UNIVERSITY OF SOUTHAMPTON

Faculty of Social Sciences
School of Mathematical Sciences

Models and Methods for Integer Programming under Structure and Uncertainty

DOI: 10.5258/SOTON/PG/T118

by

Montree Jaidee
MSc
ORCiD: 0009-0009-4298-8356

A thesis for the degree of
Doctor of Philosophy

May 2026

University of Southampton

Abstract

Faculty of Social Sciences
School of Mathematical Sciences

Doctor of Philosophy

Models and Methods for Integer Programming under Structure and Uncertainty

by Montree Jaidee

Integer Programming (IP) is a fundamental tool of modern decision-making; however, these problems are highly combinatorial and often require exponential time to solve. Such computational difficulty can also be enhanced by the specific algebraic structures of the solution space. Each combinatorial model has a unique structure, enabling the development of different methods to use these particular aspects. This thesis investigates two independent but complementary frontiers: exploiting mathematical structure in deterministic optimization and handling uncertainty in stochastic settings.

In the deterministic domain, we introduce prime programming, a subclass of integer programming whose decision variables are confined to prime numbers. Utilizing this algebraic structure, we develop a specific branch-and-bound method that employs number-theoretic branching rules designed for prime solutions. Next, we consider a quadratically constrained quadratic programming model characterized by nonconvex Euclidean norm constraints. Leveraging the geometric structure of these constraints, we derive a spatial branch-and-bound approach to solve the model more efficiently. This method is based on the intuition of replacing the Euclidean ball with a polyhedron that is iteratively refined to approximate the ball more closely. This approach provides a global optimality guarantee for k-hyperplane clustering problems where traditional heuristic methods frequently fail. We further provide a practical application of mixed-integer quadratic programming to rationalize the Household Waste Recycling Centre (HWRC) network in Hampshire, balancing fiscal efficiency with equitable public access.

In the stochastic domain, this thesis opposes traditional risk-averse paradigms by developing risk-seeking optimization models with probabilistic constraints. In this framework, constraint violations are viewed as lucky scenarios, allowing for a higher number of violations rather than requiring the constraints to be satisfied at a

near-certain level. We treat these probabilistic constraints as a union of events and provide an alternative solution methodology; specifically, approximation models based on integer programming that provide valid lower and upper bounds. Finally, a chance-constrained study of the European energy market is provided to quantify the economic value of wind-solar complementarity.

This work collectively illustrates that specialized algorithms for specific frameworks (algebraic, geometric, or probabilistic-constrained) enhance the solvability of complicated optimization problems in mathematics, machine learning, and public policy.

Contents

List of Figures

List of Tables

List of Abbreviations

***k*-HC** *k*-hyperplane clustering

ARMA autoregressive moving average

B&B branch and bound

GBD generalized Benders decomposition

HCC Hampshire County Council

HWRC Household Waste Recycling Centre

IP integer programming

IPA-SBB iterative polyhedron approximation with spatial branch and bound

LP linear programming

MINLP mixed integer nonlinear programming

MIP mixed-integer programming

MIQCQP mixed-integer quadratically constrained quadratic programming

MIQP mixed-integer quadratic programming

NLP nonlinear programming

OA outer approximation

PP prime program

PV photovoltaic

QCQP quadratically constrained quadratic programming

QP quadratic programming

SAA sample average approximation

sB&B spatial branch and bound

TSP Traveling Salesman Problem

WRAP Waste and Resources Action Programme

Declaration of Authorship

I declare that this thesis and the work presented in it is my own and has been generated by me as the result of my own original research.

I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others, this is always clearly attributed;
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
7. None of this work has been published before submission

Signed:..... Date:.....

Acknowledgements

Completing this PhD has been a journey of both academic and personal growth, made possible by several key people.

I would like to start by thanking Professor Houduo Qi. He was the first person I contacted at the University of Southampton, and I remain grateful to him for accepting me as a student and starting me on this path.

I owe a debt of gratitude to Dr. Stefano Coniglio for leading me into the fascinating world of integer programs. Even after navigating the difficulties of supervisor transitions, Stefano stayed by my side as a supervisor, even as I pivoted to a new project. I truly appreciate his dedication.

To Dr. Bismark Singh, my main supervisor for the past three years: thank you for everything. Working with you has taught me how to think and work more effectively and systematically—lessons I will carry with me throughout my career.

To my friends who endured the unlovely British weather with me: thank you for making life here so enjoyable. A special shout-out to my friends in Operation Research group for all the fun activities that provided a much-needed break from the study.

To my Thai friends in the UK: thank you for making me feel at home even when I was thousands of miles away. Because of you, I rarely felt alone, and I leave this PhD not only with a degree but with much-improved cooking skills and many memories of great Thai food.

To the Development and Promotion of Science and Technology Talents Project (DPST) and the Royal Thai Embassy: thank you for the generous financial support of my tuition and living expenses throughout my studies.

To my parents: thank you for your endless encouragement. Even though the complexities of a PhD were often unfamiliar to you, your unwavering belief in me provided the strength I needed to keep going.

Most importantly, I thank Grace, my partner. Your support in every aspect of my life has been my foundation. Thank you for being there for me.

Chapter 1

Introduction

Integer programming (IP) is a core branch of mathematical optimization related to modeling decision-making problems involving discrete and continuous variables. The problem setup can capture the need for constraints where an object (or a yes/no decision) cannot be split into a continuous number, allowing for the formulation of important problems such as the traveling salesman problem (Dantzig et al., 1954), the assignment problem (Kuhn, 1955), the scheduling problem (Pritsker et al., 1969), and the production planning problem (Hendry and Kingsman, 1989). IP has become useful in many areas such as operations research, energy systems, logistics and supply chain, finance, and data analytics. Although solvers for these problems are becoming more developed and advanced, these models are still hard to solve due to their $\mathcal{NP}$-hard nature (Karp, 1972), which involves non-convexity and highly combinatorial structures. Dealing with these problems still requires specialized methods such as efficient model formulation and algorithmic development.

This thesis investigates how exploiting problem structure can render intractable IPs tractable, both in deterministic and stochastic settings. It develops the theory and application of IP through the advancement of models and methods that serve under structured settings and uncertainty. The thesis expands deterministic global optimization via new branch-and-bound techniques, formulates stochastic models incorporating probabilistic constraints and risk preferences, and illustrates the practical applications of these methodologies in machine learning and public-sector planning. In a deterministic context, branch-and-bound (B&B) offers a structured approach for investigating the feasible region by dividing it into subproblems and employing relaxations to derive valid bounds. Subproblems that cannot improve the incumbent solution are pruned, ensuring global optimality through the solving of all regions. The present work expands B&B in two areas: the design of specialized branching rules for the Prime Programming Problems and the development of an iterative polyhedral approximation with spatial branch-and-bound for addressing Euclidean norm constrained models, which involve hyperplane clustering and linear

classification problems in machine learning. A practical application is demonstrated using a quadratic mixed-integer programming model for decision-making of Household Waste Recycling Centers' closures in Hampshire, highlighting the function in supporting data-driven decisions concerning public infrastructure. The second research theme investigates stochastic programming, especially in chance-constrained optimization, which explicitly models uncertainty by necessitating that constraints be satisfied with high probability. Joint chance constraints provide considerable difficulties due to their nonconvex feasible regions, therefore requiring the use of approximations. This thesis introduces new formulations employing probabilistic inequalities, enabling the flexible modeling of risk-averse, risk-neutral, and risk-seeking behaviors. These methods are applied to address problems including the assessment of collaboration between wind and solar energy producers and the modeling of human risk-seeking behaviors.

Deterministic IP models struggle with the combinatorial explosion induced by complex structural constraints, such as feasibility constraints or geometric clustering with unit ℓ_2-norm hyperplanes, while stochastic models struggle with the intractability of integrating over high-dimensional uncertainty spaces. In the stochastic setting, uncertainty is represented through possible discrete scenarios, which allows problems to be formulated based on these possible realizations as IPs. Therefore, both types of models advance solution methods for IPs. These methods either solve to global optimality with the branch-and-bound technique or provide an approximation using bounding techniques such as probabilistic inequalities and sample-based approximations. Together, they form a coherent framework for addressing optimization problems that involve discrete structure and uncertainty.

Finally, the problems examined in this thesis address notable necessities in practice. Structured IPs models are relevant in number theory, combinatorial design, and machine learning. In contrast, chance-constrained and risk-seeking models gain significance in energy planning, finance, and safety-critical decision-making. This thesis combines mathematical development with application-based modeling, offering theoretical insights and practical methods for optimizing complex systems subject to structural constraints and uncertainty.

1.1 Overview of Structured and Stochastic Integer Programming

IP has been under development for over 70 years. Prior research in the 1950s and 1960s was driven by the need to address scheduling, logistics, and resource allocation problems. Dantzig et al. (1954), Kuhn (1955), and Pritsker et al. (1969) developed the first modeling systems for solving such discrete optimization problems, specifically

for the traveling salesman problem (TSP), the assignment problem, and the scheduling problem, respectively. The first general-purpose framework for computing global optima in IPs, called the branch-and-bound (B&B) algorithm, was developed by Land and Doig (1960). This framework has been expanded over time with the introduction of cutting-plane techniques like Gomory's mixed-integer cuts (Gomory, 1958), Chvátal cuts (Chvátal, 1973), disjunctive cuts (Balas, 1979), and lift-and-project cuts (Balas et al., 1996), as well as ideas to strengthen the formulation by considering polyhedra of the feasible region. The book edited by Jünger et al. (2009) provides a thorough history of the evolution of techniques for solving IPs.

The B&B algorithm remains an active research area due to the inherent complexity of specific combinatorial problems. Since these problems exhibit varied geometric or algebraic structures, different valid inequalities can be derived. Classically, the problem setup only considered solving models with binary (0-1) decision variables or general integer variables by branching on a variable, say x_j, whose relaxation solution x_j^* is non-integer. This process creates two new children nodes using the branching inequalities: $x_j \leq \lfloor x_j^* \rfloor$ or $x_j \geq \lfloor x_j^* \rfloor + 1$. This idea is generalized by the concept of prime programming, one of the problems proposed in this thesis, which overcomes the restriction on the creation of prior branching nodes. In particular, the branching method provided in the thesis can be used to effectively formulate and solve the optimization model if the variable is restricted to being a prime number or a number with another special structure. This development extends traditional number theory problems, allowing them to be solved using optimization techniques, which introduces new challenges for the optimization community.

Not only can the B&B algorithm solve mixed-integer programming problems, but the spatial branch-and-bound (sB&B) technique is also a similar divide-and-conquer technique to find a deterministic globally optimal solution in nonconvex programs, especially for mixed-integer nonlinear programs (MINLPs) with nonconvex terms. The method includes replacing the nonconvex terms with their convex envelopes as a relaxation. After the relaxation problem is solved, either an integer or continuous variable is chosen as the branching variable. The convex envelopes are further tightened from this branching, and similar pruning is applied when a certain subregion cannot provide a better solution than the incumbent.

This thesis considers a specific problem in machine learning: k-hyperplane clustering, which contains nonconvex unit ℓ_2-norm constraints. This problem has a specific property where creating convex envelopes over the feasible region would result in the entire space; traditional relaxation techniques are not efficient to deal with this problem. The IPA-SBB method developed in this thesis addresses this difficulty by constructing a sequence of increasingly accurate polyhedral approximations, each solved within a sB&B framework. The idea of disjunctive programming is also applied to the algorithm by converting the formulation into multi-way branching.

This extends the traditional B&B framework, where only two children nodes are typically created at any point. This hybrid methodology preserves the global optimality guarantees of B&B while managing continuous nonconvexity and enabling applications such as hyperplane clustering and linear classification. Furthermore, an application using structural quadratic mixed-integer programs is examined in the public sector to provide analysis and policy recommendations for Hampshire County regarding the optimal recycling center closures that minimize their impact.

The second part of the thesis focuses on probabilistic constraints and studies in the field of stochastic optimization. In this setting, decision-making is required to remain feasible with a high degree of probability. Introduced by Charnes and Cooper (1959), chance-constrained optimization offers a framework for managing uncertainty without making too conservative decisions. This method is especially helpful in application areas like autonomous vehicles (Blackmore et al., 2011), portfolio optimization (Rockafellar and Uryasev, 2000), and energy markets (Cai et al., 2009). However, directly solving these chance-constrained models typically requires several multi-dimensional integrations of known probability distributions, which are almost impossible in practice. Luedtke et al. (2008) show that the problem is $\mathcal{NP}$-hard. This thesis extends the probabilistic constrained model to reflect risk-seeking human behavior, leading to the search for suboptimal decision-making supported by prospect theory. Approximation models based on the classical probability of the union of sets are developed to obtain valid lower and upper bounds for the problem. Furthermore, a chance-constrained model application in the energy market is examined to provide analysis and policy recommendations for coordinating between wind and solar farm owners.

1.2 Research Motivation and Contributions

The overall aim of this thesis is to develop models and methodology that advance the theory and applications of IP in both deterministic and stochastic settings. This involves the development of global optimization methods to solve models within specific structures and the design of approximation schemes for stochastic problems. The research extends the B&B algorithm using customized branching techniques and expands chance-constrained models to a more general framework that accommodates risk-seeking behavior, whereas most of the problems in chance-constrained programming focus either on a risk-averse or risk-neutral setting. Bounding techniques are used to approximate the objective model with classical probabilistic bounds. Applications for both the deterministic and stochastic sides are also considered by providing policy recommendations from the data-driven decision-making.

The thesis focuses on the following objectives:

1. **Structural modeling in integer programming:** To examine IP models including the prime programming problem, unit ℓ_2-norm constrained k-hyperplane clustering, quadratic facility-location models for recycling center optimization, and stochastic models within IPs frameworks and to understand how these structures can be exploited algorithmically.
2. **Developments in branch-and-bound:** To develop specialized branching schemes and B&B methods for prime programs and iterative polyhedral approximations with spatial branching mechanisms that exploit these structures, thereby improving global optimization performance and extending traditional B&B and sB&B frameworks.
3. **Chance-constrained modeling under uncertainty:** To provide alternative approximation schemes for decision-making under uncertainty using probabilistic inequalities of union sets and to extend chance-constrained models to a new framework that captures risk-seeking behavior in addition to risk-neutral and risk-averse settings.
4. **Policy recommendations:** To present policy recommendations derived from data-driven analyses of optimization models regarding the proposed closures of recycling sites in Hampshire and the cooperation of wind and solar farm operators throughout 28 European nations.

1.3 Thesis Structure

The remainder of this thesis is structured as follows.

Chapter 2 presents literature related to the methodology used in this thesis. This includes the B&B algorithm for IP, along with the sB&B. Following this, the chance-constrained model is introduced with its sample-based approximation framework, which corresponds to an IP equivalent formulation.

Chapter 3 presents a new scheme of solving number-theoretical problems in an optimization model framework where the variables are constrained by being prime numbers, which is the core study in number theory. Then, the B&B algorithm is adapted to solve the formulated prime programming problem to effectively search for solutions. A heuristic method based on the structure of prime numbers is provided to improve the branching process further. This includes the computational results to demonstrate the effectiveness of the algorithm.

