived in Section 6.3. Additionally, since the proposed approach for the case of a general p can be used for the case when p is even, we compared the optimal solutions obtained for even values of p , using both the even and general formulations, respectively. Table 6.2 displays the optimal solutions obtained for Example 1 via the RLT-based branch-and-bound algorithm for all these different cases.

Minimization Type	p -value	Optimal Objective Value	BARON Objective Value	CPU* (s)	CPU _{BARON} (s)	z^*	q^*
Least squares	2	1.5679	1.5679	0.004	0.004	1.715	[3.391, 0.844, 5.281, 2.029]
Even treated as general	2	1.5680	1.5680	0.006	0.006	1.715	[3.391, 0.844, 5.281, 2.029]
General	3	1.2091	1.2091	0.007	0.008	1.155	[3.754, 2.351, 5.601, 1.196]
Even	4	0.8725	0.8725	0.006	0.006	1.143	[3.740 2.442, 5.634, 1.219]
Even treated as general	4	0.8725	0.8725	0.010	0.009	1.143	[3.740 2.442, 5.634, 1.219]
General	5	0.6270	0.6270	0.007	0.007	1.136	[3.734, 2.496, 5.652, 1.232]
Even	10	0.1184	0.1184	0.006	0.006	1.123	[3.723, 2.606, 5.685, 1.259]
Even treated as general	10	0.1184	0.1185	0.11	0.14	1.123	[3.723, 2.606, 5.685, 1.259]
General	99	2.57×10^{-14}	0.00	0.22	0.26	1.118	[3.725, 2.683, 5.721, 1.290]
Infinite	∞	0.7147	0.7153	0.004	0.004	1.446	[3.715, 1.285, 5.715, 1.285]

Table 6.2: Global optimal solutions for Example 1 corresponding to different values of p .

From the results recorded in Table 6.2, observe that the proposed branch-and-bound algorithm as well as the commercial global optimizer BARON determined (global) optimal solutions for all the problem instances. Moreover, for even values of p , notice that the general- p formulation consistently required greater computational time to determine the optimal solutions as compared with the even- p approach. Evidently, exploiting the special polynomial structure of the problem directly via the RLT leads to tighter linear relaxations, thereby facilitating a faster convergence process. Another observation that consistently holds true is that the computational time required increases with larger values of p . We attribute this to the fact that the polyhedral outer-approximation required in the general formulation gets comparatively weaker as p increases, and thus it takes the branch-and-bound algorithm a longer time to converge. Finally, on an average, note that BARON required 15.55% greater computational time as compared to the proposed approach. These conclusions are further substantiated by the results presented below.

Next, consider the following example, which demonstrates the efficacy of the RLT-based branch-and-bound algorithm towards producing global optimal solutions, even for cases where BARON converges to sub-optimal solutions.

Example 2. The characteristic equation of the example presented in Tan (2002) can be derived as follows:

$$Q(q, s) = \sum_{j=0}^7 a_j(q) s^j, \text{ where } q = [q_1, q_2, \dots, q_{12}], \text{ and where, in expanded form, we}$$

have,

$$\begin{aligned} Q(q, s) = & (0.146q_4q_9)s^7 + [(q_4 + 0.146q_2 + 0.146q_3)q_9 + 0.146q_4q_7]s^6 + \\ & [(1.073q_2 + 1.146q_3 + 0.146q_1)q_9 + (q_4 + 0.146q_2 + 0.146q_3)q_7 + 0.146q_4(q_6 + q_7 + q_9)]s^5 + \\ & \left[(1.146q_1 + 0.5q_2 + q_3)q_9 + (1.073q_2 + 1.146q_3 + 0.146q_1)q_7 + (q_4 + 0.146q_2 + 0.146q_3)(q_6 + q_7 + q_9) \right. \\ & \quad \left. + 0.146q_4(q_6 + q_8) \right] s^4 + \\ & \left[0.3298q_8q_{12} + q_1q_9 + (1.146q_1 + 0.5q_2 + q_3)q_7 + (1.073q_2 + 1.146q_3 + 0.146q_1)(q_6 + q_7 + q_9) \right. \\ & \quad \left. + (q_4 + 0.146q_2 + 0.146q_3)(q_6 + q_8) \right] s^3 + \end{aligned}$$

$$\left[\begin{aligned} &0.34q_5q_{12} + 0.3298q_5(q_{10} + q_{11} + q_{12}) + q_1q_7 + (1.146q_1 + 0.5q_2 + q_3)(q_6 + q_7 + q_9) \\ &+ (1.073q_2 + 1.146q_3 + 0.146q_1)(q_6 + q_8) \end{aligned} \right] s^2 +$$

$$\left[\begin{aligned} &0.34q_5(q_{10} + q_{11} + q_{12}) + 0.3298q_5(2q_{10} + q_{11}) + q_1(q_6 + q_7 + q_9) + (1.146q_1 + 0.5q_2 + q_3)(q_6 + q_8) \end{aligned} \right] s +$$

$$0.34q_5(2q_{10} + q_{11}) + q_1(q_6 + q_8).$$

The constant nominal vector and the objective function weights are specified respectively as,

$$\bar{q} = [0.15, 0.45, 0.6, 0.1, 6, 1.5, 4, 0.875, 0.8, 2, 5, 1.5]$$

$$w = [0.1, 0.05, 0.05, 0.01, 0.5, 0.5, 0.5, 0.625, 0.5, 1, 1, 1.3].$$

Once again, the sweeping function is given by $u(z) = zi$, where $z \in [0, 10]$, and $i = \sqrt{-1}$. Performing algebraic manipulations identical to those for Example 1, we get the required polynomial optimization problem for any given value of p . Table 6.3 records the optimal solutions obtained via the proposed algorithm as well as the commercial global optimizer BARON. The superiority of the RLT-based approach is evident from the fact that apart from consistently determining global optimal solutions, the branch-and-bound algorithm requires only 54.61% of the computational time taken by BARON and yet produces solutions that are better (lesser) by 3.16% in terms of the objective function value. In fact, for the cases when $p = 5$ and $p = 99$, BARON requires greater computational effort and yet produces only a local optimal solution for the first instance and no solution at all for the second case. Another noteworthy fact is that the optimal stability margins for Example 2 were determined via the second term in Equation (6.12) for all the runs described in Table 6.3.

As a final exercise, we provide an example wherein the polynomial optimization problem (6.9) is detected to be infeasible, and the required stability margin is therefore determined by the first term in Equation (6.12) as defined by the optimization problem (6.10).

Example 3.

$$Q(q, s) = \sum_{j=0}^8 a_j(q) s^j, \text{ where } q = [q_1, q_2].$$

Minimization Type	p -value	Optimal Objective Value	BARON Objective Value	CPU* (s)	CPU _{BARON} (s)	z^*	q^*
Least squares	2	0.5191	0.5191	0.11	0.11	2.108	[0.158, 0.451, 0.603, 0.10, 5.974, 1.562, 4.044, 0.880, 0.598, 1.719, 4.817, 2.091]
Even treated as general	2	0.5191	0.5191	0.19	0.58	2.108	[0.158, 0.451, 0.603, 0.10, 5.974, 1.562, 4.044, 0.880, 0.598, 1.719, 4.817, 2.091]
General	3	0.1650	0.1650	0.12	0.18	2.066	[0.166, 0.455, 0.606, 0.099, 5.934, 1.60, 4.084, 0.908, 0.625, 1.698, 4.758, 1.999]
Even	4	0.0496	0.0496	0.22	0.58	2.048	[0.17, 0.457, 0.608, 0.098, 5.913, 1.615, 4.102, 0.935, 0.635, 1.698, 4.74, 1.961]
Even treated as general	4	0.0496	0.0496	0.22	0.58	2.048	[0.17, 0.457, 0.608, 0.098, 5.913, 1.615, 4.102, 0.935, 0.635, 1.698, 4.74, 1.961]
General	5	0.0146	0.0576	0.27	0.74	2.039	[0.172, 0.458, 0.610, 0.098, 5.90, 1.622, 4.111, 0.955, 0.64, 1.699, 4.732, 1.941]
Even	10	3.032×10^{-5}	0.00	0.17	0.17	2.023	[0.176, 0.461, 0.612, 0.098, 5.877, 1.635, 4.129, 1.002, 0.648, 1.705, 4.72, 1.905]
Even treated as general	10	3.032×10^{-5}	0.00	0.17	0.17	2.023	[0.176, 0.461, 0.612, 0.098, 5.877, 1.635, 4.129, 1.002, 0.648, 1.705, 4.72, 1.905]
General	99	1.267×10^{-9}	Infeasible	1.43	2.20	2.028	[0.230, 0.410, 0.56, 0.092, 5.753, 1.209, 3.598, 0.599, 0.420, 1.263, 4.277, 1.437]
Infinite	∞	0.2884	0.2884	0.22	0.38	2.011	[0.179, 0.464, 0.614, 0.097, 5.856, 1.644, 4.144, 1.055, 0.656, 1.712, 4.712, 1.875]

Table 6.3: Global optimal solutions for Example 2 corresponding to different values of p .