Chapter 4 focuses on nonconvex optimization models due to the unit ℓ_2-norm constraint, which is usually presented in machine learning models. This work studies the k-hyperplane clustering problem, where points are divided into groups with hyperplanes in the form of a quadratic mixed-integer program. We proposed an iterative polyhedron approximation method that extends the sB&B framework to enable multi-way branching of disjunctive programs by using a polyhedron to approximate the geometry of a ball shape within a unit ℓ_2-norm constraint. Then, it iteratively improves its geometry closer to the ball shape. This method can guarantee global optimality when solving the problem. Furthermore, the linear classification problem is also examined to demonstrate its applicability in general.

Chapter 5 provides a data-driven decision-making model applied to the proposal of Hampshire's council to close some of the Household Waste Recycling Centre (HWRC) recycling centers. The model provides an alternative way of facilities' closures that minimizes the impact of compromising with people in their area while considering the fairness of facility utilization.

Chapter 6 proposes a framework to reflect human risk-seeking behavior. This setup is formulated using probabilistic constraints, which allow for a high fraction of violations. We provide approximation bounds of such models by treating the probabilistic constraints as a union of sets and applying inequalities from probability theory to determine these bounds. Possible scenarios are generated, and we introduce integer variables to capture the number of time periods that are failed for each scenario. The inequalities are reformulated into optimization models, which could be solved to obtain these bounds.

Chapter 7 studies an application in the energy system of the chance-constrained optimization model. Due to the inherent uncertainty in both wind and solar power sources, the day-ahead bidding model permits farm owners to miss their energy commitments for a small fraction of the time. The data for 28 European countries for hourly wind and solar power generation is available, and then forecast scenarios are generated from statistical models. Since there is some complementarity in wind and solar power generation, we examine whether cooperative bidding between wind and solar farm owners would be more profitable or not under two circumstances: without and with battery storage. Policy recommendations are provided based on the results of solving IPs derived from the chance-constrained models.

Finally, Chapter 8 concludes this work by summarizing the main findings and discussing the overall contributions.

Note: The recycling public policy optimization project in Chapter 5 is a joint work with Zheng Sun, Sher Singh, and Thi Nguyen, supervised by Dr. Bismark Singh; here, my contributions are mainly in the analysis of results, visualizations, and preparing the manuscript (currently

under peer-review). The research in Chapters 3, 6, and 7 is supervised by Dr. Bismark Singh, while the research in Chapter 4 is supervised by Dr. Stefano Coniglio.

1.4 Publications

This thesis is based on the following under-review and in-preparation research papers:

- Jaidee, M. and Singh, B. (2025). *The Prime Programming Problem: Formulations and Solution Methods*. Manuscript under review.
- Jaidee, M. and Coniglio, S. (2025) *Iterative Polyhedron Approximation to k-Hyperplane Clustering Problem*. Manuscript in preparation.
- Jaidee, M., Sun Z., Singh S., Nguyen T., and Singh, B. (2025) *Optimizing Household Waste Recycling Centre Network Rationalization in Hampshire*. Manuscript under review.
- Jaidee, M. and Singh, B. (2025). *Risk-Seeking Problem Formulations and Their Approximations with Union Bounds*. Manuscript in preparation.
- Jaidee, M. and Singh, B. (2025). *Measuring the Economic Value of Wind–Solar Complementarity in Europe Using Chance Constraints*. Manuscript under review.

Chapter 2

Literature Review

This chapter reviews the theoretical and methodological foundations related to the two primary themes of this thesis: (i) exact optimization using B&B techniques for structured mixed-integer and nonconvex models, and (ii) chance-constrained optimization formulated within an IP framework. These subjects collectively establish the basis for the deterministic and stochastic models formulated in the following chapters. Section 2.1 focuses on deterministic global optimization methods that underpin the structural models studied in Chapters 3–6, while Section 2.2 covers chance-constrained and stochastic programming techniques that support the models developed in Chapters 6 and 7.

2.1 Exact Deterministic Methods

A broad class of optimization problems studied in this thesis can be formulated as mixed-integer nonlinear programs, for which deterministic methods are required to guarantee global optimality. This section reviews the principal families of exact deterministic algorithms used to solve such problems, including classical B&B, sB&B for nonconvex problems, decomposition-based approaches, quadratic mixed-integer models, and bilevel linear programs relevant to later chapters.

2.1.1 Branch-and-Bound for Integer and Mixed-Integer Programming

The B&B algorithm remains one of the most important frameworks for solving mixed-integer linear and nonlinear programs to global optimality. The main idea behind B&B is to divide the feasible region into subregions (called “branching”) and then solve the relaxation problems that give a dual bound on the objective function in each subregion (called “bounding”). If a subregion cannot obtain a better solution

than the best one already found, it is pruned ("pruning"). The process goes on and on until all of the subregions are either solved or pruned, at which time the global optimum is guaranteed.

Many approaches have been implemented over the years to speed up the B&B algorithm. Some of the most important ones are strong and pseudo-cost branching techniques (Achterberg et al., 2005), improved heuristics to select nodes, and the use of cutting planes that make relaxations tighter (Bixby, 2012). CPLEX, Gurobi, and Xpress are some of the most advanced commercial solvers that use these improvements. They are capable of solving problems containing millions of variables and constraints. Nonetheless, B&B persists in exhibiting an exponential increase in tree size, particularly for models that have weak relaxations, nonconvexities, or special algebraic structures.

Cutting plane methods have proven successful in solving MINLP problems. The idea is to solve the convex relaxation to optimality and then find a cutting plane (an inequality) that satisfies all feasible integer solutions but violates the current infeasible point. This derived inequality is then added to the problem's formulation, and the relaxation is re-optimized to find a new point. The idea of B&B together with cutting plane methods led to a framework called the branch-and-cut algorithm. The study of the polyhedral structure of the feasible region is important for deriving such valid inequalities.

Let $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ be a polyhedron. A face of P is a set of the form

$$F := P \cap \{x \in \mathbb{R}^n : cx = \delta\},$$

where $cx \leq \delta$ is a valid inequality for P. A face of P is a proper face if it is nonempty and properly contained in P. Maximal proper faces are called facets. A key characteristic of a facet is that it has a dimension of exactly $\dim(P) - 1$ (see Conforti et al. (2014) for more details).

There are two ways to describe a polyhedron P: as the intersection of halfspaces (H-representation) or as the convex hull of its vertices and rays (V-representation). The previous definition of P is in the H-representation, as it is described as the set of all points satisfying a finite number of linear inequalities, where each row of $Ax \leq b$ defines a halfspace. By the Minkowski-Weyl theorem (Charnes and Cooper, 1958), any polyhedron can be represented both as the intersection of finitely many halfspaces and as the convex hull of finitely many points plus the conic hull of finitely many rays. Thus, the polyhedron P can also be represented in the V-representation as:
$P = \text{conv}(V) + \text{cone}(R)$ where $V = \{v_1, \ldots, v_k\}$ is the set of vertices and
$R = \{r_1, \ldots, r_l\}$ is the set of extreme rays.

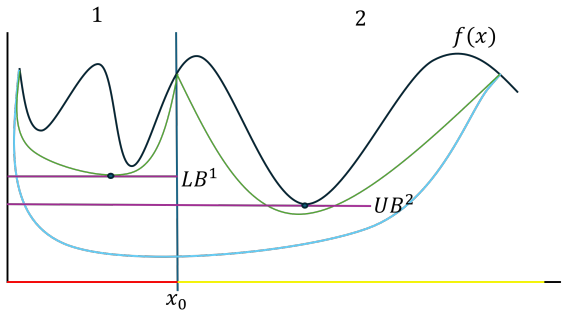


FIGURE 2.1: An example of a spatial branch-and-bound procedure. First, the blue curve is a convex relaxation for the whole feasible region of $f(x)$; the feasible region is branched at the point x_0 to split into subregions 1 and 2. The convex underestimator of each region is obtained in green. The lower bound in subregion 1 is larger than the upper bound of subregion 2; therefore, region 1 is pruned in minimization models.

The faces and facets of a polyhedron, along with its representations, provide important background for the method in Chapter 4. In this method, the branching technique is based on the facets of a polyhedron, and valid inequalities are derived from its V-representation. At the same time, classical B&B implementations cannot exploit specialized arithmetic structure in the domain of integer variables (such as prime constraints), which motivates the prime-constrained branching strategies developed in Chapter 3.

2.1.2 Spatial Branch-and-Bound for Nonconvex MINLP

The standard B&B framework depends on convex relaxations that are both tight and easy to compute. Nonetheless, attaining convex relaxations for nonlinear and nonconvex optimization problems, especially MINLPs, is challenging. The spatial branch-and-bound (sB&B) framework was created to solve this problem (Ryoo and Sahinidis, 1996; Sahinidis, 2003; Belotti et al., 2010). At now, sB&B is the primary framework for deterministic global optimization.

In sB&B, the feasible set is considered as a continuous region that contain multiple local optima. The algorithm divides this region into smaller subregions and creates convex underestimators for each one. For bilinear or multilinear terms, McCormick envelopes (McCormick, 1976) give lower bounds, and for feasible solutions found through local search or heuristics, they give upper bounds. The procedure continues until the global optimality gap attains a specified tolerance. Figure 2.1 illustrates the central idea of sB&B: convex relaxations are constructed for each subregion, enabling pruning when the relaxation proves that a subregion cannot yield an improved solution.

Practical performance of sB&B depends critically on the tightness of convex relaxations and the branching strategy used to partition the domain. A variety of

algorithmic improvements have been introduced, including branch-and-reduce (Sahinidis, 1996), domain reduction techniques (Pozo et al., 2010; Smith and Pantelides, 1999), probing heuristics (Savelsbergh, 1994), feasibility and optimality-based bound tightening (Quesada and Grossmann, 1993), and feasibility-pump heuristics. Comprehensive surveys of these techniques and their implementation in global solvers can be found in Belotti et al. (2013).

In Chapter 4, nonconvex geometric restrictions, such as unit ℓ_2-norm constraints, render classical convex relaxations ineffective: relaxing a unit ℓ_2-norm constraint leads to trivial or very weak bounds. The iterative polyhedral approximation (IPA-SBB) framework introduces adaptive polyhedral approximations that capture the geometry of nonconvex sets, building on the notions of sB&B but specifically tailored to unit ℓ_2-norm constrained clustering and classification models.

2.1.3 Other Exact Deterministic Frameworks

In addition to B&B and sB&B, decomposition-based exact methods have been developed for convex MINLPs. Outer Approximation (OA) (Duran and Grossmann, 1986) solves a mixed-integer linear master problem and continuous nonlinear subproblems by leveraging first-order information to generate cutting planes that approximate the nonlinear constraints. Generalized Benders Decomposition (GBD) (Benders, 1962; Geoffrion, 1972) uses dual information from nonlinear programming subproblems to generate Benders cuts that guide the discrete master problem.

These algorithms are highly effective for convex MINLPs with suitable structure, but they rely on convexity and strong duality—assumptions violated in the nonconvex geometric models considered in this thesis. As such, OA and GBD are not directly applicable to prime programming, unit ℓ_2-norm constrained clustering, or quadratic fairness models. Nonetheless, they form an important part of the broader landscape and highlight how exploiting problem structure via decomposition can improve solvability.

2.1.4 Quadratic and Mixed-Integer Quadratic Programming

Quadratic programming (QP) and its extensions form an important class of optimization models relevant to model setups in this thesis. A standard QP takes the form

$$\min_x \ \tfrac{1}{2} x^\top Q x + c^\top x \quad \text{s.t. } Ax \leq b,$$

where Q may be positive semidefinite (yielding a convex program) or indefinite (yielding a nonconvex program). Convex QPs can be solved efficiently using

interior-point or active-set methods (Nocedal and Wright, 2006), whereas nonconvex QPs are $\mathcal{NP}$-hard in general (Pardalos and Vavasis, 1991).

Quadratically constrained quadratic programming (QCQP) is a generalization of QP in which the constraints themselves contain quadratic expressions. QCQPs naturally arise in distance-based clustering (Peng and Wei, 2007), support vector machines (Bennett and Mangasarian, 1992), signal processings (Luo et al., 2010), and geometric fitting problems (Kanatani, 2005). Global optimization methods to solve nonconvex QCQPs rely on semidefinite programming relaxation or the spatial branch-and-bound framework discussed earlier. Mixed-integer quadratic programming (MIQP) and mixed-integer quadratically constrained quadratic programming (MIQCQP) extend these ideas by introducing discrete variables. MIQPs form the foundation for many applications in portfolio optimization (Bertsimas and Cory-Wright, 2022), network design (Shiripour et al., 2012), and energy systems (Yang et al., 2015).

Quadratic structure is directly involved in two of the applications examined in this thesis. First, the k-hyperplane clustering problem in Chapter 4 can be formulated as a MIQCQP, leading naturally to a global optimization approach based on iterative polyhedral approximation combined with spatial branch-and-bound. Second, the recycling center optimization model in Chapter 5 uses quadratic expressions to quantify accessibility and fairness, resulting in a structured MIQP.

2.1.5 Linear Bilevel Programming

Bilevel programming is an optimization framework that captures a hierarchical decision-making process involving two decision-makers, known as the leader and the follower. The leader begins by making a choice at the upper level, selecting a solution. Afterward, the follower, in response to the leader's decision, solves the lower-level optimization problem by determining the values of the lower-level variables. Since the leader's objective function is influenced by both upper- and lower-level variables, solving a bilevel problem necessitates finding a joint solution that optimizes the leader's objective.

Linear bilevel programming can be expressed as:

$$\xi^* = \max_{x,y}\{\mathbf{a}^T x + \mathbf{d}^T y : x \in X, y \in \arg\max_{\hat{y}}\{\mathbf{c}^T \hat{y} : \hat{y} \in Y(x)\}\}, \tag{2.1}$$

where $X = \{x \in \mathbb{R}^q_+ : \mathbf{H}x \leq \mathbf{h}\}$ and $Y(x) = \{y \in \mathbb{R}^n_+ : \mathbf{F}y + \mathbf{L}x \leq f\}$ represent the feasible regions for the leader and follower, respectively.

We assume that the bilevel model follows the optimistic approach, which implies that when faced with multiple lower-level optimal solutions, the follower picks the one

that favors the leader most, i.e., the leader and follower are cooperative. The most common methods for solving bilevel problems involve reformulating them as single-level problems by substituting the linear problem at the lower level with its optimality conditions, such as the Karush-Kuhn-Tucker (KKT) optimality conditions; see, e.g., Zare et al. (2017).

We first replace the lower-level problem with its optimality conditions and then linearize the resulting optimization problem by applying the big-M method. The single-level nonlinear problem is given by:

$$\begin{aligned}
\xi^* = \max_{x,y,\theta} \quad & \mathbf{a}^T x + \mathbf{d}^T y && \text{(2.2a)}\\
\text{s.t.} \quad & \mathbf{H}x \leq \mathbf{h} && \text{(2.2b)}\\
& \mathbf{F}y + \mathbf{L}x \leq \mathbf{f} && \text{(2.2c)}\\
& -\mathbf{F}^T\theta \leq -\mathbf{c} && \text{(2.2d)}\\
& (\mathbf{f} - \mathbf{F}y - \mathbf{L}x)^T\theta = 0 && \text{(2.2e)}\\
& (\mathbf{F}^T\theta - \mathbf{c})^T y = 0 && \text{(2.2f)}\\
& x \in \mathbb{R}^q_+,\ y \in \mathbb{R}^n_+,\ \theta \in \mathbb{R}^m_+. && \text{(2.2g)}
\end{aligned}$$

Let (x^*, y^*, θ^*) be an optimal solution of (2.2). Let M and $\tilde{M}$ be sufficiently large constants such that $M \geq \max\{||\mathbf{f} - \mathbf{F}y^* - \mathbf{L}x^*||_\infty, ||\theta^*||_\infty\}$ and $\tilde{M} \geq \max\{||\mathbf{F}^T\theta^* - \mathbf{c}||_\infty, ||y^*||_\infty\}$. The nonlinear constraints (2.2e) and (2.2f) can be linearized by introducing binary variables $u \in \{0,1\}^m$ and $v \in \{0,1\}^n$ as follows:

$$\begin{aligned}
\xi^* = \max_{x,y,\theta,u,v} \quad & \mathbf{a}^T x + \mathbf{d}^T y && \text{(2.3a)}\\
\text{s.t.} \quad & \text{(2.2b)} - \text{(2.2d)} && \text{(2.3b)}\\
& \mathbf{F}y + \mathbf{L}x \leq \mathbf{f} && \text{(2.3c)}\\
& \mathbf{f} - \mathbf{F}y - \mathbf{L}x \leq Mu && \text{(2.3d)}\\
& \theta \leq M(1-u) && \text{(2.3e)}\\
& \mathbf{F}^T\theta - \mathbf{c} \leq \tilde{M}v && \text{(2.3f)}\\
& y \leq \tilde{M}(\mathbf{1} - v) && \text{(2.3g)}\\
& x \in \mathbb{R}^q_+,\ y \in \mathbb{R}^n_+,\ \theta \in \mathbb{R}^m_+ && \text{(2.3h)}\\
& u \in \{0,1\}^m,\ v \in \{0,1\}^n, && \text{(2.3i)}
\end{aligned}$$

For more detail of the bilevel programming study, see Dempe and Zemkoho (2020). The approximation models formulated in Chapter 6 are in the form of linear bilevel programs. We reformulated the problem into a single-level program using this method.

These deterministic methods form the foundational concepts for the developments in Chapters 3 through 6. In the subsequent sections, we provide background on chance-constrained models and their approximation methods, which serve as the basis for Chapters 6 and 7.

2.2 Chance-Constrained Programming

In many optimization applications, models contain random variables or uncertain parameters that arise from inherently unpredictable phenomena. Stochastic programming provides a framework for incorporating such uncertainties directly into the optimization problem. In its classical form, one assumes that realizations $\omega \in \Omega$ of the uncertain parameters follow a probability distribution and seeks to minimize an expected loss (or cost) function over all possible scenarios.

A particular subclass of stochastic programming is *chance-constrained programming*, in which constraints are required to hold with high probability to ensure system reliability. The formulation was introduced by Charnes and Cooper (1959) and is typically written as

$$\min_{x \in X} f(x) \quad \text{s.t.} \quad \mathbb{P}\{G(x, \omega) \leq 0\} \geq 1 - \varepsilon, \tag{2.4}$$

where ω is a random vector whose probability distribution is a support of set $\omega \subset \mathbb{R}^d$, $G(\cdot, \cdot)$ represents the constraint function, and $\varepsilon \in (0, 1)$ is a risk parameter specifying the allowable probability of violation. This formulation enables decision-makers to balance performance and reliability by explicitly controlling the likelihood of constraint violations. This formulation helps avoid overly conservative decisions in situations where satisfying the constraints for all possible realizations is physically or practically impossible. A wide range of applications of chance-constrained programming includes reliability planning, where service levels must be met with high probability under uncertain demand (Charnes and Cooper, 1958); finance, where portfolio losses are restricted through probabilistic risk limits such as value-at-risk (Pagnoncelli et al., 2009a); production and supply chain management, where firms must satisfy uncertain demand or resource availability with prescribed reliability (Charnes et al., 1958); and energy systems, where operators must ensure reliability despite uncertainty in renewable generation, load, and network conditions (Wang et al., 2012).

In many applications, the constraint function is vector-valued, indicating randomness over stages of decision-making or time periods, denoted by $G(x, \omega) = (G_1(x, \omega), \ldots, G_m(x, \omega))^\top$. This leads to two types of chance constraints. Individual chance constraints require each component to be satisfied with high

probability,

$$\mathbb{P}\{G_i(x,\omega) \leq 0\} \;\geq\; 1-\varepsilon_i, \qquad i=1,\ldots,m, \tag{2.5}$$

with possibly different risk levels ε_i. Another type is the joint chance constraint, which enforces a single reliability requirement for all constraints simultaneously:

$$\mathbb{P}\{G_i(x,\omega) \leq 0 \;\; \forall i=1,\ldots,m\} \;\geq\; 1-\varepsilon. \tag{2.6}$$

Joint chance constraints provide a more direct handle on system-wide reliability but are much harder to solve computationally.

The computational difficulty of solving chance-constrained models is primarily due to the necessity to evaluate multidimensional probabilities and the nonconvex structure of the feasible region generated by the probabilistic requirement. Even when $G(x,\omega)$ is linear and ω has a finite discrete distribution, determining whether a point x satisfies a joint chance constraint is generally $\mathcal{NP}$-hard (Luedtke et al., 2008). For continuous distributions, the problem gets even more difficult: the feasible region is often nonconvex and may have complex geometry, rendering traditional convex optimization tools ineffective. Furthermore, the probability $\mathbb{P}\{G(x,\omega) \leq 0\}$ rarely has a closed-form expression, forcing practitioners to rely on sampling-based approximations, which can be reformulated as a mixed-integer program.

Although chance-constrained models are typically intractable to solve, certain model structures can be solved analytically. For example, when the uncertainty is one-dimensional and the constraint is monotonic in ω, the chance constraint reduces to a quantile condition. That is, the constraint $\mathbb{P}\{\omega \leq g(x)\} \geq 1-\varepsilon$ is equivalent to

$$g(x) \geq F^{-1}(1-\varepsilon),$$

where F^{-1} is the inverse cumulative distribution function (or quantile function) ω. This transforms the chance constraint into a simpler inequality constraint. Another special structure is when ω follows a multivariate normal distribution and $G(x,\omega)$ is affine in ω. Then, the chance constraint $\mathbb{P}\{G(x,\omega) \leq 0\} \geq 1-\varepsilon$ can be reduced to a second-order cone constraint using properties of Gaussian tails (Nemirovski and Shapiro, 2007) as

$$a^\top x + b^\top \mu + \Phi^{-1}(1-\varepsilon)\sqrt{b^\top \Sigma b} \leq c,$$

where $G(x,\omega) = a^\top x + b^\top \omega - c$ with $\omega \sim \mathcal{N}(\mu,\Sigma)$ and Φ^{-1} is the standard normal quantile function.

These closed-form reductions make the problem convex and efficiently solvable. However, this restricts the approach to special structural simplifications under distributional assumptions. In general, chance-constrained problems still require approximation methods in most practical settings, which is the focus of the remainder of this section and of the stochastic models in Chapters 6 and 7.

2.2.1 Sample Average Approximation

Sample average approximation (SAA) is a well-established method for replacing the unknown distribution of ω with an empirical distribution constructed from a set of independent and identically distributed (iid) samples $\{\omega_1, \ldots, \omega_N\}$. These N realizations are considered as a discretization of possible scenarios of the random vector ω. The chance constraint $\mathbb{P}\{G(x,\omega) \leq 0\} \geq 1-\varepsilon$ is equivalent to $\mathbb{P}\{G(x,\omega) > 0\} \leq \varepsilon$. Then, given a risk level $\gamma \in (0,1)$, the chance-constrained model (2.4) is approximated with SAA models,

$$\min_{x \in X} f(x) \quad \text{s.t.} \quad \frac{1}{N}\sum_{j=1}^{N} \mathbb{1}(G(x,\omega_j) > 0) \leq \gamma, \tag{2.7}$$

where $\mathbb{1} : \mathbb{R} \to \mathbb{R}$ is the indicator function whose value is one if $G(x,\omega_j) > 0$ and zero otherwise. The intuition is to find the average of violation counts over realizations when N is large: if $\gamma \leq \varepsilon$, then a feasible solution of the SAA is likely to be feasible for model (2.4), and if $\gamma \geq \varepsilon$, then the optimal value of the SAA is likely to be a lower bound to that of model (2.4). In Pagnoncelli et al. (2009b), the optimal value and the optimal solutions of the SAA model converge to their true counterparts with probability one as N approaches infinity under reasonable regularity assumptions with $\gamma = \varepsilon$.

The SAA model can be formulated as a mixed-integer programming (MIP) model by using a binary variable z_j instead of the indicator function,

$$\min f(x) \tag{2.8a}$$
$$\text{s.t. } G(x,\omega_j) \leq M_j z_j, \quad j = 1,\ldots,N, \tag{2.8b}$$
$$\sum_{j=1}^{N} z_j \leq \gamma N, \tag{2.8c}$$
$$z_j \in \{0,1\}, \quad j = 1,\ldots,N, \tag{2.8d}$$
$$x \in X, \tag{2.8e}$$

where M_j is a large positive number with $M_j \geq \max_x G(x,\omega_j)$ for all $j = 1,\ldots,N$.

In the simplest cases, when both functions f and G are linear in x and the set X is polyhedral, moderate-sized instances of the MIP (2.8) are nevertheless challenging to solve with state-of-the-art solvers. However, particular structures of the model (2.8), such as joint chance-constrained formulations where uncertain parameters appear only on the right-hand side, for instance $G(x,\omega) = \max_i\{\omega_i - G_i(x)\}$, are computationally more tractable. From the constraint $\sum_{j=1}^{N} z_j \leq \gamma N$, it follows that at most $p := \lfloor \gamma N \rfloor$ scenarios can fail. Assuming without loss of generality that $\omega_i^1 \geq \omega_i^2 \geq \ldots \geq \omega_i^N$, the constraints $v_i + \omega_i^j z_j \geq \omega_i^j$ are redundant for

$j = p+1, \ldots, N$. The mixing set developed in Günlük and Pochet (2001); Luedtke et al. (2008) defined as

$$M_i := \left\{ (v_i, z_1, \ldots, z_N) \in \mathbb{R}_+ \times \{0,1\}^N : v_i + (\omega_i^j - \omega_i^{p+1}) z_j \geq \omega_i^j,\ j = 1, \ldots, N \right\} \quad (2.9)$$

has a well-studied convex hull that provides valid inequalities, enabling the joint chance-constrained model to be solved more efficiently in this setting.

In summary, integrating SAA with strong valid inequalities is an effective approach to solving moderate-scale chance-constrained problems exactly while maintaining interpretable probabilistic guarantees. We solve the exact SAA in the setting of Chapter 7, where computing the profit for each setting is tractable. However, in the setting of Chapter 6, we introduce an alternative framework that bounds the two-stage chance-constrained model using approximation methods based on probabilistic bounding in order to capture risk-seeking behavior.

2.2.2 Two-Stage Chance-Constrained Formulations

The main model studied in this thesis is a two-stage chance-constrained formulation, where the first-stage decisions are made before the realization of the uncertainty. Subsequently, a recourse decision is optionally taken to cover any shortfall in specific scenarios, incurring a corresponding penalty cost in the objective.

A generic two-stage chance-constrained model can be written as

$$\begin{aligned}
\min \quad & c^\top x + \mathbb{E}_\omega[Q(x,\omega)] && (2.10a)\\
\text{s.t.} \quad & Ax = b, && (2.10b)\\
& \mathbb{P}(T(\omega)x + Wy(\omega) \leq h(\omega)) \geq 1 - \varepsilon, && (2.10c)\\
& x \geq 0,\ y(\omega) \geq 0, && (2.10d)
\end{aligned}$$

where the recourse function for a scenario ω is defined as

$$Q(x,\omega) := \min_y \left\{ q^\top y : Wy \leq h(\omega) - T(\omega)x,\ y \geq 0 \right\}.$$

Constraint (2.10c) requires that the second-stage system remains feasible for at least a $(1-\varepsilon)$ portion of all realizations. The SAA replaces the true probability with its empirical counterpart over N sampled scenarios, producing the following

deterministic equivalent formulation:

$$\min \quad c^\top x + \frac{1}{N}\sum_{j=1}^{N} q^\top y^j \tag{2.11a}$$

$$\text{s.t.} \quad Ax = b, \tag{2.11b}$$

$$Wy^j \le h(\omega^j) - T(\omega^j)x + Mz_j, \qquad j = 1,\dots,N, \tag{2.11c}$$

$$\sum_{j=1}^{N} z_j \le \varepsilon N, \tag{2.11d}$$

$$x \ge 0,\ y^j \ge 0,\ z_j \in \{0,1\}. \tag{2.11e}$$

The model (2.11) can be solved directly with the state-of-the-art MIP solvers. Two-stage formulations of this kind provide a flexible template for modeling decision-making under uncertainty with recourse. In this thesis, they form the basis of the energy-market model in Chapter 7, whereas in the risk-seeking framework in Chapter 6 approximation methods are used to provide bounds for the models.

2.2.3 Boole–Bonferroni Approximations and Probabilistic Bounding

We consider the joint chance constraint (2.6) and define, for each constraint index i, the *failure event*

$$A_i := \{\, \omega : G_i(x,\omega) > 0 \,\},$$

with $A_i^{\complement}$ denoting the corresponding success event. Feasibility of all m constraints in scenario ω therefore corresponds to $\omega \in \bigcap_{i=1}^{m} A_i^{\complement}$. The joint chance constraint can thus be written as

$$\mathbb{P}\Big(\bigcap_{i=1}^{m} A_i^{\complement}\Big) = 1 - \mathbb{P}\Big(\bigcup_{i=1}^{m} A_i\Big) \ \ge\ 1-\varepsilon,$$

which is equivalent to requiring

$$\mathbb{P}\Big(\bigcup_{i=1}^{m} A_i\Big) \le \varepsilon. \tag{2.12}$$

Hence, enforcing a joint chance constraint reduces to controlling the probability that the system fails in at least one constraint.

We define the individual and pairwise failure probabilities

$$S_1 := \sum_{i=1}^{m} \mathbb{P}(A_i), \qquad S_2 := \sum_{1\le i<j\le m} \mathbb{P}(A_i \cap A_j).$$

Using the inclusion–exclusion identity, the probability of the union of failure events satisfies

$$\mathbb{P}\Big(\bigcup_{i=1}^{m} A_i\Big) = S_1 - S_2 + S_3 - \cdots .$$

Classical probabilistic bounds, such as the Boole–Bonferroni inequalities (Bonferroni, 1936), approximate the union probability by truncating the inclusion–exclusion expansion. In particular,

$$\mathbb{P}\Big(\bigcup_{i=1}^{m} A_i\Big) \leq S_1, \qquad \mathbb{P}\Big(\bigcup_{i=1}^{m} A_i\Big) \geq S_1 - S_2.$$

These bounds allow the joint chance constraint to be approximated by replacing the union probability with an analytically simpler upper or lower bound. When the left hand side term of (2.12) is replaced with an *upper* bound, the feasible region becomes more restrictive, yielding a valid *lower bound* on the optimal objective value in a maximization problem. Conversely, replacing the left hand side term of (2.12) with a *lower* bound enlarges the feasible region and yields an *upper bound*.

The Boole's inequality is widely used to derive tractable deterministic approximations of joint chance constraints, since it replaces the original probabilistic constraint with

$$\sum_{i=1}^{m} \mathbb{P}(A_i) \leq \varepsilon.$$

By further allocating the risk evenly as $\varepsilon_i = \frac{\varepsilon}{m}$, the joint chance constraint can be approximated by a set of individual chance constraints:

$$\mathbb{P}(A_i) \leq \varepsilon_i, \qquad i = 1, \ldots, m.$$

This preserves convexity and enables efficient solution techniques; see, e.g., Vitus and Tomlin (2011); Baker and Toomey (2017). Such bounding techniques will be used conceptually in Chapter 6 of this thesis when constructing approximations for risk-seeking chance-constrained models.