Here, the polynomial coefficients, $a_j(q)$, $\forall j = 0, \dots, 8$, are given as:

$$\begin{aligned}
a_8(q) &= q_1^2 q_2^2, \quad a_7(q) = 50 q_1^2 q_2^2 + 1080 q_1 q_2, \\
a_6(q) &= 1.25 \times 10^3 q_1^2 q_2^2 + 16.8 q_1^2 q_2 + 53.9 \times 10^3 q_1 q_2 + 270 \times 10^3, \\
a_5(q) &= 15.6 \times 10^3 q_1^2 q_2^2 + 840 q_1^2 q_2 + 1.35 \times 10^6 q_1 q_2 + 13.5 \times 10^6, \\
a_4(q) &= 1.45 \times 10^6 q_1^2 q_2 + 16.8 \times 10^6 q_1 q_2 + 338 \times 10^6, \\
a_3(q) &= 6.93 \times 10^6 q_1^2 q_2 + 911 \times 10^6 q_1 + 4220 \times 10^6, \\
a_2(q) &= 5.72 \times 10^6 q_1^2 q_2 + 113 \times 10^6 q_1^2 + 4250 \times 10^6 q_1, \\
a_1(q) &= 528 \times 10^6 q_1^2 + 3640 \times 10^6 q_1, \quad \text{and } a_0(q) = 453 \times 10^6 q_1^2.
\end{aligned}$$

The constant nominal parameter and the objective function weights are given by $\bar{q} = [15.25, 20.975]$ and $w = [24.5, 22.05]$, respectively. The sweeping function is a hyperbola defined by $u(z) = z + i \sqrt{25z^2 - (1.75)^2}$, where $z \in [-5, -0.35]$, and $i = \sqrt{-1}$.

Substituting $s = u(z)$, and pursuing the same procedure as outlined in the previous examples, we can construct a polynomial optimization problem for any given value of p . For the data given in Example 3, noting the complexity of the problem in the (q, z) space, which limits solving this problem directly via an expanded polynomial program of degree twelve, we employed the parametric plot approach for solving Problem MDE. However, our algorithm discovered that the optimization problem defined by (6.9) is infeasible for this example, for all values of z . (This can be established in the branch-and-bound scheme when the list of active nodes is empty and there is no incumbent solution obtained.) Hence, for any p , the stability margin is therefore determined via the optimization problem (6.10), which is independent of the sweeping variable z . For the case of $p = 2$, this yields,

$$\text{Minimize } f(q) = \sum_{k=1}^2 c_k (q_k - \bar{q}_k)^2 \quad (6.32a)$$

$$\text{subject to } q_1^2 q_2^2 = 0 \quad (6.32b)$$

$$(q_1, q_2) \in \mathbb{R}^2,$$

where $c_k = 1/(w_k)^p$, $k = 1, 2$.

The optimal objective function value is equal to 0.3874, corresponding to the solution $q^* = [0, 20.975]$. Clearly, this solution is optimal for all values of p , with another local minimum occurring at the solution $q = [15.25, 0]$.

6.5. Discussions and Conclusions

In this chapter, we presented a global optimization algorithm for determining parameter stability margins for uncertain linear time invariant systems, where the coefficients of the characteristic equation of the system are defined as polynomial functions of the uncertain parameters. The associated stability margin problem was posed as a problem of computing the maximum size of a hypersolid, defined with respect to various l_p -norms, where $p \in (1, \infty)$. An underlying polynomial programming problem was constructed and a tight linear programming relaxation was derived using the RLT methodology. This relaxation was then embedded in a (convergent) branch-and-bound scheme to determine global optimal solutions. Three test examples with different values of p were utilized to illustrate the efficacy of the proposed methodology for finding global optimal solutions, and the superiority of RLT-based approach over the commercial global optimizer BARON was demonstrated.

7. Conclusions and Future Research

7.1. Summary and Conclusions

While efficient solution techniques have been developed for nearly all types of convex programming problems, optimization research is yet to address the more difficult class of nonconvex optimization problems. These problems are often difficult to solve, primarily because most algorithms tend to gravitate towards local optimal solutions, or get quagmired in searching for solutions along non-improving directions in the search space. Hence, the primary focus of this dissertation has been on employing the broadly applicable RLT methodology in conjunction with problem-specific techniques, to develop tight model formulations and solution methodologies for different classes of nonconvex optimization problems that arise in a host of applications such as hard and fuzzy clustering problems, risk management problems, and problems encountered in control systems design. The underlying structure of many of these nonconvex problems conforms with that of polynomial or factorable programming problems, thereby facilitating an application of the RLT methodology.

The field of cluster analysis is primarily concerned with the sorting of data points into different clusters so as to optimize a certain criterion. There are essentially two types of clustering problems addressed in the literature: *hard clustering*, where each data point is to be assigned to exactly one cluster (refer Späth, 1980), and *fuzzy clustering* where the data points are assigned grades of membership on $[0, 1]$ with respect to different identified clusters (see Höppner *et al.*, 1999).

The hard clustering problem can be defined as follows. Given a set of n data points, each having some s attributes, we are required to assign each of these points to exactly one of some c clusters (where c is given), so as to minimize the total squared Euclidean distance between the data points and the centroid of the clusters to which they are assigned. That is to say, if data point i , having a location descriptor $a_i \in \mathbb{R}^s$ is assigned to cluster j having a to-be-determined centroid $z_j \in \mathbb{R}^s$, then the associated penalty is assumed to be proportional to the square of the straight line distance separation between a_i and z_j in \mathbb{R}^s . This results in an objective function, given by $\sum_{i=1}^n \sum_{j=1}^c w_{ij} \|a_i - z_j\|^2$, where w_{ij} is a binary variable that takes

on a value of 1 if data point i is assigned to cluster j , and 0 otherwise. The product of the w - and z -variables in this function render the problem nonconvex in nature, and difficult to solve to global optimality.

We designed an RLT-based approach for solving this hard clustering problem that includes the generation of additional valid inequalities based on approximations to the convex hull of the data points, along with symmetry-defeating strategies. A tight equivalent 0-1 linear mixed-integer programming representation is derived and a specialized branch-and-bound algorithm is designed to determine a global optimal solution. Results based on computational experiments performed using standard as well as synthetically generated data sets establish the efficacy and robustness of the proposed approach, in contrast with the popular k -means algorithm (Forgy, 1965, McQueen, 1967), as well as in comparison with the global optimization package BARON (see Sahinidis, 1996). *Specifically, the RLT-based branch-and-bound algorithm dominated BARON in terms of both CPU time and quality of the resulting solution (objective function value) by 34.3% and 26.5%, respectively. With regard to the k -means heuristic, even a simple rounding scheme applied to the node-zero solution for the proposed approach itself outperformed the k -means solution by 17.2% and 13.3% in terms of CPU time and objective function value, respectively.* Note that in practice, cluster analysis problems can involve very large data sets, and the results in this work suggest that designing heuristic methods based on constructs that are borrowed from strong effective exact procedures might be a prudent approach for addressing such problems.

Continuing in the same vein as in the case of hard clustering, we proposed an RLT-based optimization approach to solve the fuzzy clustering problem where the objective function in this case is given by $\sum_{i=1}^n \sum_{j=1}^c w_{ij}^2 \|a_i - z_j\|^2$ based on a quadratic degree of fuzziness (see Kamel and Selim, 1994). It was shown that this problem can be equivalently reduced to a cubic nonconvex polynomial program, for which a specialization of the RLT methodology was designed in concert with additional valid inequalities and symmetry-defeating constraints. *On an average, for data sets involving three and five cluster centers, the proposed approach required only 14.05% and 9.85% of the CPU time taken by the FCMA, respectively, and yet yielded solutions that were respectively superior by 69.32% and 77.88% in terms of objective function value. In contrast, using the commercial software BARON to directly solve the nonconvex program resulted in suboptimal solutions*

deteriorating the objective function values by 28.53% and 53.99%, while consuming an additional 50.80% and 45.43% of CPU time as compared to the proposed approach.

The second portion of this dissertation dealt with the applications of factorable programming problems in the realm of risk management. Specifically, we considered the problem of allocating certain available emergency response resources to mitigate risks that arise in the aftermath of a hazardous event. This macro-level problem was modeled as a nonconvex factorable program, for which a tight linear programming relaxation was derived by reducing the nonconvex terms in the problem to linearized functions via a suitable concave outer-envelope construction process. Subsequently, this relaxation is embedded within a specialized branch-and-bound procedure and the overall proposed methodology is proven to converge to a global optimum. Computational experience was provided for a hypothetical case scenario based on different parameter inputs and alternative theoretically convergent branch-and-bound strategies. *The results exhibited that, while consuming comparable computational effort, our algorithm significantly outperforms BARON as well as an ad-hoc intuitive method, respectively, by yielding optimal solutions that are, on an average, better by margins of 14.6% and 17.4%.* Moreover, sensitivity analyses conducted with respect to the equity parameters reveal that the proposed approach also yielded relatively more equitable allocations when compared with these alternative methods.

Next, we considered the strategic planning problem of allocating certain available preventive and mitigation resources to respectively reduce the failure probabilities of system safety features and the total expected loss, arising in the aftermath of a hazardous event. A novel modeling strategy, based on an event tree optimization approach was devised to cast this micro-level cascading risk scenario problem as a nonconvex factorable program. A tight linear programming relaxation was derived using a polyhedral outer-approximation process. Several theoretical insights that serve to lay the foundation for designing a specialized branch-and-bound procedure that is proven to converge to global optimality were derived. *Computational experience reported for a hypothetical case scenario based on different parameter inputs and alternative partitioning strategies demonstrated the dominance of the proposed approach versus the commercial global optimizer BARON by more robustly yielding provable optimal solutions that are, on an average, better by 14.73% in terms of objective function value, while consuming the same degree of computational effort.*

Finally, we established the applicability of the RLT methodology in solving polynomial programs that arise in the context of robust control systems design. The control of linear systems with uncertain physical parameters has been a main subject of research in the field of control engineering over the last two decades (see Ackermann, 2002). One approach to deal with this problem is to study the behavior of the characteristic equations of these systems, and in particular, to study the effects of uncertain parameters on the location of the roots of these polynomials, and thence on the stability and performance of the system. A fundamental problem in such a robust control context is the calculation of stability margins for parameter perturbations, *i.e.*, determining the maximum allowable perturbation in uncertain parameters of a stable system without losing stability. To compute such D -stability margins, we demonstrated that this problem can be equivalently reduced to the form of minimizing $\sum_{k=1}^m c_k |q_k - \bar{q}_k|^p$, where $1 \leq p < \infty$ is a selected l_p - distance based separation measurement parameter. Reformulating the absolute-valued objective terms, various alternative tailored global optimization procedures were developed for solving the resulting nonconvex polynomial programming problem for different values of the parameter p . Several open test cases from the literature have been solved using this methodology to demonstrate its efficacy. *On an average, the proposed optimization approach required only 54.61% of CPU time taken by BARON and yet yielded solutions that were better in terms of objective function value by 3.16%.*

In conclusion, note that a common theme in the study of the five aforementioned challenging nonconvex factorable programs is the development of tight model formulations and relaxations, and the design of effective algorithmic procedures that are not only theoretically convergent, but also yield a more robust solution methodology in comparison with existing solution procedures. We hope that reformulation-based modeling and algorithmic excerpts from this dissertation are incorporated within global optimization software to make them more robust and effective, thereby advancing the frontiers of nonconvex optimization in both theory and practice.

7.2. Future Research

In this section we present the basis for extending the RLT methodology for solving black-box optimization problems as well to strengthen the RLT-based LP relaxations for

solving polynomial programming problems. Recall that the basic idea behind the RLT approach for solving a nonconvex programming problem is to augment the given nonconvex program by adding bound- and constraint-factor product constraints, linearizing the resulting model to obtain a formulation in a higher dimensional space via the introduction of new RLT variables, and then embedding the resulting LP relaxation mechanism in an appropriate branch-and-bound algorithmic framework to obtain convergence to a global optimum. This RLT process extends the traditional idea of solving nonconvex programs using *valid inequalities*.

Adopting the fundamental RLT philosophy, we delineate in this section a basis to develop an all encompassing framework for solving polynomial, factorable, as well as certain black-box optimization problems. Also, note that one of the drawbacks of the RLT approach is that higher-level RLT relaxations tend to generate a large number of variables and constraints, many of which might prove to be redundant, thereby encumbering the branch-and-bound process. To circumvent this difficulty, various constraint filtering techniques as expounded by Sherali and Tuncbilek (1995) can be utilized. Furthermore, in order to accelerate convergence by virtue of deriving tighter relaxations, a *branch-and-cut* philosophy needs to be employed, wherein the RLT relaxations are tightened by the addition of cutting planes derived from semidefinite programming constructs. This enhancement can (possibly) be done at each node of the branch-and-bound tree to obtain good quality feasible solutions relatively quickly. The procurement of feasible solutions serves the dual purpose of fathoming nodes and/or unexplored branches, as well as updating the current incumbent solution, which speeds up the convergence process. For this purpose, in the latter part of this chapter, we lay the foundation for generating various kinds of cutting planes, derived from the solution of the RLT linear programming relaxations via semidefinite programming constructs.

We begin by formulating a wide class of *nonconvex problems (NCP)*, where the objective function and constraints can be comprised of polynomial, or factorable, or black-box functions. Consider the mathematical formulation of a generic *nonconvex programming problem (NCP)* as given below.

NCP(Ω): Minimize $\{f_0(x) : x \in Z_1 \cap Z_2 \cap Z_3 \cap \Omega\}$

where, $Z_1 = \{x : f_r(x) \leq \beta_r, r = 1, \dots, R_1\}$, $Z_2 = \{x : f_r(x) \leq \beta_r, r = R_1 + 1, \dots, R_2\}$,

$Z_3 = \{\text{set of black-box constraints, defined by functions } \phi_r(x), \forall r = R_2 + 1, \dots, R_3\}$,

$\Omega = \{x : 0 \leq l_j \leq x_j \leq u_j < \infty, j = 1, \dots, n\}$, and where,

$$f_r(x) = \sum_{t \in T_r} \alpha_{rt} \prod_{j \in J_{rt}} f_{rtj}(x_j) \equiv \sum_{t \in T_r} \alpha_{rt} f_{rt}(x), \quad \forall r = 1, \dots, R_1, R_1 + 1, \dots, R_2. \quad (7.1)$$

As defined in (7.1), although the class of factorable programs subsumes the class of polynomial programs, for the purpose of algorithmic simplicity, we separate the sets of polynomial and factorable constraints, and denote Z_1 to be the set of polynomial constraints, and Z_2 as the set of factorable constraints. Therefore, in our representation of $f_r(x)$, $f_{rtj}(x_j) \equiv x_j$, $\forall r = 0, \dots, R_1$, whereas $f_r(x)$, $\forall r = R_1 + 1, \dots, R_2$, includes some *nonpolynomial* term containing a nonpolynomial univariate functions $f_{rtj}(x_j)$. Here, T_r is an index set for the terms defining $f_r(\cdot)$, and α_{rt} are (real) coefficients, $t \in T_r$, $r = 0, \dots, R_2$. Note that a repetition of indices is allowed within J_{rt} . For example, if $J_{rt} = \{1, 2, 2, 3\}$, and $f_{rtj}(x_j) = x_j, \forall j \in J_{rt}$, then the corresponding polynomial term is $x_1 x_2^2 x_3$. Denote $N = \{1, \dots, n\}$ and define $\bar{N} = \{N, \dots, N\}$. Then each $J_{rt} \subseteq \bar{N}$, with $1 \leq |J_{rt}| \leq \delta_1$, for $t \in T_r$, $r = 0, 1, \dots, R_1$, where δ_1 is the maximum specified degree of any polynomial term appearing in NCP(Ω). For the factorable functions defining the region Z_2 , $f_r(x)$ is a nonconvex factorable function that is stated as a sum of terms $\alpha_{rt} f_{rt}(x)$, indexed by $t \in T_r$, $r = R_1 + 1, \dots, R_2$. For each term, $t \in T_r$, $r = R_1 + 1, \dots, R_2$, $f_{rt}(x)$ is a product of twice continuously differentiable, univariate functions $f_{rtj}(x_j)$ of x_j , indexed by $j \in J_{rt} \subseteq N$, at least one of which is nonpolynomial.

Next, the set of black-box constraints defining the set Z_3 represent those restrictions that are analytically complex and are implicitly defined in terms of the decision variables. Suppose that for every black-box function $\phi_r(x)$, $r = R_2 + 1, \dots, R_3$, which defines the set Z_3 , we are given ordered pairs (x^{rk}, v^{rk}) , $k = 1, \dots, m_r$, where $x^{rk} = (x_1^{rk}, \dots, x_n^{rk}) \in \Omega$, and $v^{rk} \equiv \phi_r(x^{rk})$ is the corresponding function value. Using this data, we construct *polynomial*

surrogates representing the original functions (as explained in the sequel), and exploit the functional forms of these surrogates in an RLT-based branch-and-bound algorithmic process. Furthermore, since we have assumed $f_0(x)$ to be a polynomial function, if the objective function happens to be either a factorable or a black-box function, then we symbolize it as f_0 , and introduce the nonpolynomial functional constraint $f_0 \geq f_0(x)$ explicitly into the corresponding constraint set, so as to reduce the given problem into the standard form (7.1). Finally, if *equality* constraints exist in NCP, then we can represent each equality constraint within (7.1) as two less-than-or-equal-to sign-interchanged inequalities.

Recognizing the ability of the RLT to solve polynomial programs to (global) optimality, the basic idea is to transform the given nonconvex program to a series of equivalent polynomial programming approximations via the addition of valid inequalities and using variable substitution strategies. For the black-box functions in NCP, this process would essentially involve a two-step approximation scheme, wherein first, surrogate factorable functions are derived, and then these surrogate functions are coupled with the original factorable terms in the problem. Subsequently, we derive lower/upper bounding polynomial approximations for the *augmented* set of factorable functions, and rely on the demonstrated ability of the RLT for solving the resulting polynomial program. The construction and manipulation of these polynomial approximations needs to be conducted in a manner that achieves convergence of the overall algorithmic scheme to global optimality, assuming a degree of fidelity of the factorable functional approximations to the black-box constraints.

Among available methods for approximating black-box functions, of recent interest is the field of *Response Surface Methodology* (RSM) (refer Myers, 1995), and its application to global optimization algorithms. Response surfaces are gaining popularity as a means of developing fast surrogates for time-consuming computer simulations (Jones, 2001). The appeal of the response surface approach is the fact that statistical analyses, as well as sensitivity analyses of the surrogates to input parameters can be performed relatively easily. Existing approaches that use response surfaces for global optimization can be classified based on the type of response surface being considered, and the method used to select the initial search points. Typically, response surfaces can be differentiated depending on whether they are non-interpolating (minimize the sum of squared errors from a pre-determined functional form), or interpolating (pass through all points).