Chapter 3

The Prime Programming Problem: Formulations and Solution Methods

This chapter introduces the prime programming problem, a deterministic subclass of integer programming where feasible solutions are restricted to prime numbers. Then, we demonstrate how several classical problems in number theory can be formulated as prime programs. To solve such problems with a commercial optimization solver, we extend the branch-and-bound procedure of integer programming and the cutting planes method. To reduce the computational effort required by this branch-and-bound method, we utilize the special structure of prime numbers to incorporate new branching rules and variable fixing strategies. We present numerical results using a variety of such strategies on a classic conjecture on linear equations in prime numbers. Finally, by employing the concept of the inference dual of an integer program, we show how to perform sensitivity analysis on general prime programs; this allows us to quickly compute changes in the optimal objective function value for small perturbations in the constraints. We publicly release all of our code to solve a general prime programming model.

Note: a preprint of this work is available at `https://optimization-online.org/?p=27681`.

3.1 Introduction

Consider the following optimization problem

$$\begin{aligned} \min \quad & f(x,y,z) && \text{(3.1a)} \\ \text{s.t.} \quad & g(x,y,z) \geq 0, && \text{(3.1b)} \\ & x \in \mathbb{P}, y \in \mathbb{R}^+, z \in \mathbb{Z}^+, && \text{(3.1c)} \end{aligned}$$

where $\mathbb{Z}^+$ denotes the set of non-negative integers, $\mathbb{R}^+$ denotes the set of non-negative real numbers, and $\mathbb{P} = \{2,3,5,\ldots\} \subset \mathbb{Z}^+$ denotes the set of prime numbers. Then, model (3.1) is a standard mixed-integer non-linear program with the inclusion of an additional set of variables, x, restricted to be prime. We call model (3.1) as a *Prime Program (PP)*. When functions f and g are linear, model (3.1) is a mixed-integer linear program with prime number constraints. Such models are pervasive in the number theory literature although they are often not formulated in the form of a mathematical program such as optimization model (3.1). For example, the famous Goldbach's Conjecture rests on proving feasibility of the constraint $2a + 2 = x_1 + x_2$ where $x_1, x_2 \in \mathbb{P}$ for any given $a \in \mathbb{Z}^+$, or its infeasibility for some a. Similarly, the Vinogradov's Theorem says that the constraint $2a + 1 = x_1 + x_2 + x_3$ where $x_1, x_2, x_3 \in \mathbb{P}$ is always feasible for positive integers $a \geq 3$; analogously, the Chen's Theorem says that either of the two constraints $2a = x_1 + x_2$ for some $x_1, x_2 \in \mathbb{P}$ or $2a = x_3 + x_4 \cdot x_5$ for some $x_3, x_4, x_5 \in \mathbb{P}$ is always feasible for a given sufficiently large $a \in \mathbb{Z}^+$. For an introduction to several such problems, see, e.g., Wells (2005).

There is some work at the intersection of mathematical optimization and prime numbers. One stream of work develops heuristics to identify or construct large prime numbers, see, e.g., Knezevic (2021) that uses genetic programming to generate large prime numbers. Dass et al. (2013) study the prime factorization problem using three metaheuristics. Both these works do not formulate an explicit mathematical program, as is standard in the optimization community, to solve the underlying problem using a numerical optimization solver. In contrast, Mehrotra and Pal (2018) develop various objective functions that, when optimized over, solve instances of prime factorization. The work by Kumar (2022) is more general — although without prime number constraints — and, formulates non-linear Diophantine equations as complete mathematical optimization models. For instance, the nonlinear Diophantine $a_1 x_1^p + a_2 x_2^p + \ldots + a_n x_n^p = b$ is reformulated as $\min_x \left(a_1 x_1^p + a_2 x_2^p + \ldots + a_n x_n^p - b\right)^2$ where x is constrained within a feasible region of integers. Our work is in a similar spirit to this: we formulate problems concerning prime numbers as general mathematical optimization models. Such mathematical programs are then easily solvable with standard optimization solvers up to the solver's numerical limits. As such, we introduce the PP in the discrete optimization community.

The structure of this chapter is as follows. In Section 3.2, we motivate this work by studying six elementary problems from number theory and formulate them as prime programs. In their current form, such problems are not solvable with an optimization solver (e.g., CPLEX or Gurobi), since there is no analytical expression to formulate the restriction $x \in \mathbb{P}$. Thus, in Section 3.3, we present a method to solve a PP by modifying the standard B&B scheme for solving integer problems together with the cutting planes method. Further, we conduct sensitivity analysis for a PP that allows quick solutions of a slightly perturbed problem without resolving it entirely. Then, we present problems that can be formulated into PPs as our case studies in Section 3.4. In Section 3.5, we present detailed computational experiments and analysis by employing an example of finding a sequence of distinct primes such that the arithmetic mean of any two of these primes is also a prime. We formulate this problem as a PP, and we present additional solution strategies that extend our naive B&B method. We conclude in Section 3.6 and provide additional numerical examples and results in the appendix. We provide all of our code to solve a PP at our GitHub repository cited below.

3.2 Preliminaries

A classic problem in number theory, that is expressible as an integer program, is the computation of the greatest common divisor (gcd) of two integers (Conforti et al., 2014, Chapter 1). For $a, b \in \mathbb{Z}^+$, the gcd of a and b is given by $\gcd(a,b) = \max_z z$ s.t. $\{z$ divides both a and b with $z \in \mathbb{Z}^+\}$. Then, the optimization model

$$\gcd(a,b) = \min_x ax_1 + bx_2 \text{ s.t.}\{ax_1 + bx_2 \geq 1 \text{ with } x_1, x_2 \in \mathbb{Z}\} \tag{3.2}$$

computes the gcd; see, Proposition 1.6 of Conforti et al. (2014) for a proof. Enforcing the additional restriction of solutions, (x_1, x_2), being prime numbers transforms model (3.2) into a PP. In this section, we provide optimization models for six elementary problems in number theory considered in the classic textbook of Sierpiński (1970). We let $\bar{p}_x$ and $\underline{p}_x$ denote the smallest prime number greater than $x \in \mathbb{R}^+$ and the largest prime number lesser than $x \in \mathbb{R}^+$, respectively.

Problem 1. Find four triplets of solutions, (x_1, x_2, x_3), of the equation $x_1^2 + 1 = x_2^2 + x_3^2$ with $x_1, x_2, x_3 \in \mathbb{P}$.

Solution. The corresponding optimization model is:

$$\min_x 0 \text{ s.t.}\{x_1^2 + 1 = x_2^2 + x_3^2 \text{ with } x_1, x_2, x_3 \in \mathbb{P}\}. \tag{3.3}$$

A solution to model (3.3) provides one such triplet, (x_1^*, x_2^*, x_3^*). In Section 3.3.1 we describe a procedure to solve this problem, and obtain *an* optimal solution

$(x_1^*, x_2^*, x_3^*) = (7, 5, 5)$. To obtain another solution, we add one cutting plane: $x_1 \geq \bar{p}_{x_1^*}$; equivalently, we can add the cutting planes $x_2 \geq \bar{p}_{x_2^*}$, $x_3 \geq \bar{p}_{x_3^*}$, or the weaker cutting plane $x_1 \geq x_1^* + 1$. We continue this process until we obtain four triplets of solutions. Again, using the procedure in Section 3.3.1, we obtain $(7, 5, 5)$, $(13, 7, 11)$, $(17, 11, 13)$, and $(23, 13, 19)$ as four triplets.

An equivalent optimization formulation is obtained by (i) using the identity $(5z + 13)^2 + 1 = (3z + 7)^2 + (4z + 11)^2, \forall z \in \mathbb{Z}^+$, and (ii) the fact that there are infinitely many primes of the form $az + b$ when $\gcd(a, b) = 1$. Then, the following optimization model is equivalent to model (3.3):

$$\min_{x,z} 0 \text{ s.t.} \{x_1 = 5z + 13, x_2 = 3z + 7, x_3 = 4z + 11 \text{ with } x_1, x_2, x_3 \in \mathbb{P}, z \in \mathbb{Z}^+\}. \quad (3.4)$$

A variation of this problem is to enforce the largest of these primes, x_1, to be the smallest prime number feasible to model (3.3). Then, we simply change the objective function of model (3.3); the corresponding model is

$$\min_{x} x_1 \text{ s.t.} \{x_1^2 + 1 = x_2^2 + x_3^2 \text{ with } x_1, x_2, x_3 \in \mathbb{P}\}. \quad (3.5)$$

□

Problem 2. Find all solutions of the equation $x_1(x_1 + 1) + x_2(x_2 + 1) = x_3(x_3 + 1)$ with $x_1, x_2, x_3 \in \mathbb{P}$.

Solution. The corresponding optimization model is:

$$\min_{x} 0 \text{ s.t.} \{x_1(x_1 + 1) + x_2(x_2 + 1) = x_3(x_3 + 1) \text{ with } x_1, x_2, x_3 \in \mathbb{P}\}. \quad (3.6)$$

Model (3.6) is a non-linear integer optimization model due to the product of integer variables but is easily linearized with a McCormick envelope, see, e.g., Wolsey (1998). Again, using the procedure in Section 3.3.1, one such optimal triplet is $(x_1^*, x_2^*, x_3^*) = (2, 2, 3)$; and, for the next solution, we add a similar cutting plane as in Problem 1. However, in this case the solution is unique as we show below; thus, the computational method of adding cutting planes yields the same solution.

To verify uniqueness of the solution $(2, 2, 3)$, consider the constraint of model (3.6) which is equivalent to $x_1(x_1 + 1) = (x_3 - x_2)(x_3 + x_2 + 1)$. From this equation, we immediately have that the prime number x_1 divides at least one of $x_3 - x_2$ or $x_3 + x_2 + 1$. We distinguish two exclusive cases below.

(i) Consider x_1 divides $x_3 - x_2$. Then, we have $x_1 \leq x_3 - x_2$, and it follows that $x_1(x_1 + 1) \leq (x_3 - x_2)(x_3 - x_2 + 1)$. Hence, $x_3 + x_2 + 1 \leq x_3 - x_2 + 1$, which is false since $x_2 \in \mathbb{Z}^+$.

(ii) Consider x_1 divides $x_3 + x_2 + 1$. Then, there exists a $k \in \mathbb{Z}^+$ such that $x_3 + x_2 + 1 = kx_1$, hence $x_1 + 1 = k(x_3 - x_2)$. If $k = 1$, then $x_3 + x_2 + 1 = x_1$ and $x_1 + 1 = x_3 - x_2$. However, this yields $x_1 - x_2 = x_3 + 1$ and $x_1 + x_2 = x_3 - 1$, which cannot be true since $x_2 \in \mathbb{Z}^+$. Now, consider $k \geq 2$ and the identity $2x_2 = (x_3 + x_2) - (x_3 - x_2)$. We have

$$\begin{aligned} 2x_2 =& (x_3 + x_2) - (x_3 - x_2) \\ =& (kx_1 - 1) - (x_3 - x_2) \\ =& k[k(x_3 - x_2) - 1] - 1 - (x_3 - x_2) \\ =& (k+1)[(k-1)(x_3 - x_2) - 1]. \end{aligned}$$

Since x_2 is prime, the divisors of $2x_2$ are 1, 2, x_2, and $2x_2$. Further, since $k + 1 \geq 3$ and divides $2x_2$, either $k + 1 = x_2$ or $k + 1 = 2x_2$. Again, we distinguish the two exclusive cases below.

(a) Consider $k + 1 = x_2$. Then $(k - 1)(x_3 - x_2) = 3$, hence $(x_2 - 2)(x_3 - x_2) = 3$. Thus, either $x_2 - 2 = 1$ and $x_3 - x_2 = 3$, or $x_2 - 2 = 3$ and $x_3 - x_2 = 1$. In the first case, $x_2 = 3$ and $x_3 = 6$, which is not prime. In the second case, $x_2 = 5$ and $x_3 = 6$, again resulting in a non-prime solution for x_3.

(b) Consider $k + 1 = 2x_2$. Then $(k - 1)(x_3 - x_2) = 2$, hence $2(x_2 - 1)(x_3 - x_2) = 2$. Thus, $x_2 - 1 = x_3 - x_2 = 1$, hence $x_2 = 2$ and $x_3 = 3$. In this case, we obtain the solution $(x_1^*, x_2^*, x_3^*) = (2, 2, 3) \in \mathbb{P}$ which is that computed by our procedure above.

Similar to Problem 1, a variation of this problem is to enforce the largest of these primes, x_3, to be the smallest prime number feasible to model (3.6). Then, we simply change the objective function of model (3.6); the corresponding model is

$$\min_x x_3 \text{ s.t.} \{x_1(x_1 + 1) + x_2(x_2 + 1) = x_3(x_3 + 1) \text{ with } x_1, x_2, x_3 \in \mathbb{P}\}. \tag{3.7}$$

□

Problem 3. Find all solutions $z \in \mathbb{Z}^+$ with $z + 1, z + 3, z + 7, z + 9, z + 13, z + 15 \in \mathbb{P}$.

Solution. The corresponding optimization model is:

$$\begin{aligned} \min_x 0 \text{ s.t.} \{&x_1 = z + 1, x_2 = z + 3, x_3 = z + 7, x_4 = z + 9, \\ &x_5 = z + 13, x_6 = z + 15 \text{ with } x_1, \ldots, x_6 \in \mathbb{P}, z \in \mathbb{Z}^+\}. \end{aligned} \tag{3.8}$$

In model (3.8), the integrality restriction, $z \in \mathbb{Z}^+$, is replaceable with its continuous relaxation, $z \in \mathbb{R}^+$. Again, using the procedure in Section 3.3.1, one feasible tuple of

solutions is $(x_1^*, x_2^*, x_3^*, x_4^*, x_5^*, x_6^*, z^*) = (5, 7, 11, 13, 17, 19, 4)$; for the next solution, we add a similar cutting plane as in Problem 1 and continue this process.

However, again similar to Problem 2, the solution of model (3.8) is unique. To verify this claim, we note that a feasible solution is of the form $\{x_1^*, x_2^*, x_3^*, x_4^*, x_6^*\} = \{a+1, a+3, a+7, a+9, a+15\}$, where $a \in \mathbb{Z}^+$. This set has exactly one element divisible by 5. Hence, if the smallest number of this set, $a+1$, exceeds 5, then all the numbers cannot simultaneously be prime. Since $z = 1, 2, 3$ are all infeasible, the obtained solution $z^* = 4$ is unique. Indeed, adding a constraint $z \geq 5$ (or, equivalently $x_1 \geq 6$) to the optimization solver renders the model infeasible.

Similar to Problem 1 and Problem 2, a variation of the problem is the following model:

$$\begin{aligned} \min_{x,z} z \text{ s.t.} \{ & x_1 = z+1, x_2 = z+3, x_3 = z+7, x_4 = z+9, \\ & x_5 = z+13, x_6 = z+15 \text{ with } x_1, \ldots, x_6 \in \mathbb{P}, z \in \mathbb{R}^+ \}. \end{aligned} \tag{3.9}$$

□

Problem 4. Find all primes which can be represented both as sums and differences of two primes.

Solution. The corresponding optimization model is:

$$\min_x 0 \text{ s.t.} \{ x_1 = x_2 + x_3, x_1 = x_4 - x_5 \text{ with } x_1, \ldots, x_5 \in \mathbb{P} \}. \tag{3.10}$$

Here, the decision variables $(x_2^*, x_3^*, x_4^*, x_5^*)$ are auxiliary variables while x_1^* provides the required solution. Then, we obtain $x_1^* = 5$ with our procedure of Section 3.3.1. For the next solution, we proceed by adding cutting planes similar to Problems 1-3, e.g., $x_1 \geq \bar{p}_{x_1^*}$. However, model (3.10) is simplified further by a few observations. All prime numbers except 2 are odd, and hence $x_2 + x_3$ is an even number (i.e., not prime) unless exactly one of x_2 and x_3 is 2. Without loss of generality, set $x_3 = 2$. Similarly, we have $x_5 = 2$. We thus have the equivalent optimization model:

$$\min_x x_1 \text{ s.t.} \{ x_1 = x_2 + 2, x_1 = x_4 - 2 \text{ with } x_1, x_2, x_4 \in \mathbb{P} \}. \tag{3.11}$$

Again, the solution to this problem is unique. To verify this claim, let $x_1^* = a$ for some $a \in \mathbb{Z}^+$. Consider the set $\{x_1^*, x_2^*, x_4^*\} = \{a, a-2, a+2\}$. This set has exactly one element divisible by 3 since any integer a is of the form of either $a = 3k$, $a = 3k+1$, or $a = 3k+2$ for some $k \in \mathbb{Z}^+$; in each of the three cases, exactly one element is divisible by 3. Hence, all elements of the set are at most three for them to be prime. Since $x_1 = 1, 2, 3, 4$ are all infeasible, the obtained solution $x_1^* = 5$ is unique. Again, adding a constraint $x_1 \geq 6$ to the optimization solver renders the model infeasible.

□

Problem 5. Find the five least positive integers, z, such that each of the integers z, $z+1$, $z+2$ is a product of two different primes.

Solution. The corresponding optimization model is:

$$\min_x z \text{ s.t.}\{z = x_1x_2, z+1 = x_3x_4, z+2 = x_5x_6 \text{ with } x_1,\ldots,x_6 \in \mathbb{P}, z \in \mathbb{Z}^+\}. \quad (3.12)$$

To ensure that the primes are different, we enforce $x_1 \neq x_2$, $x_3 \neq x_4$, and $x_5 \neq x_6$. Without loss of generality, we add an additional set of constraints: $x_{i+1} \geq x_i + 1, i = 1,3,5$. Then, the solution to model (3.12) provides one feasible value for z; we obtain $(x_1^*, x_2^*, x_3^*, x_4^*, x_5^*, x_6^*, z^*) = (3,11,2,17,5,7,33)$ as the solution using the procedure in Section 3.3.1. For the next solution, we add a cutting plane: $z \geq z^* + 1$. We repeat this process until we get five solutions for z obtaining $z^* = 33, z^* = 85, z^* = 93, z^* = 141$, and $z^* = 201$, respectively. We are unaware of how many such solutions exist.

□

Problem 6. Find all integers, z, such that each of the six integers z, $z+2$, $z+6$, $z+8$, $z+12$, $z+14$ are primes.

Solution. The corresponding optimization model is:

$$\begin{aligned}\min_x 0 \text{ s.t.}\{&x_2 = x_1+2, x_3 = x_1+6, x_4 = x_1+8,\\ &x_5 = x_1+12, x_6 = x_1+14 \text{ with } x_1,\ldots,x_6 \in \mathbb{P}\}.\end{aligned} \quad (3.13)$$

Then, we obtain one feasible solution to model (3.13) using the procedure in Section 3.3.1 as $(x_1^*,\ldots,x_6^*) = (5,7,11,13,17,19)$. The uniqueness of this solution is verifiable in a method similar to that for Problem 4. Let $x_1^* = a$ for some $a \in \mathbb{Z}^+$. Consider the set $\{x_1^*, x_2^*, x_3^*, x_4^*, x_6^*\} = \{a, a+2, a+6, a+8, a+14\}$ which has exactly one element divisible by 5. Thus, $x_1 \leq 5$ is a valid inequality. Since, $x_1 = \{2,3\}$ is not feasible (and, $x_1 = \{1,4\}$ is not prime), the obtained solution $x_1^* = 5$ is unique. □

In Appendix A.1, we provide three further illustrative examples. Although Problems 1-6 are "toy" problems, they demonstrate how problems in number theory are expressible as PPs. Further, they show how such formulations allow an easy application of a commercial solver, if available; we describe the details of such a solution method in Section 3.3.1. However, the above formulations mask three subtleties from the original problems of number theory.

(i) First, our optimization problems implicitly upper bound the decision variables, e.g., the solver Gurobi treats numbers larger than 10^{20} as infinite (Gurobi

Optimization, LLC, 2024). Thus, the optimization problems are unable to compute solutions larger than this value. Although such large limits are expected to be reasonable for most practical problems, problems in number theory do not have such a restriction. Scaling the numbers is one possible direction to overcome this.

(ii) Second, computing *all* solutions requires running the procedure possibly an exponential number of times; even then we are only guaranteed to obtain solutions up to the solver-imposed implicit upper bound. This exhaustive enumeration is expected as most problems in number theory (and, IP) are $\mathcal{NP}$-complete. Even verifying uniqueness of an integer solution of a linear system of equations is $\mathcal{NP}$-complete (Mangasarian and Ferris, 2010). However, when a finite number of solutions are required—such as, Problem 1 and Problem 5—we expect the solutions to be obtained in a few repetitions, particularly if good cuts are identified. Indeed, in the procedure we describe in Section 3.3.1, we obtain solutions for these two problems in four and five repetitions, respectively. Further, several problems in number theory typically have solutions that are small prime numbers. For example, in Problem 6 we observe that $z \leq 5$. This is because for $z > 5$ at least one element in the set $\{z+1, z+3, z+7, z+9, z+15\}$ is divisible by 5 and, hence, is not prime.

(iii) Third, the above illustrated optimization procedure rests on a scheme to determine $\bar{p}_x$ given x; this provides an added cutting plane. Unfortunately, there is no analytical formula to determine the prime number immediately proceeding a given integer. However, several theorems from the number theory literature assist in this computation. For example, Bertrand's postulate sets the range of $\bar{p}_x \in (x, 2x-2), \forall x \in \mathbb{P} \setminus \{2,3\}$, see, e.g., Sylvester (1881). In computational practice, as we also mention above, instead of the cut $x \geq \bar{p}_{x^*}$ we can simply add the relaxed cut $x \geq x^* + 1$ which allows the solver to compute a different optimal solution than x. We build up on this to construct a modified branch-and-bound procedure in the next section.

3.3 Solving Prime Programs

3.3.1 Branch-and-bound for Prime Programs

We now consider the PP presented in model (3.1) with linear functions as follows:

$$z^* = \min \quad cx \tag{3.14a}$$

$$\text{s.t.} \quad Ax \geq b, \tag{3.14b}$$

$$x_i \in [2, M_i] \cap \mathbb{P}, \quad i = 1, \ldots, n, \tag{3.14c}$$

where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, $M_i \in \mathbb{Z}^+, \forall i = 1, \ldots, n$. Here, M denotes a vector of upper bounds on feasible prime numbers. If such a vector of large enough prime numbers is available, we use that; however, if this is difficult to estimate, we employ large enough integers. In the absence of the prime constraint given by equation (3.14c), model (3.14) is solvable via the standard B&B procedure of IP, see, e.g., Wolsey (1998). We now adapt this procedure to solve model (3.14).

To this end, we first solve the LP relaxation of model (3.14). If the solution, $x^* = [x_1^*, \ldots x_n^*]$, is prime we stop at the root node itself since the solution is optimal. If at least one element is not prime, we select the element which is farthest from its closest prime to branch on; this is similar to the so-called most fractional rule in IP. Given the non-prime branching variable, i, we create two nodes, right and left, that include the additional constraints $x_i \geq \bar{p}_{x_i^*}$ and $x_i \leq \underline{p}_{x_i^*}$, respectively. We store these in a heap, $\mathcal{H}$, in this order. We then explore the last node (i.e., the left child node) in the heap via a depth-first strategy. We prune nodes if one of the following conditions holds: (a) the node is optimal; i.e., the relaxed solution is prime, (b) the node is infeasible, or (c) the objective function value of the relaxed solution at that node is not better than that of the incumbent; i.e., the node is pruned if the objective function value of the node is greater than the incumbent's objective function value in a minimization problem. When a node is pruned, we remove it from the heap. We terminate the algorithm when all nodes are explored or pruned; i.e., $\mathcal{H} = \phi$. We note that solving the LP relaxation with such a branching procedure is sufficient as it partitions the feasible space into disjoint spaces without cutting off any feasible solution. We illustrate this B&B procedure with the following numerical example.

$$\begin{aligned}
\min \quad & -x_1 - 4x_2 && \text{(3.15a)}\\
\text{s.t.} \quad & -3x_1 - x_2 \geq -2000, && \text{(3.15b)}\\
& -x_1 - 5x_2 \geq -450, && \text{(3.15c)}\\
& x_1, x_2 \in [2, 100] \cap \mathbb{P}. && \text{(3.15d)}
\end{aligned}$$

Figure 3.1 presents the B&B tree for model (3.15) that we summarize here; nodes marked with white are pruned below while those in black are further branched. We first strengthen the upper bound of $M = 100$ in constraint (3.15d) with $M = \underline{p}_{100} = 97$. Then, the LP solution at the root node is $x^* = [97, 70.6]$. Since x_1 is prime, we branch on x_2. We identify $\bar{p}_{70.6} = 71$ (right branch) and $\underline{p}_{70.6} = 67$ (left branch). We proceed to the left child-branch first (marked "2"); it is pruned, identifying an incumbent solution. We then backtrack to the right branch at node "3", which is the next element in the heap, and continue the process. Node "5" is pruned via optimality and we backtrack to node "4" to its right branch. Continuing, we identify a new incumbent solution at node "8". The right branch at node "9" identifies a solution that is not prime which is then pruned by bound. Backtracking we proceed first to the right branch of node "6" (i.e., node "10") and then to the right branch of node "3" (i.e., node

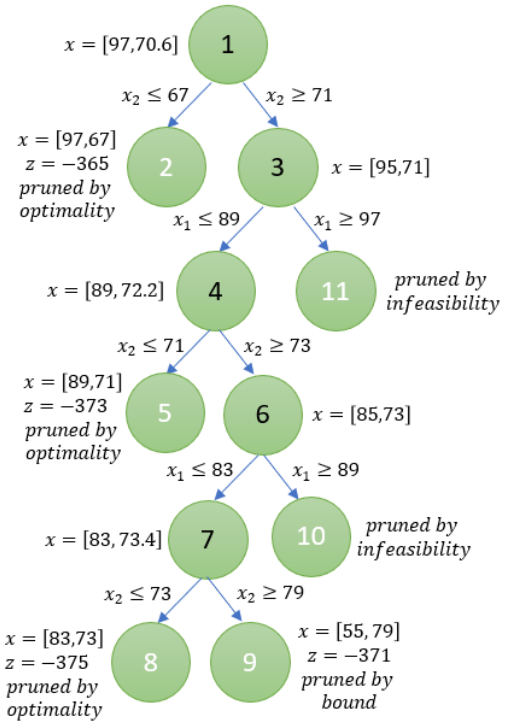


FIGURE 3.1: Enumeration tree for the prime B&B procedure applied to model (3.15). The six nodes marked in white are leaf nodes that are pruned below for reasons indicated, while the other five nodes are further branched on. For details, see Section 3.3.1.

"11") — both are pruned by infeasibility. The optimal solution is then $x^* = [83, 73]$ with an optimal objective function value of -375. The enumeration tree contains 11 nodes, six of which are leaf nodes.

We note that the idea of branching presented above is not restricted to PPs where the solution variables are restricted to prime numbers, but it is applicable to any set of discrete values. For example, if we need the solution to be Fibonacci numbers, we can branch on the structure of the set of Fibonacci sequences. This branching split the feasible region without eliminating any feasible points.

3.3.2 Cutting Planes for Prime Programs

As prime programming is a subclass of integer programming, the derivation of cutting planes for integer programs is still applicable to prime programs. We present cutting plane methods where the derivations are based on fractional values in the relaxation solution. Mixed-integer rounding and Gomory cuts are these types, and they can be improved within the prime program by considering the gap of prime numbers instead of the gap of one in the integer program.

Let $X = \{(x, y) \in \mathbb{P} \times \mathbb{R}^+ \mid x + y \geq b\}$. We consider two branches, either $x \leq \underline{p}_b$ or $x \geq \overline{p}_b$. Assume $x \leq \underline{p}_b$, the inequality is equivalent to $y \geq b - x$. Then the tightest bound is at $x = \underline{p}_b$, which requires $y \geq b - \underline{p}_b = \delta$. Thus, the point $V_1 = (\underline{p}_b, \delta)$ is the extreme point of this sub-region. On the other hand, $x \geq \overline{p}_b$, the variable x already satisfies the constraint; the extreme point of this feasible sub-region is $V_2 = (\overline{p}_b, 0)$. The mixed-prime rounding cut, the inequality that passes through these two points, can be obtained as follows:

$$y - 0 \geq \frac{0 - \delta}{\overline{p}_b - \underline{p}_b} \left(x - \underline{p}_b \right), \tag{3.16}$$

where $\delta = b - \underline{p}_b$. This is equivalent to

$$y \geq \delta - \frac{\delta}{\Delta} \left(x - \underline{p}_b \right). \tag{3.17}$$

Next, we consider the Gomory mixed-integer cuts in prime programs. Assume an optimal basis of LP relaxation takes a non-prime value b_i. The tableau row is:

$$x_i + \sum_{j \in N} a_{ij} x_j = b_i, \tag{3.18}$$

where N is the set of non-basic variables. Then, we define the left prime remainder $f = b_i - \underline{p}_{b_i}$, the right prime remainder $g = \overline{p}_{b_i} - b_i$, and the prime gap $\Delta = f + g$. Similarly, we consider the prime split disjunction $x_i \leq \underline{p}_{b_i} \vee x_i \geq \overline{p}_{b_i}$. In the first case, we have $b_i - \sum_j a_{ij} x_j \leq \underline{p}_{b_i}$ which is equivalent to $\sum_j a_{ij} x_j \geq f$. To satisfy the constraint using $x_j \geq 0$, we consider only positive coefficients $a_{ij} > 0$, we have

$$\sum_{j : a_{ij} > 0} \frac{a_{ij}}{f} x_j \geq 1. \tag{3.19}$$

In the case of $x_i \geq \overline{p}_{b_i}$, we have $b_i - \sum_j a_{ij} x_j \geq \overline{p}_{b_i}$ which is equivalent to $\sum_j -a_{ij} x_j \geq g$. To satisfy the constraint using $x_j \geq 0$, we consider only negative coefficients $a_{ij} < 0$, we have

$$\sum_{j : a_{ij} < 0} \frac{a_{ij}}{g} x_j \geq 1. \tag{3.20}$$

Because any valid prime assignment must satisfy at least one side of the disjunction, combining these requirements yields the valid prime Gomory cut:

$$\sum_{j : a_{ij} > 0} \left(\frac{a_{ij}}{f} \right) x_j + \sum_{j : a_{ij} < 0} \left(\frac{a_{ij}}{g} \right) x_j \geq 1. \tag{3.21}$$

3.3.3 Sensitivity Analysis for Prime Programs

We now describe a procedure to conduct sensitivity analysis on a linear PP. This allows us to compute changes in the optimal objective function value for small perturbations in the problem parameters without re-solving the entire program. To this end, we employ the technique of the inference dual of an integer program as proposed in Dawande and Hooker (2000). Consider the following problem that formulates a perturbation in model (3.14).

$$z^*_\delta = \min \quad (c + c_\delta)x \tag{3.22a}$$
$$\text{s.t.} \quad (A + A_\delta)x \geq b + b_\delta, \tag{3.22b}$$
$$x_i \in [2, M_i] \cap \mathbb{P}, \quad i = 1, \ldots, n. \tag{3.22c}$$

In the absence of constraint (3.22c), the work in Dawande and Hooker (2000) provides a sufficient condition ensuring $z^* - \Delta \leq z^*_\delta$, where $\Delta \geq 0$, by examining all the leaf nodes of the B&B tree for model (3.14). At a leaf node p of the B&B tree, we let $\bar{x}^p$ and $\underline{x}^p$ denote upper and lower bounds for x, and λ^p denote the optimal value of the dual variables of constraint (3.14b). Further, we let z^{UB}_p denote the best upper bound on the optimal objective function value obtained up to node p in the tree; if no such bound is available, we set $z^{UB}_p = \infty$. We distinguish three cases for the three types of leaf nodes.