Traditionally, in order to derive a surrogate polynomial, a *curve-fitting* approach is often used. Note that all curve-fitting methods belong to the class of non-interpolating response surface methods. Here, the functional forms of (typically) either a first-degree or a second-degree polynomial approximation is assumed to fit the data points and the coefficients of the surrogate polynomial are determined by minimizing the sum total of squared errors, which leads to solving a set of simultaneous linear equations in the coefficients of the assumed polynomial. The simplest method that is in vogue is to first fit a quadratic function so as to minimize the total squared error. The minimizing solution for this quadratic function is then computed, and the response surface points are updated. This approach is illustrated in Figure 7.1 for the case of a nonconvex univariate function. Notice that the minimum of the quadratic function misses not only the global minimum of f but also the local minimum as well.

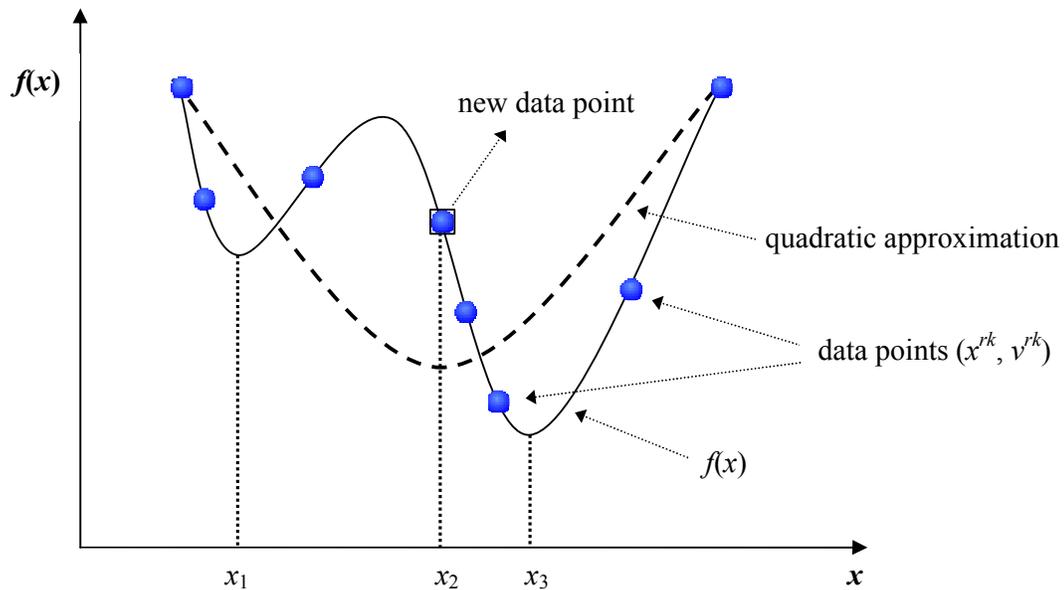


Figure 7.1: A quadratic approximation for a nonconvex univariate function that misses both the global and local optima.

Noting the obvious difficulties involved in non-interpolating methods, interpolating response surface techniques such as cubic splines, multiquadrics, and kriging have come to the fore. The surrogate function obtained using the kriging predictor is an interpolating function that passes through all the data points. Sometimes, as a result of not having a

sufficient number of data points (function evaluations), the response surface obtained might not capture the true shape of the black-box function. In such a case, treating the generated response function as being true, additional function evaluations could be obtained. Then, the response surface parameters can be recomputed, taking both the original as well as newly obtained data points into consideration. This process, when applied recursively, yields the required black-box functional forms. We are currently investigating the generation of surrogate functional forms for the black-box terms in $\text{NCP}(\Omega)$, and the results are forthcoming.

Next, we consider the issue of further enhancing RLT-based LP relaxations for solving polynomial programming constructs via *semidefinite cutting plane* techniques. Given an RLT relaxation, instead of merely imposing nonnegativity constraints on the RLT product variables, we can impose positive semidefiniteness on the variable-product matrix, and correspondingly derive implied semidefinite cuts. In the case of polynomial programming problems, there are several possible variations that can potentially be chosen to form this variable-product matrix on which positive semidefiniteness can be imposed.

To illustrate the underlying concept here, consider the quadratic polynomial programming problem. Note that the new RLT variables in this context are represented by the $n \times n$ matrix $X \equiv [xx^T]_{\text{L}}$, where $[\cdot]_{\text{L}}$ represents the linearization of the expression $[\cdot]$ under the substitutions (2.4). Observe that since xx^T is symmetric and positive semidefinite (denoted $\succeq 0$), we could require that $X \succeq 0$, as opposed to simply enforcing nonnegativity on this matrix. In fact, a stronger implication in this same vein is obtained by considering

$$x_{(1)} = \begin{bmatrix} 1 \\ x \end{bmatrix}, \text{ and defining the matrix } M_1 \equiv [x_{(1)} x_{(1)}^T]_{\text{L}} = \begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix}, \quad (7.2)$$

and requiring that $M_1 \succeq 0$.

In lieu of solving the resulting semidefinite programming relaxations, which would detract from the robustness and efficiency that accrues from relying on LP relaxations, Sherali and Fraticelli (2002) have proposed the use of a class of RLT constraints known as semidefinite cuts that are predicated on the fact that

$$M_1 \succeq 0 \Leftrightarrow \alpha^T M_1 \alpha = [(\alpha^T x_{(1)})^2]_{\mathbb{L}} \geq 0, \forall \alpha \in \mathbb{R}^{n+1}, \|\alpha\| = 1. \quad (7.3)$$

Accordingly, given a certain solution (\bar{x}, \bar{X}) to the RLT relaxation of the underlying quadratic polynomial program for which $\bar{X} \neq \bar{x} \bar{x}^T$ (*i.e.*, the condition of Lemma 1 does not hold true), Sherali and Fraticelli (2002) invoke (7.3) to check in polynomial time having a worst-case complexity $O(n^3)$ whether or not $\bar{M}_1 \succeq 0$, where \bar{M}_1 evaluates M_1 at the solution (\bar{x}, \bar{X}) . In case that \bar{M}_1 is not positive semidefinite, they show that this process also automatically generates an $\bar{\alpha} \in \mathbb{R}^{n+1}$ such that $\bar{\alpha}^T \bar{M}_1 \bar{\alpha} < 0$, which in turn yields the semidefinite cut

$$\bar{\alpha}^T M_1 \bar{\alpha} = [(\bar{\alpha}^T x_{(1)})^2]_{\mathbb{L}} \geq 0. \quad (7.4)$$

Several alternative polynomial-time schemes for generating rounds of cuts (18) based on suitable vectors $\bar{\alpha}$ are described and are computationally exhibited to yield a substantial reduction (by a factor of 2-3) on the class of quadratic programs tested, in comparison with an RLT approach that does not employ such cuts.

We mention here that Konno *et al.* (2003) have proposed using a similar cut of the type $\tilde{\alpha}^T X \tilde{\alpha} \geq 0$, where $\tilde{\alpha} \in \mathbb{R}^n$ is the normalized eigenvector corresponding to the smallest eigenvalue of the matrix \bar{X} , given that \bar{X} is not positive semidefinite. However, computing $\tilde{\alpha}$ can be relatively burdensome, and moreover, the round of cuts (7.4) generated for the augmented matrix M_1 can yield potentially tighter relaxations.

As a further generalization of the RLT procedure, Lasserre (2001, 2002) discussed the generation of tight relaxations for polynomial programming problems using linear matrix inequalities (LMIs). In the spirit of (7.2), let us define $x_{(m)}$ as the augmentation of the vector $x_{(1)}$ with all quadratic terms involving the x -variables, then all such cubic terms, and so on until all possible multinomials of order m . Accordingly, define the moment matrix

$$M_m \equiv [x_{(m)} x_{(m)}^T]_{\mathbb{L}}. \quad (7.5)$$

Then, for the case of an unconstrained polynomial program of the type: Minimize $\{\phi_0(x) : x \in \mathbb{R}^n\}$, where $\phi_0(x)$ is a polynomial of degree δ , Lasserre (2001) considered the

relaxation \mathbf{R}_m given below, where $m \geq \lceil \delta/2 \rceil$, and where $\theta(x) \equiv a^2 - \|x\|^2$, with $a > 0$ being the radius of a ball that is known to contain an optimal solution.

$$\mathbf{R}_m: \text{ Minimize } \{ [\phi_0(x)]_{\mathbb{L}} : M_m \succeq 0, [\theta(x) x_{(m-1)} x_{(m-1)}^{\mathbb{T}}]_{\mathbb{L}} \succeq 0 \}. \quad (7.6)$$

Lasserre (2001) proved that \mathbf{R}_m is asymptotically exact in that as $m \rightarrow \infty$, the optimal value of \mathbf{R}_m approaches that of the underlying polynomial program. In fact, for a *univariate* polynomial program to minimize $\phi_0(x)$ subject to $a \leq x \leq b$, where $\phi_0(x)$ is of odd degree $2m+1$ (some additional manipulation is required for even degree problems), Lasserre (2002) showed that the optimal value is recovered via the semidefinite program

$$\text{Minimize } \{ [\phi_0(x)]_{\mathbb{L}} : [(x-a) x_{(m)} x_{(m)}^{\mathbb{T}}]_{\mathbb{L}} \succeq 0, [(b-x) x_{(m)} x_{(m)}^{\mathbb{T}}]_{\mathbb{L}} \succeq 0 \}. \quad (7.7)$$

For multivariate constrained polynomial programming problems of the type: Minimize $\{ \phi_0(x) : \phi_r(x) \geq \beta_r, r = 1, \dots, R \}$, having a degree δ , a similar relaxation to that in (7.6) was generated as follows, where $m \geq \lceil \delta/2 \rceil$.

$$\mathbf{CR}_m: \text{ Minimize } \{ [\phi_0(x)]_{\mathbb{L}} : M_m \succeq 0, [(\phi_r(x) - \beta_r) x_{(m-\lceil \delta_r/2 \rceil)} x_{(m-\lceil \delta_r/2 \rceil)}^{\mathbb{T}}]_{\mathbb{L}} \succeq 0, \forall r = 1, \dots, R \}, \quad (7.8)$$

where δ_r is the degree of $\phi_r(x)$, $\forall r = 1, \dots, R$. Under certain stringent conditions on the feasible region, Lasserre (2001, 2002) exhibited that the relaxation \mathbf{CR}_m becomes asymptotically exact as $m \rightarrow \infty$.