(i) If the leaf node is pruned by infeasibility, we let $q^p = \lambda^p A$, $q^p_\delta = \lambda^p A_\delta$, and $z_p = \varepsilon$, where ε is a small positive quantity.

(ii) If the leaf node is pruned by bound, we let $q^p = \lambda^p A - c$, $q^p_\delta = \lambda^p A_\delta - c_\delta$, and $z_p = z^{UB}_p - \Delta$.

(iii) If the leaf node is pruned by optimality, we let $q^p = \lambda^p A - c$, $q^p_\delta = \lambda^p A_\delta - c_\delta$, and $z_p = \bar{z}_p - \Delta$ where $\bar{z}_p$ is the objective function value at that node.

Then, the following theorem relates the objective functions of model (3.14) and its perturbation in model (3.22).

Theorem 3.1. *Consider model* (3.14) *and its perturbation in model* (3.22). *Let* q_p, q^p_δ, z_p, *and* λ^p *be as defined above. For a given* $\Delta \geq 0$, *the bound* $z^* - \Delta \leq z^*_\delta$ *holds if there exist* $\bar{q}^p_j, j = 1, \ldots, n$ *satisfying*

$$\sum_{j=1}^{n} \left((q^p_j + q^p_{\delta j})\underline{x}^p_j + \bar{q}^p_j (\bar{x}^p_j - \underline{x}^p_j) \right) \leq \lambda^p (b + b_\delta) - z_p, \tag{3.23a}$$
$$\bar{q}^p_j \geq q^p_j + q^p_{\delta j}, \ \bar{q}^p_j \geq 0, \quad j = 1, \ldots, n, \tag{3.23b}$$

for each leaf node p *of the branch-and-bound tree of model* (3.14).

Proof. See Dawande and Hooker (2000) and Appendix A.2. □

The proof of Theorem 3.1 mirrors that in Dawande and Hooker (2000) for mixed-integer programs (i.e., in the absence of prime number constraints); hence, we reserve it for the appendix. Although Theorem 3.1 and the results in Dawande and Hooker (2000) provide only a sufficient, and not necessary, condition, there is value in such sensitivity analysis as we demonstrate next. In Problem 7, we consider an illustrative example inspired by the Goldbach conjecture.

Problem 7. Find the smallest prime number, x_1, satisfying the condition $2k + 2 = x_1 + x_2$ for a given $k \in \mathbb{Z}^+$ where $x_2 \in \mathbb{P}$.

Solution. The following PP formulates Problem 7.

$$\begin{aligned} z^* = \min_{x_1,x_2} \quad & x_1 && \text{(3.24a)}\\ \text{s.t} \quad & x_1 + x_2 \geq 2k + 2, && \text{(3.24b)}\\ & -x_1 - x_2 \geq -2k - 2, && \text{(3.24c)}\\ & x_1, x_2 \in [2, M] \cap \mathbb{P}. && \text{(3.24d)} \end{aligned}$$

where M is a large enough upper bound. □

Now consider a perturbation in Problem 7: find the smallest prime number, x_1, satisfying the condition $2k + 2 + k_\delta = x_1 + x_2$ for given k where k_δ is an even integer and $x_2 \in \mathbb{P}$. In other words, we change the right hand sides of equations (3.24b) and (3.24c). Then, we need to solve the following PP:

$$\begin{aligned} z^*_\delta = \min_{x_1,x_2} \quad & x_1 && \text{(3.25a)}\\ \text{s.t} \quad & x_1 + x_2 \geq 2k + 2 + k_\delta, && \text{(3.25b)}\\ & -x_1 - x_2 \geq -2k - 2 - k_\delta, && \text{(3.25c)}\\ & x_1, x_2 \in [2, M] \cap \mathbb{P}. && \text{(3.25d)} \end{aligned}$$

Following the notation of model (3.22), in model (3.25) we have $A_\delta \leftarrow 0, c_\delta \leftarrow 0$, and $b_\delta = [k_\delta, -k_\delta]$. For a numerical demonstration, consider $k = 1699$. Figure 3.2 presents the B&B tree for this instance of model (7); we have, $z^* = 11$. The enumeration tree contains five nodes, three of which are leaf nodes (marked in white).

(i) Consider node 3. We have $\underline{x}^3 = [2, 2], \bar{x}^3 = [7, 3391], z_3 = \varepsilon, \lambda^3 = [0, 1]$. Then, $q^3 = [1, 1], q^3_\delta = [0, 0]$. In order to satisfy the sufficient condition of Theorem 3.1 at this node, we require a solution $\bar{q}^3_1, \bar{q}^3_2$ such that $4 + 5\bar{q}^3_1 + 3389\bar{q}^3_2 \leq 3400 + k_\delta - \varepsilon$ and $\bar{q}^3_1, \bar{q}^3_2 \geq 1$. By setting $\bar{q}^3_1, \bar{q}^3_2 = [1, 1]$ we obtain $k_\delta \geq -2 + \varepsilon$.

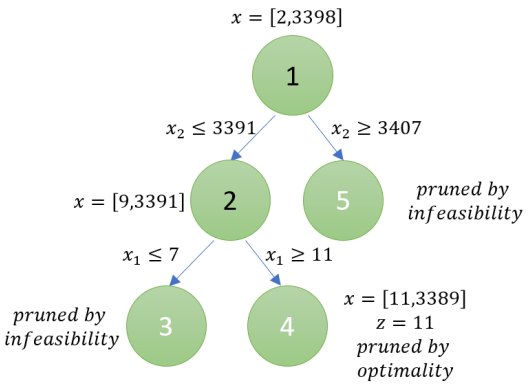


FIGURE 3.2: Enumeration tree for the prime B&B procedure applied to model (3.24) with $k = 1699$. The three nodes marked in white are leaf nodes that are pruned below for reasons indicated, while the other two nodes are further branched on. For details, see Section 3.3.3.

(ii) Consider node 4. We have $\underline{x}^4 = [11, 2], \bar{x}^4 = [M, 3391], z_4 = 11 - \Delta, \lambda^4 = [0, 0]$. Then, $q^4 = [-1, 0], q^4_\delta = [0, 0]$. In order to satisfy the sufficient condition of Theorem 3.1 at this node, we require a solution $\bar{q}^4_1, \bar{q}^4_2$ such that $-11 + (M - 11)\bar{q}^4_1 + 3389\bar{q}^4_2 \leq -11 + \Delta$ and $\bar{q}^4_1, \bar{q}^4_2 \geq 0$. By setting $\bar{q}^4_1, \bar{q}^4_2 = [0, 0]$ we obtain $\Delta \geq 0$.

(iii) Consider node 5. We have $\underline{x}^5 = [2, 3407], \bar{x}^5 = [M, M], z_5 = \varepsilon, \lambda^5 = [1, 0]$. Then $q^5 = [-1, -1], q^5_\delta = [0, 0]$. In order to satisfy the sufficient condition of Theorem 3.1 at this node, we require a solution $\bar{q}^5_1, \bar{q}^5_2$ such that $-3409 + (M - 2)\bar{q}^5_1 + (M - 3407)\bar{q}^5_2 \leq -3400 - k_\delta - \varepsilon$ and $\bar{q}^5_1, \bar{q}^5_2 \geq 0$. By setting $\bar{q}^5_1, \bar{q}^5_2 = [0, 0]$ we obtain $k_\delta \leq 9 - \varepsilon$.

Thus, we conclude that for any $\Delta \geq 0$, $z^*_\delta \geq 11 - \Delta$ holds if $k_\delta \in (-2, 9)$. The optimal solution of model (3.24) is $x^* = [11, 3389]$. We now illustrate the value of sensitivity analysis for PP via the following arguments.

Since k_δ is even, the set of small-enough allowed values is $\{2, 4, 6, 8\}$. Now, consider, $k_\delta = 2$ and the corresponding optimization model: $\min_x x_1$ s.t.$\{x_1 + x_2 = 3402$ with $x_1, x_2 \in \mathbb{P}\}$. Instead of re-solving this model, we simply check the feasibility of the optimal solution of model (3.24) for model (3.25). From the condition, $x^*_1 \geq 11 - \Delta, \forall \Delta \geq 0$ it is sufficient to check feasible values of x_1 that are at least 11. Then, since $[x_1, x_2] = [11, 3391]$ is feasible for model (3.25), it is also optimal; hence, the optimal objective function value is 11. We thus obtain the optimal solution without any additional computation (except checking that 3391 is a prime number). In contrast, the B&B tree for this instance of model (3.25) has three nodes. Next, consider $k_\delta = 4$ and the corresponding optimization model: $\min_x x_1$ s.t.$\{x_1 + x_2 = 3404$ with $x_1, x_2 \in \mathbb{P}\}$. Then, $[x_1, x_2] = [11, 3393]$ is infeasible. The next solution $[x_1, x_2] = [13, 3391]$ is feasible; hence, the optimal objective function value is 13. We

obtain this solution in two evaluations. In contrast, the B&B tree for this instance of model (3.25) again has three nodes. Similarly, for $k_\delta = 6$ and $k_\delta = 8$, we obtain the optimal objective function value of 17 in only three additional evaluations. Thus, following Theorem 3.1, we save computational effort by not considering prime numbers below 11; i.e., we obviate checking the primes 3, 5, and 7. In Appendix A.3, we provide another problem which is also inspired by the Goldbach conjecture to illustrate the value of Theorem 3.1.

3.4 Case Studies

In this section, we discuss problems where we can reformulate them into the PPs and provide their optimization models, including an application in cryptography and a research problem in number theory to find supporting numerical examples.

3.4.1 Cryptographic Hardware Optimization: FHE Residue Selection

A real-world application of the prime programs is in cryptographic hardware design within the domain of *Fully Homomorphic Encryption (FHE)*. FHE is a cryptographic scheme that enables arbitrary computations on encrypted data without requiring decryption (Gentry, 2009). However, the approach has a significant computational overhead burden. To improve this, modern accelerators implement the *CKKS* scheme (Cheon et al., 2017), using approximate arithmetic to process encrypted vectors. This algorithm relies on the *Residue Number System (RNS)* (Bajard et al., 2017). RNS is a non-positional representation that decomposes large ciphertext coefficients into a vector of smaller, independent residues. This decomposition allows for massive parallelization, as arithmetic operations can be performed on each residue independently without the bottleneck of sequential bitwise dependence.

The selection of the moduli $\{q_1, q_2, \ldots, q_C\}$ in an RNS-based accelerator is an important factor for the efficiency. The *BitPacker* architecture (Samardzic and Sanchez, 2024) introduces a scheme of decoupling the total cryptographic scale T_L from the hardware's physical word size w. This is a selection of a set of primes whose product is at least T_L with the objective of minimizing the cryptographic noise. Beyond bit-width constraints, the selected moduli must support the *Number Theoretic Transform (NTT)* (Harvey, 2014), which is important for accelerating polynomial multiplications. For a polynomial of degree N, each modulus q_i must be a prime satisfying the congruence $q_i \equiv 1 \pmod{2N}$. The challenge of identifying a set of *NTT*-friendly primes that fit hardware registers while minimizing redundant scale can be formulated as a PP. By transforming the product of moduli into a logarithmic

summation, we minimize the overshoot noises relative to the target scale T_L as follows:

$$\min_{q,k} \quad \sum_{i=1}^{C} \log_2(q_i) - \log_2(T_L) \tag{3.26a}$$

$$\text{s.t.} \quad \sum_{i=1}^{C} \log_2(q_i) \geq \log_2(T_L), \tag{3.26b}$$

$$q_i = 2Nk_i + 1, \quad \forall i \in \{1, \ldots, C\}, \tag{3.26c}$$

$$q_i \leq q_{i+1} - 2, \quad \forall i \in \{1, \ldots, C-1\}, \tag{3.26d}$$

$$q_i \leq 2^w - 1, \quad \forall i \in \{1, \ldots, C\}, \tag{3.26e}$$

$$q_i \in \mathbb{P}, \quad \forall i \in \{1, \ldots, C\}, \tag{3.26f}$$

$$k_i \in \mathbb{Z}^+. \tag{3.26g}$$

In this formulation, the objective function (3.26a) minimizes the distance between the RNS dynamic range and the target cryptographic scale. Constraint (3.26b) acts as a safety boundary, ensuring the product of the moduli is sufficient to prevent modular overflow. Constraint (3.26c) ensures NTT compatibility, while hardware register limits are enforced by (3.26e). By defining $r_i = \log_2(q_i)$ for each $i = 1, \ldots, C$, the optimization model becomes a convex PP which is able to solve using the algorithm in Section 3.3.1.

3.4.2 Linear Equations in Primes

We first consider the following conjecture proposed in Pomerance et al. (1988).

Conjecture 1 (Pomerance et al. (1988)). There exist n distinct prime numbers $x_1 < x_2 < \ldots < x_n$ such that the average of any two of these primes, given by $\frac{1}{2}(x_i + x_j), \forall i, j = 1, \ldots, n, i \neq j$, is also a prime number for $n \geq 2$. □

Conjecture 1 is proven in Granville (1990) assuming the so-called k-Tuple Conjecture of Hardy and Littlewood (Hardy and Littlewood, 1923) is true. A solution of this conjecture is not unique for a given n, e.g., consider $n = 3$ then $[3, 7, 19]$ and $[5, 17, 41]$ are both valid solutions. We are only aware of solutions of Conjecture 1 until $n = 12$; for $n = 12$ a solution is provided in Balog (1992): $[5, 17, 521, 42281, 138461, 195137, 204137, 221537, 363497, 367001, 414737, 434717]$.

To solve Conjecture 1, we formulate a sequence of PPs of increasing fidelity. These PPs take as input both n and an upper bound, M, for the considered set of prime numbers.

We begin with a naive PP as follows.

$$\min_{x,y} \quad 0 \tag{3.27a}$$
$$\text{s.t.} \quad 2y_{ij} = x_i + x_j, \qquad 1 \le i < j \le n, \tag{3.27b}$$
$$x_{i+1} \ge x_i + 1, \qquad 1 \le i \le n-1, \tag{3.27c}$$
$$x_i \in [2, M] \cap \mathbb{P}, \qquad 1 \le i \le n, \tag{3.27d}$$
$$y_{ij} \in [2, M] \cap \mathbb{P}, \qquad 1 \le i < j \le n. \tag{3.27e}$$

An optimal solution of model (3.27), if it exists, provides a vector $[x]_n$, with each element below M, that satisfies Conjecture 1. Here, constraints (3.27b) and (3.27e) ensure that the average of any two components of $[x]$ is also prime, where y_{ij} is the average of x_i and x_j for $1 \le i < j \le n$. Constraint (3.27c) ensures that the solutions are distinct primes, while constraint (3.27d) restricts x to be prime. Then, from Section 3.3.1, model (3.27) is a PP.

We now tighten the formulation in model (3.27) with a few observations. First, constraint (3.27b) requires the elements of x to be odd primes; thus, we exclude the prime 2 from the optimal solution of model (3.27). We make two further observations: (i) prime numbers differ by at least 2, and (ii) we require at least one prime number between x_i and x_j. Thus, we tighten constraint (3.27c) and have a more efficient model given by the PP below.

$$\min_{x,y} \quad 0 \tag{3.28a}$$
$$\text{s.t.} \quad 2y_{ij} = x_i + x_j, \qquad 1 \le i < j \le n, \tag{3.28b}$$
$$x_{i+1} \ge x_i + 4, \qquad 1 \le i \le n-1, \tag{3.28c}$$
$$y_{ij} \ge x_i + 2, \qquad 1 \le i < j \le n, \tag{3.28d}$$
$$x_j \ge y_{ij} + 2, \qquad 1 \le i < j \le n, \tag{3.28e}$$
$$x_i \in [3, M] \cap \mathbb{P}, \qquad 1 \le i \le n, \tag{3.28f}$$
$$y_{ij} \in [5, M] \cap \mathbb{P}, \qquad 1 \le i < j \le n. \tag{3.28g}$$

Table 3.1 presents our computational results for model (3.28) with $M = 1000$. We solve model (3.28) with the B&B method presented in Section 3.3.1. Solutions for $n \le 5$ are obtained practically instantly, while for $n = 6, 7, 8$ the solution times are slightly longer. However, no solution is obtained for $n = 9$ within 50 minutes and neither is the model proven infeasible. Increasing the time limit (to allow exploration of a greater number of nodes of the B&B tree) to 100 minutes or increasing M (to allow exploration of a greater number of potentially feasible prime numbers) to 10000 still provides no feasible solution. Each B&B node solves a LP, and this computation is

n	M	Nodes	Time	Solution
2	1000	7	-	[3, 7]
3	1000	31	-	[3, 7, 919]
4	1000	119	-	[3, 7, 19, 139]
5	1000	269	-	[3, 7, 19, 139, 859]
6	1000	189049	90	[3, 11, 23, 71, 191, 443]
7	1000	159473	82	[5, 17, 41, 101, 257, 461, 521]
8	1000	1286503	821	[5, 17, 41, 101, 257, 521, 761, 881]
9	1000	4080930	3000	ϕ
	1000	8209983	6000	ϕ
	10000	3658991	3000	ϕ
	10000	7353380	6000	ϕ

TABLE 3.1: Computational results on solving model (3.28) using the `Naive` B&B solution strategy presented in Section 3.3.1. Here, n and M are inputs to model (3.28), Nodes denotes the number of explored nodes in the B&B tree, and Time denotes the computational time in seconds. Entries of "-" in the time column denote the solution is obtained practically instantaneously. The Solution column presents an optimal solution found with the given parameters, here the entry ϕ in the rows with $n = 9$ denotes that no solution is obtained; increasing the value of M and Time still does not provide a solution.

highly effective benefiting from modern LP solvers; for $n = 6, 7, 8, 9$, the computational time per 1000 nodes is only marginally over half a second. Thus, we now explore tailored strategies to determine the branching variables as well as fixing certain variables to solve model (3.28).

3.5 Solution Strategies

In Section 3.3.1, we present a B&B strategy to solve prime programs. This B&B procedure is generic and capable of solving any PP; however, the computational effort could be prohibitive. Similar to LP, certain prime programs allow us to exploit their structural properties to obtain solutions in a computationally efficient way. In this section, we exploit such properties of the presented B&B procedure with the problems discussed in the previous section as a numerical case study.

We carry out all computational experiments on a computing cluster with an Intel Xeon Gold 6138 2.0GHz processor with 192 GB of RAM and Gurobi version 10.0.3 using Python 3.8.3. We require only an LP solver, and use Gurobi's default LP optimality tolerance of 10^{-6}. We provide the corresponding B&B code at our GitHub page. As we demonstrate in the previous section, solving the problem we consider —given an upper bound of the considered feasible solution—via the B&B method of Section 3.3.1 is computationally challenging. We refer to this branching strategy as the `Naive` strategy, and begin with variable fixing strategies.

3.5.1 Variable-fixing Strategies

Computing a solution for model (3.28) becomes increasingly computationally difficult as n increases; however, the solution $[x]_n$ includes several components from the solution $[x]_{n-1}$. This observation motivates a variable fixing strategy where the first $n-1$ components of $[x]_n$ are fixed from the previously obtained solution. We then solve model (3.28) for the largest missing element alone. We denote such a strategy of fixing the decision variables as `SelectAll`.

Routine 1 `fixing strategy` (name, arg)

Input: *name* $= \{$`Naive`, `SelectAll`, `ExcludeOne`, `ExcludeTwo`$\}$ of chosen fixing strategy; a vector $[\bar{x}]$ of dimension $n-1$, additional arguments arg of appropriate dimension. Below, $a \leftarrow \phi$ denotes no value is assigned to a.
Output: a vector $[x]$ of dimension n.

1: **if** name $=$ `Naive` **then**
2: $x_i \leftarrow \phi, \forall i = 1, \ldots, n$.
3: **if** name $=$ `SelectAll` **then**
4: $x_i \leftarrow \bar{x}_i, i = 1, \ldots, n-1$; $x_n \leftarrow \phi$.
5: **if** name $=$ `ExcludeOne` **then**
6: $(l) \leftarrow$ arg.
7: $x_i \leftarrow \bar{x}_i, i = 1, \ldots, l-1$; $x_i \leftarrow \bar{x}_{i+1}, i = l, \ldots, n-2$; $x_i \leftarrow \phi, i = n-1, n$.
8: **if** name $=$ `ExcludeTwo` **then**
9: $(l_1, l_2) \leftarrow$ arg, with $l_1 < l_2$.
10: $x_i \leftarrow \bar{x}_i, i = 1, \ldots, l_1 - 1$; $x_i \leftarrow \bar{x}_{i+1}, i = l_1, \ldots, l_2 - 2$; $x_i \leftarrow \bar{x}_{i+2}, i = l_2 - 1, \ldots, n-3$; $x_i \leftarrow \phi, i = n-2, \ldots, n$.

The `SelectAll` strategy might result in infeasibility of model (3.28) due to too many variables being fixed. Thus, rather than fixing all the first $n-1$ components of x, we selectively exclude some elements from fixing, e.g., those near their upper bound. To this end, we define two other strategies — `ExcludeOne` and `ExcludeTwo` — that exclude one and two elements from fixing while keeping the two and three largest elements unassigned, respectively. Routine 1 summarizes the three variable fixing strategies, while Example 3.1 illustrates them numerically; here, the quantity arg denotes the input arguments to the respective strategies. In Section 3.5.3, we provide the benefit of these three enhanced strategies over the `Naive` strategy.

Example 3.1. *An illustration of the four variable fixing strategies of Routine 1.*

Let $n = 7$ and consider a known solution, $\bar{x}$, for $n = 6$ with $\bar{x} = [7, 19, 67, 127, 547, 607]$. From the seven positions of x to determine, we fix at most six positions from the known solution $\bar{x}$. In the `Naive` strategy we do not fix any variables. Our remaining three strategies are as follows.

(i) In the `SelectAll` strategy, all six elements are fixed. Then, we have $x = [7, 19, 67, 127, 547, 607, \phi]$.

(ii) In the `ExcludeOne` strategy, we have one element, l, excluded as determined by its input argument while the other five elements are fixed. Consider $l = 3$. Then, we have $x = [7, 19, 127, 547, 607, \phi, \phi]$.

(iii) In the `ExcludeTwo` strategy, we have two elements, l_1 and l_2, that are excluded as determined by its input argument, while the other four elements are fixed. Consider $(l_1, l_2) = (3, 4)$. Then, we have $x = [7, 19, 547, 607, \phi, \phi, \phi]$.

□

3.5.2 Branching Strategies

By examining the solutions in Table 3.1, we observe that the solutions for $n = 2$ to 6 are similar; this observation motivated the variable fixing strategies of Section 3.5.1. However, none of the elements of the solutions from $n = 2$ to 6 are present in the solutions for $n = 7$ and $n = 8$. In this section, we present a different strategy motivated by number theory. Unlike integer programming, prime numbers are not spaced equally. We thus investigate strategies related to the remainder of an integer when divided by another integer. In this section, we consider prime numbers that are at least 5; checking whether 3 forms part of the solution is easily done as a post-processing step to the optimization. We begin with two simple lemmas.

Lemma 3.2. *Any prime number $x \geq 5$ is expressible as $x = 12z + k$ for some $z \in \mathbb{Z}$ and some $k \in \{1, 5, 7, 11\}$.*

Proof. Any integer x is expressible as $x = 12z + k$ for some $z \in \mathbb{Z}$, where $0 \leq k \leq 11$ is the remainder of x when divided by 12. When $k = 0$, $x = 12z$, which is divisible by 12; thus, x is not prime. Similarly, if $k = 2, 3, 4, 6, 8, 9, 10$, then x is divisible by $2, 3, 4, 6, 4, 3, 2$, respectively; hence, it is not prime. Therefore, for a given prime $x \geq 5$, it is expressible only in the form $x = 12z + k$ for some $z \in \mathbb{Z}$ and some $k \in \{1, 5, 7, 11\}$. For example, the prime number $23 = 12 \cdot 1 + 11$. □

Lemma 3.3. *In an optimal solution for model* (3.28)*, the prime number y_{ij} is given by the average of two prime numbers, x_i and x_j, both of which leave the same remainder when divided by 12. Hence, y is of the form $6z + k$ for some $z \in \mathbb{Z}$, and some $k \in \{1, 5\}$.*

Proof. From Lemma 3.2, consider $x_1 = 12z_1 + k_1$ and $x_2 = 12z_2 + k_2$ for some $z_1, z_2 \in \mathbb{Z}$ and some $k_1, k_2 \in \{1, 5, 7, 11\}$, and $x_1, x_2 \geq 5$. Table 3.2 provides the average of x_i and x_j for different values of k_1 and k_2. The average cannot be prime unless $k_1 = k_2$. This completes the first part of the lemma. Then, from constraint (3.28) (equivalently, from the diagonal entries of Table 3.2), y_{ij} is of the form $y_{ij} = 6z_{ij} + k$ for some $z_{ij} \in \mathbb{Z}$ where $k \in \{1, 5\}$. □

Routine 2 `branching strategy` (name, LP solution at node, arg)

Input: *name* ={`Naive`, `Modulo`} of chosen branching strategy; arguments arg of the chosen branching strategy, arg $= \phi$ if *name* = `Naive`, arg $= \{1, 5, 7, 11\}$ if *name* = `Modulo`; $(x^*, y^*) \leftarrow$ a solution of the linear programming relaxation of model (3.28) at a given node.

Output: constraints for right and left child nodes of the input node.

if name = `Naive` **then**
  for $i = 1, \ldots, n$ **do**
    if $x_i^* \notin \mathbb{P}$ **then**
      $distX_i \leftarrow \min\{\bar{p}_{x_i^*} - x_i^*, x_i^* - \underline{p}_{x_i^*}\}$.
    else
      $distX_i \leftarrow 0$.
    for $j = i + 1, \ldots, n$ **do**
      if $y_{ij}^* \notin \mathbb{P}$ **then**
        $distY_{ij} \leftarrow \min\{\bar{p}_{y_{ij}^*} - y_{ij}^*, y_{ij}^* - \underline{p}_{y_{ij}^*}\}$.
      else
        $distY_{ij} \leftarrow 0$.
  $indX \leftarrow \text{argmax}_i\, distX_i$; $indY \leftarrow \text{argmax}_{ij}\, distY_{ij}$.
  if $distX_{indX} > distY_{indY}$ **then**
    return right and left nodes that contain additional constraints $x_{indX} \geq \bar{p}_{x_{indX}^*}$ and $x_{indX} \leq \underline{p}_{x_{indX}^*}$, respectively.
  else
    return right and left nodes that contain additional constraints $y_{indY} \geq \bar{p}_{y_{indY}^*}$ and $y_{indY} \leq \underline{p}_{y_{indY}^*}$, respectively.
if name = `Modulo` **then**
  $k \leftarrow$ arg.
  $k_0 \leftarrow k \mod 6$.
  for $i = 1, \ldots, n$ **do**
    if $(x_i^* \notin \mathbb{P})$ or $(x_i^* \mod 12 \neq k)$ **then**
      $distX_i \leftarrow \min\{\bar{p}_{x_i^*}(12, k) - x_i^*, x_i^* - \underline{p}_{x_i^*}(12, k)\}$.
    else
      $distX_i \leftarrow 0$.
    for $j = i + 1, \ldots, n$ **do**
      if $(y_{ij}^* \notin \mathbb{P})$ or $(y_{ij}^* \mod 6 \neq k_0)$ **then**
        $distY_{ij} \leftarrow \min\{\bar{p}_{y_{ij}^*}(6, k_0) - y_{ij}^*, y_{ij}^* - \underline{p}_{y_{ij}^*}(6, k_0)\}$.
      else
        $distY_{ij} \leftarrow 0$.
  $indX \leftarrow \text{argmax}_i\, distX_i$; $indY \leftarrow \text{argmax}_{ij}\, distY_{ij}$.
  if $distX_{indX} > distY_{indY}$ **then**
    return right and left nodes that contain additional constraints $x_{indX} \geq \bar{p}_{x_{indX}^*}(12, k)$ and $x_{indX} \leq \underline{p}_{x_{indX}^*}(12, k)$, respectively.
  else
    return right and left nodes that contain additional constraints $y_{indY} \geq \bar{p}_{y_{indY}^*}(6, k_0)$ and $y_{indY} \leq \underline{p}_{y_{indY}^*}(6, k_0)$, respectively.

x_1 \ x_2	$12z_2+1$	$12z_2+5$	$12z_2+7$	$12z_2+11$
$12z_1+1$	$6(z_1+z_2)+1$	$3(2z_1+2z_2+1)$	$2(3z_1+3z_2+2)$	$6(z_1+z_2)$
$12z_1+5$	$3(2z_1+2z_2+1)$	$6(z_1+z_2)+5$	$6(z_1+z_2+1)$	$2(3z_1+3z_2+4)$
$12z_1+7$	$2(3z_1+3z_2+2)$	$6(z_1+z_2+1)$	$6(z_1+z_2+1)+1$	$3(2z_1+2z_2+3)$
$12z_1+11$	$6(z_1+z_2)$	$2(3z_1+3z_2+4)$	$3(2z_1+2z_2+3)$	$6(z_1+z_2+1)+5$

TABLE 3.2: Averages of pairs of prime numbers of the form $x_1 = 12z_1 + k$ and $x_2 = 12z_2 + k$ for $k = \{1, 5, 7, 11\}$. For details, see Section 3.5.2 and Lemma 3.3.