Based on the approach and experience of Sherali and Fraticelli, and based on the results embodied by the relaxations (7.6), (7.7), and (7.8), note the corresponding LMIs in these problems can be replaced by associated semidefinite cuts of the type (7.3), and results from this investigation will be forthcoming.

References

- [1] Ackermann, J. (2002), *Robust Control: The Parameter Space Approach*, Springer-Verlag, London.
- [2] Ackermann, J., Kaesbauer, D. and Muench, R. (1990), Robust gamma-stability analysis in a plant parameter space, *Automatica* 27, 75-85.
- [3] Adams, W.P., Lassiter, J.B. and Sherali, H.D. (1998), Persistency in 0-1 polynomial programming, *Mathematics of Operations Research* 23(2), 359-389.
- [4] Adams, W.P. and Sherali, H.D. (1986), A tight linearization and an algorithm for zero-one programming problems, *Management Science* 32(10), 1274-1290.
- [5] Aggarwal, A. and Floudas, C.A. (1990), A decomposition approach for global optimum search in QP, NLP, and MINLP problems, *Annals of Operations Research* 25, Special volume on computational methods in global optimization, eds., Pardalos, P.M. and Rosen, J.B.
- [6] Alexandrov, N.M., Dennis, J.E. Jr., Lewis, R.M. and Torczon, V. (1998), A trust-region framework for managing the use of approximation models in optimization, *Structural Optimization* 15, 16-23.
- [7] Al-Sultan, K.S. and Khan, M.M. (1996), Computational experience on four algorithms for the hard clustering problem, *Pattern Recognition Letters* 17, 295-308.
- [8] Amendola, A., Ermoliev, Y., Ermolieva, T.Y., Gitis, V., Koff, G. and Linnerooth-Bayer, J. (2000), A Systems approach to modeling catastrophic risk and insurability, *Natural Hazards* 21, 381-393.
- [9] Andrews, J. D. and Dunnett, S. J. (2000), Event-tree analysis using binary decision diagrams, *IEEE Transactions on Reliability* 49(2), 230-238.
- [10] Atlas, M.K. (2001), Safe and sorry: Risk, environmental equity, and hazardous waste management facilities, *Risk Analysis* 21(5), 939-954.
- [11] Audet C., Brimberg, J., Hansen, P., Le Digabel, S., and Mladenović, N. (2000b), Pooling problem: Alternate formulations and solution methods, Le Cahiers du GERAD, Manuscript G-2000-23, Montreal, Canada.
- [12] Audet, C., Hansen, P., Jaumard, B. and Savard, G. (2000a), A branch and cut algorithm for nonconvex quadratically constrained quadratic programming, *Mathematical Programming* 87, Series A, 131-152.
- [13] Balas, E. (1988), On the convex hull of the union of certain polyhedra, *Operations Research Letters* 7(6), 279-283.
- [14] Balas, E. and Mazzola, J.B. (1984a), Nonlinear 0-1 programming: I. Linearization techniques, *Math Programming* 30, 22-45.
- [15] Balas, E. and Mazzola, J.B. (1984b), Nonlinear 0-1 programming: II. Dominance relations and algorithms, *Math Programming* 30, 22-45.
- [16] Baraldi, A. and Blonda, P. (1999a), Survey of fuzzy clustering analysis for pattern recognition - Part I, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 29(6), 778-785.

- [17] Baraldi, A. and Blonda, P. (1999b), Survey of fuzzy clustering analysis for pattern recognition - Part II, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 29(6), 786-801.
- [18] Barmish, R. B., (1994), *New Tools for Robustness of Linear Systems*, Macmillan, New York, N.Y.
- [19] Bazaraa, M.S., Sherali, H.D. and Shetty, C.M. (1993), *Nonlinear Programming: Theory and Algorithms*, John Wiley & Sons, Inc., 2nd edition, New York, N.Y.
- [20] Beim, G. K. and Hobbs, B.F. (1997), Event tree analysis of lock closure risks, *Journal of Water Resources Planning & Management-ASCE* 123(3), 169-178.
- [21] Belousov, E.G. and Klatte, D. (2002), A Frank-Wolfe type theorem for convex polynomial programs, *Computational Optimization and Applications* 22, 37-48.
- [22] Ben-Tal, A., Eiger, G. and Gershovitz, V. (1994), Global minimization by reducing the duality gap, *Mathematical Programming* 63, 193-212.
- [23] Bezdek, J.C. (1981), *Pattern recognition with fuzzy objective function algorithms*, Plenum Press, New York, N.Y.
- [24] Bhattacharyya, S. P., Chapellat, H. and Keel, L. H. (1995), *Robust Control: The Parametric Approach*, Prentice-Hall, Englewood Cliffs, N.J.
- [25] Bhuyan, J.N., Raghavan, V.V. and Elayavalli, V.K. (1991), Genetic algorithm for clustering with an ordered representation, *Proceedings of the Fourth International Conference on Genetic Algorithms*, San Diego, CA.
- [26] Bier, V.M., Haimes, Y.Y., Lambert, J.H., Matalas, N.C. and Zimmermann, R. (1999), A survey of approaches for assessing and managing the risk of extremes, *Risk Analysis* 19(1), 83-94.
- [27] Booker, A.J., Dennis, J.E. Jr., Frank, P.D., Serafini, D.B., Torczon, V. and Trosset, M.W. (1999), A rigorous framework for optimization of expensive functions by surrogates, *Structural Optimization* 17, 1-13.
- [28] Bozorg, M. and Nebot, E. M. (1999), l^p parameter perturbation and design of robust controllers for linear systems, *International Journal of Control* 72, 267-275.
- [29] Brekelmens, R., Driessen, L., Hamers, H. and den Hertog, D. (2001), Constrained optimization involving expensive function evaluations: A sequential approach, Working paper, CentER, Tilburg, The Netherlands.
- [30] Chang, C.T. and Chang, C.C. (2000), A linearization method for mixed 0-1 polynomial programs, *Computers and Operations Research* 27, 1005-1016.
- [31] Chapellat, H., Keel, L. H. and Bhattacharyya, S. P. (1993), Robust stability manifolds for multilinear interval systems, *IEEE Transactions on Automatic Control* 34, 314-318.
- [32] Chazelle, B. (1991), An optimal convex hull algorithm and new results on cuttings, *Annual Symposium on Foundations of Computer Science*, 29-38.
- [33] Collins, E. W. and Cooley, W. L. (1983), Use of event tree analysis to optimize electrical accident counter-measure systems for metal/nonmetal mines, *Conference Record - IAS Annual Meeting (IEEE Industry Applications Society)*, Piscataway, NJ, 139-151.

- [34] Conn, A.R., Gould, N.I.M. and Toint, P.L. (2000), *Trust Region Methods*, MPS/SIAM Series on Optimization, SIAM, Philadelphia.
- [35] Conn, A.R., Scheinberg, K. and Toint, P.L. (1997), Recent progress in unconstrained nonlinear optimization without derivatives, *Mathematical Programming*, 79(1-3), 397-414.
- [36] De Gaston., R. R. and Safonov, M. G. (1988), Exact calculation of the multiloop stability margin, *IEEE Transactions on Automatic Control* 33, 156-171.
- [37] Desages, C., Castro, L. and Cendra, H. (1991), Distance of a complex coefficient stable polynomial from the boundary of the stability set, *Multidimensional Systems and Signal Processing* 2, 189-210.
- [38] Dey, P.K. (2002), Quantitative risk management aids refinery construction, *Hydrocarbon Processing* 81(3), 85-95.
- [39] Djaferis, T. E. (1995), *Robust Control Design: A Polynomial Approach*, Kluwer Academic Publishers, Dordrecht, The Netherlands.
- [40] Driessen, L., Brekelmens, R., Hamers, H. and den Hertog, D. (2001), On D-optimalitt based trust regions for black-box optimization problems, Working paper, CentER, Tilburg, The Netherlands.
- [41] Dubes, R.C. (1987), How many clusters are best? – an experiment, *Pattern Recognition* 20, 645-663.
- [42] Dunn, J.C. (1973), A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters, *Journal of Cybernetics* 3(3), 32-57.
- [43] Eisenmann, T.R. (2002), The Effects of CEO Equity Ownership and Firm Diversification on Risk Taking, *Strategic Management Journal* 23(6), 513-534.
- [44] Falk, J.E. and Soland, R.M. (1969), An algorithm for separable nonconvex programming problems, *Management Science* 15, 550-569.
- [45] Fenyvesi, L., Rothwell, B., and Colquhoun, I. (2002), Meta-Risk as a Method for Addressing Uncertainty in a Pipeline Risk Management System, *Proceedings of the International Pipeline Conference*, IPC (A), 781-786.
- [46] Fisher, A., Chestnut, L.G. and Violette, D.M. (1988), The value of reducing risks of death: A note on new evidence, *Journal of Policy Analysis and Management* 8(1).
- [47] Floudas, C.A. and Pardalos, P.M. (1990), *A Collection of Test Problems for Constrained Global Optimization Algorithms*, Springer-Verlag, Berlin.
- [48] Floudas, C.A. and Visweswaran, V. (1990a), A global optimization algorithm for certain classes of nonconvex NLPs-I: Theory, *Computers and Chemical Engineering* 14(12), 1397-1417.
- [49] Floudas, C.A. and Visweswaran, V. (1990b), A global optimization algorithm for certain classes of nonconvex NLPs-II: Application of theory and test problems, *Computers and Chemical Engineering* 14(12), 1419-1434.
- [50] Floudas, C.A. and Visweswaran, V. (1995), Quadratic optimization, in: Horst, R. and Pardalos, P.M., eds., *Handbook of Global Optimization*, Kluwer Academic Publishers, Dordrecht, The Netherlands.

- [51] Forgy, E.W. (1966), Cluster analysis of multivariate data: Efficiency versus interpretability of classification, *Biometric Society Meetings*, Riverside, CA, *Abstract in Biometrics* 21, 768.
- [52] Freimut, B., Hartkopf, S., Kaiser, P., Kontio, J. and Kobitzsch, W. (2001), An industrial case study of implementing software risk management, *Proceedings of the ACM SIGSOFT Symposium on the Foundations of Software Engineering*, 277-287.
- [53] GAMS Solver Descriptions (2003), GAMS/OQNLP, www-document, <http://www.gams.com/solvers/solvers.htm#OQNLP>.
- [54] Gath, I. and Geva, A.B. (1989), Unsupervised optimal fuzzy clustering, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11(7), 773-781.
- [55] Geoffrion, A.M. (1972), Generalized Benders' decomposition, *Journal of Optimization Theory and Applications* 10(4), 237-260.
- [56] Gonzalez, M., Castro, J.L.U., Betancourt, C.F. and Rodriguez, E. (2002), Risk management to support maintenance and investment in gas transmission pipelines, *Proceedings of the International Pipeline Conference, IPC (A)*, 787-793.
- [57] Ghotb, F. (1987), Constrained nonlinear optimization with factorable programming, in: Teo, K.L., *Techniques and Applications*, National University of Singapore, Singapore, 842-847.
- [58] Groetschel, M. and Wakabayashi, Y. (1989), Cutting plane algorithm for a clustering problem, *Mathematical Programming Series B* 45(1), 59-96.
- [59] Groupe, A. (1995), Computation of prime implicants of a fault tree within Aralia, *Proceedings of the European Safety and Reliability Association Conference*, 190-202.
- [60] Guoyao, F. (1998), Optimization methods for fuzzy clustering, *Fuzzy Sets and Systems* 93, 301-309.
- [61] Gustafson, D.E. and Kessel, W.C. (1979), Fuzzy clustering with a fuzzy covariance matrix, *International Proceedings of the IEEE Conference on Decision and Control*, 761-766.
- [62] Gutmann, H.-M. (2001), A radial basis function method for global optimization, *Journal of Global Optimization* 19, 201-227.
- [63] Hadipriono, F. C., Lim, C. and Wong, K. (1986), Event tree analysis to prevent failures in temporary structures, *Journal of Construction Engineering & Management - ASCE* 12(4), 500-513.
- [64] Hartigan, J.A. (1975), *Clustering algorithms*, John Wiley and Sons, New York, N.Y.
- [65] Helmberg, C. (2002), Semidefinite programming, *European Journal of Operational Research* 137, 461-482.
- [66] Hinrichsen, D. and Pritchard, A. J. (1989), An application of state space methods to obtain explicit formulae for robustness measures of polynomials, in: *Robustness in Identification and Control*, eds., Milanese, M., Tempo, R. and Vicino, A., Plenum Press, New York, N.Y.
- [67] Höppner, F., Klawonn, F., Kruse, R. and Runkler, T. (1999), *Fuzzy Cluster Analysis*, John Wiley & Sons, Inc., New York, N.Y.

- [68] Horst, R. (1990), Deterministic methods in constrained global optimization: Some recent advances and new fields of application, *Naval Research Logistics Quarterly* 37, 433-471.
- [69] Horst, R. and Pardalos, M.P. eds., (1995), *Handbook of Global Optimization*, Kluwer Academic Publishers, Dordrecht, The Netherlands.
- [70] Horst, R. and Tuy, H. (1993), *Global Optimization: Deterministic Approaches*, Springer-Verlag, 2nd edition, Berlin, Germany.
- [71] Huang, D., Chen, T. and Wang, M. J. (2001), Fuzzy set approach for event tree analysis, *Fuzzy Sets and Systems* 118(1), 153-165.
- [72] Ismail, M.A. and Selim, S.Z. (1986), Fuzzy c-means: optimality of solutions and effective termination of the algorithm, *Pattern Recognition* 19, 481-485.
- [73] Jensen, R.E. (1969), A dynamic programming algorithm for cluster analysis, *Operations Research* 17, 1034-1057.
- [74] Jin, C. L., Yan, J. and Zhou, S. (2003), Application of event tree analysis based on fuzzy sets in risk analysis, *Journal of Dalian University of Technology* 43(1), 97-100.
- [75] Jones D.R. (2001), A taxonomy of global optimization methods based on response surfaces, *Journal of Global Optimization* 21(4), 345-383.
- [76] Jones, D.R., Pertunnen, C.D. and Stuckmann, B.E. (1993), Lipschitzian optimization without the Lipschitz constant, *Journal of Optimization Theory and Applications* 79, 157-181.
- [77] Jones, D.R., Schonlau, M. and Welch, W.J. (1998), Efficient global optimization of expensive black-box functions, *Journal of Global Optimization* 13, 455-492.
- [78] Joshi, S.S., Sherali, H.D. and Tew, J.D. (1998), An enhanced response surface methodology algorithm using gradient deflection and second-order search strategies, *Computers and Operations Research* 25(7/8), 531-541.
- [79] Jung, Y., Park, H., Du, Z. and Drake, B.L. (2003), A decision criterion for the optimal number of clusters in hierarchical clustering, *Journal of Global Optimization* 25, 91-111.
- [80] Kafka, P. (2002), Reflections on the status of risk management in nuclear technology, *European Space Agency (Special Publication)*, ESA SP 486, 403-410.
- [81] Kamel, M.S. and Selim, S.Z. (1994), New algorithms for solving the fuzzy clustering problem, *Pattern Recognition* 27(3), 421-428.
- [82] Kaplan, S. (1982), Matrix theory formalism for event tree analysis: Application to nuclear-risk analysis, *Risk Analysis* 2(1), 9-18.
- [83] Keel, L.H. and Bhattacharyya, S. P. (1993), Parametric stability margin for multilinear interval control systems, *Proceedings of American Control Conference*, San Francisco, California, 262-266.
- [84] Kenarangui, R. (1991), Event-tree analysis by fuzzy probability, *IEEE Transactions on Reliability* 40(1), 120-124.
- [85] Klapper, A. (1987), Lower bound on the complexity of the convex hull problem for simply polyhedra, *Information Processing Letters* 25(3), 159-161.

- [86] Klawonn, F. and Keller, A. (1998), Fuzzy clustering based on modified distance measures, *Advances in Intelligent Data Analysis, Proceedings of the 3rd International Symposium*, eds. Hand, D.J., Kok, J.N. and Berthold, K.R., 291-301, Amsterdam, The Netherlands.
- [87] Klein, R.W. and Dubes, R.C. (1989), Experiments in projection and clustering by simulated annealing, *Pattern Recognition* 22, 213-220.
- [88] Konno, H., Kawada, N. and Tuy, H. (2003), Cutting plane algorithms for nonlinear semidefinite programming problems with applications, *Journal of Global Optimization* 25, 141-155.
- [89] Konno, H. and Kuno, T. (1990), Generalized linear multiplicative and fractional programming, *Annals of Operations Research* 25, 147-162.
- [90] Konno, H. and Kuno, T. (1995), Multiplicative programming problems, in: Horst, R. and Pardalos, P.M., eds., *Handbook of Global Optimization, Nonconvex Optimization and its Applications*, Kluwer Academic Publishers, Dordrecht, The Netherlands.
- [91] Koontz, W.L., Narendra, P.M. and Fukunaga, K. (1975), A branch-and-bound clustering algorithm, *IEEE Transactions on Computing* 23, 908-914.
- [92] Krovi, R. (1992), Genetic algorithm for clustering: A preliminary investigation, *Proceedings of the 25th Hawaii International Conference on Systems Sciences*, 540-544.
- [93] Kuno, T. and Konno, H. (1992), A parametric successive underestimation method for convex multiplicative programs, *Journal of Global Optimization* 1, 267-285.
- [94] Kuno, T., Konno, H. and Yamamoto, Y. (1992), A parametric successive underestimation method for convex programming problems with an additional convex multiplicative constraint, *Journal of the Operations Research Society of Japan* 35, 290-299.
- [95] Kuno, T. Yajima, Y. and Konno, H. (1993), An outer approximation method for minimizing the product of several convex functions on a convex set, *Journal of Global Optimization* 3, 322-335.
- [96] Lasserre, J.B. (2001), Global optimization with polynomials and the problem of moments, *SIAM Journal of Optimization* 11(3), 796-817.
- [97] Lasserre, J.B. (2002), Semidefinite programming versus LP relaxations for polynomial programming, *Mathematics of Operations Research* 27(2), 347-360.
- [98] Laurent, M. and Rendl, F. (2002), Semidefinite programming and integer programming, Working paper, CWI, Amsterdam, The Netherlands.
- [99] Leung, Y., Zhang, J. and Xu, Z. (1997), Neural networks for convex hull computation, *IEEE Transactions on Neural Networks* 8(3), 601-611.
- [100] Li, H.L. and Chang, C.T. (1998), An approximate approach of global optimization for polynomial programming problems, *European Journal of Operational Research* 107, 625-632.
- [101] Lichtenberg, E. and Zilberman, D. (1988), Efficient regulation of environmental health risks, *The Quarterly Journal of Economics* 103(1), 167-178.

- [102] Linares, P. (2002), Multiple criteria decision making and risk analysis as risk management tools for power systems planning, *IEEE Transactions on Power Systems* 17(3), 895-900.
- [103] Lindo Systems Inc., (2005), New LINGO 9.0, www-document, <http://www.lindo.com>.
- [104] Liu, Y. and Guan, X. (2002), Optimization of purchase allocation in dual electric power markets with risk management, *Automation of Electric Power Systems* 26(9), 41-44 and 48.
- [105] Lukashin, A.V. and Fuchs, R. (2000), Analysis of temporal gene expression profiles: clustering by simulated annealing and determining the optimal number of clusters, *Bioinformatics* 17(5), 405-414.
- [106] Luss, H. (1999), On equitable resource allocation problems: A lexicographic minimax approach, *Operations Research* 47(3), 361-378.
- [107] Manber, U. (1989), *Introduction to algorithms: a creative approach*, Addison-Wesley Publishing Company, Reading, MA.
- [108] Mangiameli, P., Chen, K.S. and West, D. (1996), A comparison of SOM neural network and hierarchical clustering methods, *European Journal of Operations Research* 93, 402-417.
- [109] McCormick, G.P. (1976), Computability of global solutions to factorable nonconvex programs: Part I-convex underestimating problems, *Mathematical Programming* 10, 147-175.
- [110] McCormick, G.P. (1983), *Nonlinear Programming: Theory, Algorithms, and Applications*, John Wiley & Sons, Inc., New York, N.Y.
- [111] McQueen, J.B. (1967), Some methods of classification and analysis of multivariate observations, *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, Berkeley, CA, 281-297.
- [112] Mills, E. (2002), The insurance and risk management industries: New players in the energy-efficient and renewable energy products and services, Lawrence Berkeley National Laboratory, Energy Analysis Department, University of California, Berkeley, CA 94720, USA.
- [113] Mulvey, J.M. and Crowder, H.P. (1979), Cluster analysis: An application of lagrangian relaxation, *Management Science* 25(4), 329-340.
- [114] Morgan, M.G. (2000), Risk management should be about efficiency and equity, *Environmental Science and Technology* 34(1), 32A-34A.
- [115] Mosler, K. (1997), De minimis and equity in risk, *Theory and Decision* 42, 215-233.
- [116] Murthi, S. (2002), Preventive Risk Management for Software Projects, *IT Professional* 4(5), 9-10 and 12-15.
- [117] Myers, R.H. (1995), *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, John Wiley & Sons, Inc., New York, N.Y.
- [118] Neumaier, A., Shcherbina, O. and Huyer, W. (2004), A comparison of complete global optimization solvers, Working Paper, Institute for Mathematics, Nordbergstr, Wien, Austria.

- [119] Ohba, Y., Hayashi, T., Yoshida, Y. and Takahashi, K. (1984), Reliability analysis of ultra-high voltage DC transmission system by event tree analysis, *Electrical Engineering in Japan* 104(1), 118-128.
- [120] Paté-Cornell, M.E. and Fischbeck, P.S. (1994), Risk management for the tiles of a space shuttle, *Interfaces* 24(1), 64-86.
- [121] Patra, S., Soman, K. P. and Misra, R. B. (1995), Event tree analysis of a power system using Bayesian and fuzzy set approach, *Journal of the Institution of Engineers (India), Part Et: Electronics & Telecommunication Engineering Division* 76, 11-18.
- [122] Perriera, S. (2002), Risk management for the International Space Station, *European Space Agency (Special Publication)*, ESA SP 486, 339-344.
- [123] Pinter, J.D. (1996), *Global Optimization in Action*, Kluwer Academic Publishers, The Netherlands.
- [124] Polyak, B. T. and Kogan, J. (1995), Necessary and sufficient conditions for robust stability of linear systems with multiaffine uncertainty structure, *IEEE Transactions on Automatic Control* 40, 1255-1260.
- [125] Porter, M., and Savigny, K.W. (2002), Natural Hazard and Risk Management for South American Pipelines, *Proceedings of the International Pipeline Conference*, IPC (A), 861-869.
- [126] Powell, M.J.D. (2000), UOBYQA: Unconstrained optimization by quadratic approximation, Numerical Analysis Report DAMTP 2000/NA14, University of Cambridge.
- [127] Qiu, L. and Davison, E. J. (1989), A simple procedure for the exact stability robustness computation of polynomials with affine coefficient perturbations, *Systems and Control Letters* 13, 413-420.
- [128] Raman, R. (2004), Accounting for dynamic processes in process emergency response using event tree modeling, *Center for Chemical Process Safety, 19th Annual International Conference - Emergency Planning Preparedness, Prevention, and Response*, 197-213.
- [129] Rao, M.R. (1971), Cluster analysis and mathematical programming, *Journal of American Statistical Association* 66, 622-626.
- [130] Rasmussen, N.C. (1975), Reactor Safety Study: An assessment of accident risks in US commercial nuclear power plants, *Nuclear Regulatory Commission Report*.
- [131] Rauzy, A. (1993), New algorithms for fault tree analysis, *Reliability Engineering and System Safety* 40, 203-211.
- [132] Rauzy, A. (1996), A Brief Introduction to Binary Decision Diagrams, *European Journal of Automation* 30(8), 1033-1051.
- [133] Renson, L. (2002), Risk management of future reusable launcher mission using active health monitoring systems (HMS), *European Space Agency (Special Publication)*, ESA SP 486, 247-253.
- [134] Rivard, J.B. (1971), Risk minimization by optimum allocation of resources available for risk reduction, *Nuclear Safety* 12(4), 305-309.

- [135] Rote, G. (1992), The convergence rate of the sandwich algorithm for approximating convex functions, *Computing* 48, 337-361.
- [136] Roubens, M. (1982), Fuzzy clustering algorithms and their cluster validity, *European Journal of Operational Research* 10(3), 294-301.
- [137] Royden, H. (2001), *Real Analysis*, McGraw Hill, New York, N.Y.
- [138] Ruspini, E.H. (1973), New experimental results in fuzzy clustering, *Information Science* 6, 273-284.
- [139] Ryoo, H.S. and Sahinidis, N.V. (2001), A branch-and-reduce approach to global optimization, *Journal of Global Optimization* 8, 107-139.
- [140] Ryoo, H.S. and Sahinidis, N.V. (2001), Analysis of bounds for multilinear functions, *Journal of Global Optimization* 19, 403-424.
- [141] Ryoo, H.S. and Sahinidis, N.V. (2003), Global optimization of multiplicative programs, *Journal of Global Optimization* 26, 387-418.
- [142] Sahinidis, N.V. (1996), BARON: A general purpose global optimization software package, *Journal of Global Optimization* 8(2), 201-205.
- [143] Sahinidis, N.V. and Tawarmalani, M. (2002a), *GAMS/BARON 5.0: Global optimization of mixed-integer nonlinear programs*, GAMS Users Guide.
- [144] Sahinidis, N.V. and Tawarmalani, M. (2002b), *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming*, Kluwer Academic Publishers, Dordrecht, The Netherlands.
- [145] Sahinidis, N.V. and Tawarmalani, M. (2003), Accelerating branch-and-bound through a modeling language construct for relaxation-specific constraints, Working paper, Department of Chemical and Biomolecular Engineering, University of Illinois, Urbana Champaign, Illinois.
- [146] Schichl, H. (2003), Mathematical modeling and global optimization, Habilitation Thesis, Cambridge University Press, to appear.
- [147] Selim, S.Z. (1982), A global algorithm for the clustering problem, *Presentation at the ORSA/TIMS Joint Meeting*, San Diego, CA.
- [148] Selim, S.Z. and Al-Sultan, K.S. (1991), A simulated annealing algorithm for the hard clustering problem, *Pattern Recognition* 24, 1003-1008.
- [149] Shcherbina, O., Neumaier, A., Sam-Haroud, D., Vu, X-H. and Nguyen, T-V. (2004), Benchmarking global optimization and constraint satisfaction nodes, Proceedings of COCOS'02, Springer-Verlag, Berlin, to appear.
- [150] Shectman, J.P. and Sahinidis, N.V. (1996), A finite algorithm for global minimization of separable concave programs, in: Floudas, C.A. and Pardalos, P.M., eds., *State of the Art in Global Optimization*, Computational Methods and Applications, Kluwer Academic Publishers, Dordrecht, The Netherlands.
- [151] Sherali, H.D. (1998), Global optimization of nonconvex polynomial programming problems having rational exponents, *Journal of Global Optimization* 12, 267-283.

- [152] Sherali, H.D. and Adams, W.P. (1990), A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems, *SIAM Journal on Discrete Mathematics* 3(3), 411-430.
- [153] Sherali, H.D. and Adams, W.P. (1994), A hierarchy of relaxations and convex hull characterizations for mixed-integer zero-one programming problems, *Discrete Applied Mathematics* 52, 83-106.
- [154] Sherali, H.D. and Adams, W.P. (1999), Reformulation-linearization techniques for discrete optimization problems, In: Du, D.-Z. and Pardalos, P.M., eds., *Handbook of Combinatorial Optimization I*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 479-532.
- [155] Sherali, H.D., Alameddine, A. and Glickman, T.S. (1995), Biconvex models and algorithms for risk management problems, *American Journal of Mathematical and Management Sciences* 3-4, 197-228.
- [156] Sherali, H.D., Brizendine, L.D., Glickman, T.S. and Subramanian, S. (1997), Low probability-high consequence considerations in routing hazardous material shipments, *Transportation Science* 31(3), 237-251.
- [157] Sherali, H.D. and Fraticelli, B.M.P. (2002), Enhancing RLT relaxations via a new class of semidefinite cuts, *Journal of Global Optimization* 22, 233-261.
- [158] Sherali, H.D. and Ganesan, V. (2003), A pseudo-global optimization approach with application to the design of containerships, *Journal of Global Optimization* 26, 335-360.
- [159] Sherali, H.D. and Smith, J.C. (2001), Improving discrete model representations via symmetry considerations, *Management Science* 47(10), 1396-1407.
- [160] Sherali, H.D., Smith, J.C., and Trani, A.A. (2002), An Airspace Planning Model for Selecting Flight Plans under Workload, Safety, and Equity Considerations, *Transportation Science* 36, 378-397.
- [161] Sherali, H.D., Staats, R.W. and Trani, A.A. (2003c), An airspace planning and collaborative decision-making model: Part I-Probabilistic conflicts, workload, and equity considerations, *Transportation Science*, 37(4), 434-456.
- [162] Sherali, H.D. and Subramanian, S. (1999), Opportunity cost-based models for traffic incident response problems, *Journal of Transportation Engineering* 125(3), 176-185.
- [163] Sherali, H.D. and Tuncbilek, C.H. (1992), A global optimization algorithm for polynomial programming problems using a Reformulation-Linearization Technique, *Journal of Global Optimization* 2, 101-112.
- [164] Sherali, H.D. and Tuncbilek, C.H. (1995), A reformulation-convexification approach for solving nonconvex quadratic programming problems, *Journal of Global Optimization* 7, 1-31.
- [165] Sherali, H.D. and Tuncbilek, C.H. (1997), Comparison of two Reformulation-Linearization Technique based linear programming relaxations for polynomial programming problems, *Journal of Global Optimization* 10, 381-390.

- [166] Sherali, H.D. and Wang, H. (2001), Global optimization of nonconvex factorable programming problems, *Mathematical Programming* 89(3), 459-478.
- [167] Shor, N.Z. (1990), Dual quadratic estimates in polynomial and boolean programming, *Annals of Operations Research* 25, 163-168.
- [168] Shor, N.Z. (1998), *Nondifferentiable Optimization and Polynomial Problems*, Kluwer Academic Publishers, Dordrecht, The Netherlands.
- [169] Sideris, S. and Sánchez Peña, R. S. (1989), Fast computation of multivariable stability margin for real interrelated uncertain parameters, *IEEE Transactions on Automatic Control* 34, 1272-1276.
- [170] Sinnamon, R.M. and Andrews, J.D. (1996), Quantitative fault tree analysis using binary decision diagrams, *European Journal of Automation* 30(8), 1052-1073.
- [171] Sinnamon, R.M. and Andrews, J.D. (1997a), Improved accuracy in quantitative fault tree analysis, *Quality and Reliability Engineering International* 13, 285-292.
- [172] Sinnamon, R.M. and Andrews, J.D. (1997b), Improved efficiency in qualitative fault tree analysis, *Quality and Reliability Engineering International* 13, 293-298.
- [173] Sivakumar, R.A., Batta, R. and Karwan, M.H. (1993), A network based model for transporting extremely hazardous materials, *Operations Research Letters* 13, 85-93.
- [174] Späth, H. (1980), *Cluster Analysis Algorithms for Data Reduction and Classification of Objects*, John Wiley and Sons, New York, N.Y.
- [175] Starr, C. and Whipple, C. (1982), Risk of risk decisions, *Risk in the Technological Society. AAAS Selected Symposia Series*, Hohenemser, C. and Kasperson, J. X. eds., American Association for Advancement of Science, Westview Press, Inc., Boulder, CO.
- [176] Sultan, M., Wigle, D.A., Cumbaa, C.A., Maziarz, M., Glasgow, J., Tsao, M.S. and Jurisica, I. (2002), Binary tree-structured vector quantization approach to clustering and visualizing microarray data, *Bioinformatics* 18(1), 111-119.
- [177] Takaragi, K., Sasaki, R. and Shingai, S. (1983), Algorithm for obtaining simplified prime implicant sets in fault tree and event tree analysis, *IEEE Transactions on Reliability* 4, 386-390.
- [178] Tawarmalani, M. and Sahinidis, N.V. (1999), BARON on the web, <http://archimedes.scs.uiuc.edu/baron/baron.html>.
- [179] Tawarmalani, M. and Sahinidis, N. V. (2002a), Convexification and global optimization in continuous and mixed-integer nonlinear programming: Theory, algorithms, software, and applications, *Nonconvex Optimization and its Applications* 65, Ch. 9, Kluwer Academic Publishers, Dordrecht, The Netherlands.
- [180] Tawarmalani, M. and Sahinidis, N. V. (2002b), Convexification and global optimization of the pooling problem, Manuscript, Department of Chemical and Biomolecular Engineering, University of Illinois, Urbana Champaign, Urbana Champaign, Illinois.
- [181] Teboulle, M. and Kogan, J. (1994), Applications of optimization methods to robust stability of linear systems, *Journal of Optimization Theory and Applications* 81, 169-192.

- [182] Tesi, A. and Vicino, A. (1990), Robustness analysis for linear dynamical systems with linearly correlated parametric uncertainties, *IEEE Transactions on Automatic Control* 35, 186-191.
- [183] Turner, J.V. (2002), Risk management of the space shuttles upgrades development program, *European Space Agency (Special Publication)*, ESA SP 486, 345-356.
- [184] Unwin, S. D. (1984), Binary event string analysis: A compact numerical representation of the event tree, *Risk Analysis* 4(2), 83-87.
- [185] Vandenberghe, L. and Boyd, S. (1996), Semidefinite programming, *SIAM Review* 38(1), 49-95.
- [186] Vanderbei, R.J. and Benson, H.Y. (2000), On formulating semidefinite programming problems as smooth convex nonlinear optimization problems, Working paper, Department of Operations Research and Financial Engineering, Princeton University, Princeton, N.J.
- [187] Vinod, H.D. (1969), Integer programming and the theory of grouping, *Journal of American Statistical Society* 64, 506-519.
- [188] Volkov, E.A. (1990), *Numerical Methods*, Hemisphere Publishing, New York, N.Y.
- [189] Ward, J.H. Jr. (1963), Hierarchical grouping to optimize an objective function, *Journal of American Statistical Society* 58, 236-244.
- [190] Weinstein, M.C. (1979), Decision making for toxic substance control, *Public Policy* 27, 333-338.
- [191] Windham, M.P. (1982), Cluster validity for the fuzzy c-means clustering algorithm, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 4, 357-363.
- [192] Windham, M.P. (1983), Geometric fuzzy clustering algorithms, *Fuzzy Sets and Systems* 10, 271-279.
- [193] Yang, Y. and Qiu, L. (1993), Event tree analysis for the system of hybrid reactor, *Nuclear Power Engineering* 14(6), 516-522.
- [194] Young, H.P. (1994), *Equity in Theory and Practice*, Princeton University Press, Princeton, NJ.
- [195] Zadeh, L.A. and Desoer, C. A. (1963), *Linear Systems Theory*, McGraw-Hill, New York, N.Y.
- [196] Zahid, N., Limouri, M. and Essaid, A. (1999), New cluster-validity for fuzzy clustering, *Pattern Recognition* 32(7), 1089-1097.
- [197] Zhang, X. and Yan, S. (1999), Event-tree analysis of steam generator tube ruptures, *Nuclear Power Engineering* 20(2), 169-173.
- [198] Zhang, Z., Wu, C., Xia, T., Zhang, B. and Li, A. (2004), Chemical hazards assessing and accidents estimation by event tree modeling, *Proceedings of the 2004 International Symposium on Safety Science and Technology* 4 - Part B, 1753-1758.