Lemmas 3.2 and 3.3 allow an improvement in the branching behavior of model (3.28). A straightforward way of doing so is to add additional integer variables to model (3.28). However, instead we change the way the B&B method constructs the disjunctive branches. Consider $x = 24.7$ at some node of the B&B tree of the `Naive` strategy. Then, the two branches are $x \leq 23$ and $x \geq 29$. Now, let $\bar{p}_x(m,k)$ denote the smallest prime number of the form $mz + k$ for some $z \in \mathbb{Z}$ which is greater than $x \in \mathbb{R}^+$, and $\underline{p}_x(m,k)$ denote the largest prime number of the form $mz + k$ for some $z \in \mathbb{Z}$ which is less than $x \in \mathbb{R}^+$. We add the additional constraint for the two branching nodes as $x_i \geq \bar{p}_{x_i^*}(m,k)$ and $x_i \leq \underline{p}_{x_i^*}(m,k)$, respectively, for $m = 12$ and $k = 1$ (from Lemma 3.3). Then, the two branches under this new branching rule are $x \leq \underline{p}_{24.7}(12,1) = 13$ and $x \geq \bar{p}_{24.7}(12,1) = 37$. With the `Naive` strategy, we need to introduce a new integer variable $z \in \mathbb{Z}$ such that $x = 12z + 1$ and then branch on one variable at a time. Branching on the variable $x = 24.7$ leads to the two branches: $x \leq 23$ and $x \geq 29$. In contrast, branching on the variable $z = 1.975$ leads to the two branches: $z \leq 1$ and $z \geq 2$ which are equivalent to $x \leq 13$ and $x \geq 25$. The advantages of changing the branching method are twofold. First, it provides stronger constraints than those obtained from naive branching method since the solution is always a prime of the form we are considering. Second, it is not necessary to introduce additional variables into the model. We call such a branching strategy as `Modulo` with the argument arg denoting the value of $k \in \{1, 5, 7, 11\}$. Routine 2 summarizes the `Modulo` branching strategies.

3.5.3 Computation Results

In this section, we present our computational experiments to solve model (3.28) using the fixing strategy described in Routine 1 and the branching strategy described in Routine 2. However, rather than directly use them as solution procedures, we begin with a need for an iterative algorithm for solving model (3.28) that takes an input both these strategies. For example, consider a moderate sized instance of model (3.28) with $n = 5$ and $M = 3000$ that we solve with the `ExcludeTwo` fixing strategy and the `Modulo` branching strategy. From the total of six instances each for arg $= 1, 5, 7$, and 11, we obtain feasible solutions in only two, one, two, and three instances, respectively; the

remaining instances are all infeasible. However, increasing the upper bounds results in all instances being feasible within two seconds. Similarly, there are easily identifiable instances where neither feasibility nor infeasibility is proved within a given time limit; Example A.1 in Appendix A.4 provides a numerical instance that shows the number of such instances that fail to obtain a feasible solution or proven infeasibility within a given time limit is reduced when the time limit is increased. Further, in Example A.2 and A.3 of Appendix A.4 we demonstrate the computational benefits of the variable fixing and branching strategies over their naive counterparts. Since we know model (3.28) is feasible until at least $n = 12$ given a sufficiently large time limit and upper bounds, we initialize solution procedures with a small enough upper bound and iteratively increase the bound if an instance is proven infeasible. Algorithm 3 presents the complete scheme we employ to solve model (3.28). We illustrate the working of the algorithm in Example 3.2. In the interest of space, we present only the most relevant computational results here while we reserve the complete set of results for Appendix A.4.

Algorithm 3 An algorithm to solve the PP given by model (3.28).

Input: an instance of model (3.28) given by n and $M_i, \forall i = 1, \ldots, n$; time limits T and t; a scalar α; subroutines `fixing strategy` defined by Routine 1 and `branching strategy` defined by Routine 2.

Output: a feasible solution of model (3.28), $x_i^*, i = 1, \ldots, n$ or a statement of intractability under the given input parameters.

$k \leftarrow 0, m \leftarrow 1, time \leftarrow 0, [x^0]_0 = []$.
while $m \leq n$ **do**
  Fix $[x^k]_m \leftarrow$ `fixing strategy`$(name, [x^k]_{m-1})$.
  Solve instance of model (3.28) with input m and M_i for $i = 1, \ldots, m$ using `branching strategy` up to at most t seconds.
  $k \leftarrow k + 1$.
  Update *time* to wall-clock time.
  if feasible solution found **then**
    $[x^k]_m \leftarrow$ feasible solution of model (3.28).
    if $m = n$ **then**
      return $[x^*]_m \leftarrow [x^k]_m$.
    $m \leftarrow m + 1$.
  else
    $[x^k]_{m-1} \leftarrow [x^{k-1}]_{m-1}$.
    $M_i \leftarrow M_i(1 + \alpha), \forall i = 1, \ldots, m$.
  if $time \geq T$ **then**
    return statement of intractability.

Example 3.2. *Consider an instance of model* (3.28) *with inputs* $n = 3$, $M = 1000$, $T = 600$ *seconds,* $t = 300$ *seconds,* $\alpha = 0.2$, `fixing strategy` $=$ `SelectAll` *with* arg $= \phi$, *and* `branching strategy` $=$ `Modulo` *with* arg $= 1$ *applied to each node in Algorithm 3.*

We initialize Algorithm 3 with $k \leftarrow 0$, $m \leftarrow 1$, $time \leftarrow 0$, and $[x^0]_0 \leftarrow []$. In the first iteration, we apply the `fixing strategy` $=$ `SelectAll`, resulting in $[x^0]_1 \leftarrow [\phi]$.

Using this fixed solution, we solve model (3.28) with the `Modulo` with arg $= 1$ branching strategy to obtain a feasible solution in Step 4. Then, $k \leftarrow 1$ and *time* is updated; currently, the fixed solution is $[x^1]_1 = [13]$ and $m \leftarrow 2$. In the second iteration, we apply the `fixing strategy` again, now yielding $[x^1]_2 = [13, \phi]$. Solving the model, we obtain a solution of $[13, 853]$. After this, $k \leftarrow 2$ and *time* is updated, leading to $[x^2]_2 = [13, 853]$ and $m \leftarrow 3$. In the third iteration, applying the `fixing strategy` results in $[x^2]_3 = [13, 853, \phi]$. However, now the model becomes infeasible; $k \leftarrow 3$ and *time* is updated. Hence, we retain the previous fixed solution $[x^3]_2 \leftarrow [x^2]_2 = [13, 853]$, and the parameter M is updated to 1200 for $i = 1, 2, 3$. In the fourth iteration, the `fixing strategy` yields $[x^3]_3 \leftarrow [13, 853, \phi]$. Solving the model, we obtain the final solution $[13, 853, 1129]$. Finally, $k \leftarrow 4$, *time* is updated, and the algorithm terminates, returning the feasible solution $[x] = [13, 853, 1129]$.

□

Now, we compare the four fixing and five branching strategies to each other. For a given n and each of the branching strategies, we report results for each instance of the four fixing strategies; i.e., one for `Naive`, one for `SelectAll`, $n - 1$ for `ExcludeOne` and $\frac{(n-1)(n-2)}{2}$ for `ExcludeTwo`. Thus, for the five branching strategies we report solutions and the time taken to obtain these solutions for $5 \cdot (2 + \frac{n(n-1)}{2})$ instances for each n; for instances that fail to obtain a feasible solution or proven infeasibility within a given time limit, we report as "Intractable". Tables A.1-A.6 of Appendix A.4 present all of these results. Table 3.3 presents a summary of the "best" branching strategy (i.e., where a solution is obtained in the least amount of time) results for each n for each of the four fixing strategies; the corresponding best fixing strategy is always `SelectAll` (we discuss this below).

Among the fixing strategies, the `SelectAll` strategy stands out as the most effective by successfully obtaining a feasible solution in each of the 30 instances, for any branching strategy, and all $n = 5, \ldots, 10$; see, Table 3.4b. On the other hand, the `ExcludeOne` and `ExcludeTwo` strategies encounter a significant number of intractable instances — nearly two-fifths and three-fifths of the instances for `ExcludeOne` and `ExcludeTwo` are intractable (see, Table 3.4b). Both these strategies are also poor at scaling with increasing n: the proportion of intractable instances increases from 3% and 4% for $n = 7$ to 93% and 100% for $n = 10$ for `ExcludeOne` and `ExcludeTwo`, respectively.

Further, the `SelectAll` strategy consistently produces the best solutions among the four fixing strategies in 29 of its 30 instances. The only exception occurs for the case of $n = 6$, where the `Naive` and `ExcludeOne` fixing strategies obtain a solution instantly for the `Modulo` with arg $= 5$ branching strategy; however, here too the `SelectAll` strategy is worse by only one second. In contrast, the `ExcludeOne` and `ExcludeTwo` strategies yield the best solutions in a limited number of instances. Specifically, for $n = 5, \ldots, 10$, `ExcludeOne` produces the best solutions in 20/20, 3/25, 0/30, 0/35, 0/40, and 0/45

n	`fixing strategy`	`branching strategy`	Time
6	`Naive`	`Modulo` 5; 7; 11	-
	`SelectAll`	`Modulo` 1; 7; 11	-
	`ExcludeOne`	`Modulo` 1; 5; 11	-
	`ExcludeTwo`	`Modulo` 1; 11	-
7	`Naive`	`Modulo` 5	2
	`SelectAll`	`Modulo` 7; 11	-
	`ExcludeOne`	`Modulo` 7; 11	1
	`ExcludeTwo`	`Modulo` 11	1
8	`Naive`	`Modulo` 5	56
	`SelectAll`	`Modulo` 1; 7	1
	`ExcludeOne`	`Modulo` 7	8
	`ExcludeTwo`	`Modulo` 1	30
9	`Naive`	$\times$	3000
	`SelectAll`	`Modulo` 1; 7; 11	2
	`ExcludeOne`	`Modulo` 7	11
	`ExcludeTwo`	`Modulo` 11	16
10	`Naive`	$\times$	6000
	`SelectAll`	`Modulo` 1	18
	`ExcludeOne`	`Modulo` 1	1147
	`ExcludeTwo`	$\times$	6000
11	`Naive`	$\times$	6000
	`SelectAll`	`Modulo` 7	664
	`ExcludeOne`	$\times$	6000
	`ExcludeTwo`	$\times$	6000

TABLE 3.3: Selected results on solving model (3.28) with Algorithm 3 that obtain feasible solutions in the least computational time for each `fixing strategy` in Routine 2 for $n = 6$ to 11; i.e., the best performers for each fixing strategy. The Time column shows the computational time in seconds. Entries of "-" denote the solution is obtained practically instantaneously while entries of $\times$ denote no solution is obtained in the given time limit. The `Naive` branching strategy is consistently worse than the `Modulo` branching strategy. The `SelectAll` fixing strategy is consistently the best performer among the fixing strategies for any given branching strategies.

instances, respectively, while `ExcludeTwo` produces the best solutions in 30/30, 6/50, 1/75, 0/105, 0/140, and 0/180 instances, respectively (see, Tables A.1-A.6 of Appendix A.4). For $n = 5$, all instances are solved practically instantly.

Among the branching strategies, the `Modulo` with arg $= 7$ and 11 strategies both achieve the best performance in 9/24 instances; see, Table 3.3. This is followed by the `Modulo` with arg $= 1$ and `Modulo` with arg $= 5$ strategies which perform the best in 8/24 and 4/24 instances, respectively. In contrast, none of the instances with the `Naive` strategy achieve the lowest computation time for any n; see, Table 3.3. Among the `Modulo` strategies, for $n = 6, \ldots, 9$, there is no clear consensus for the best performer. For $n = 10$ and $n = 11$, five instances are intractable for any branching strategy; however, the `Modulo` with arg $= 1$ and 7 strategies consistently dominate the performance for these n.

	Instances	Feasible	Avg. Time
`Naive`	167	33.5%	352
`Modulo` 1	167	53.9 %	290
`Modulo` 5	167	47.9 %	442
`Modulo` 7	167	52.1 %	377
`Modulo` 11	167	50.3 %	274

(A) `branching strategy`

	Instances	Feasible	Avg. Time
`Naive`	30	63.3 %	374
`SelectAll`	30	100.0 %	30
`ExcludeOne`	195	59.5 %	261
`ExcludeTwo`	580	40.0 %	393

(B) `fixing strategy`

TABLE 3.4: Summary of computational results on solving model (3.28) with Algorithm 3 for $n = 5, \ldots, 10$ for each (a) `branching strategy` and (b) `fixing strategy`. Here, Instances denotes the number of instances, Feasible denotes the percentage of instances that obtain a feasible solution in the given time limit, and Avg. Time denotes the average computational time of these feasible instances. For details, see Section 3.5.3 and Appendix A.4.

Table 3.5 presents the averaged computational times, separately for each n, across instances where a feasible solution is obtained by Algorithm 3 (i.e., not intractable). We find that the `Modulo` strategy—with any of the four arguments—is almost an order of magnitude faster, on average, than the `Naive` strategy for small n. For example, consider $n = 5, 6, 7$: the slowest of the four arguments of the `Modulo` strategy (with any argument) is faster than the `Naive` in $12/12, 17/17, 16/23$ instances (there are no intractable instances in this case). It might appear that the `Naive` strategy is better for large n due to the lower computational times in Table 3.5. However, this masks the fact that there are only few instances where a feasible solution is obtained by the `Naive` strategy; see, Table 3.3. Indeed, for $n = 8, 9, 10$, the best `Modulo` strategy is faster than the `Naive` strategy in $27/30$, $38/38$, $47/47$ of the instances, respectively. The `Naive` strategy manages to obtain feasible solutions in only $23/30$, $1/38$, and $1/47$ instances for $n = 8$, 9, and 10, respectively; i.e., most of the nearly one-third (see, Table 3.4) feasible instances of the Naive strategy are from small values of n. With this background, the `Modulo` strategy outperforms the `Naive` strategy.

Next, we compare the four `Modulo` strategies to each other. Here, there is no clear consensus for the best performer. To see this, first consider all instances for $n = 5, \ldots, 10$. Then, the `Modulo` with arg $= 1$ strategy reports the fewest intractable instances: $77/167$. In contrast, the `Modulo` with arg $= 5$, 7, and 11 strategies report higher numbers of intractable solutions: $87/167$, $80/167$, and $83/167$, respectively. For $n = 8$, only $4/30$ instances are intractable for `Modulo` with arg $= 1$; while the next best performer, `Modulo` with arg $= 7$, has $6/30$ intractable instances. We find that arg $= 5$

	$n = 6$	$n = 7$	$n = 8$	$n = 9$	$n = 10$
`Naive`	127	779	356	113	156
`Modulo` 1	5	65	571	737	1186
`Modulo` 5	4	119	1082	1386	299
`Modulo` 7	5	158	857	827	221
`Modulo` 11	2	112	656	1596	1343

TABLE 3.5: Average computation time of the feasible instances alone on solving model (3.28) with Algorithm 3 for $n = 6, \ldots, 10$ for each `branching strategy`. Solutions for $n = 5$ are obtained practically instantaneously.

performs relatively poorly. It takes the most computational effort (among the four `Modulo` strategies): for $n = 8, 9, 10$, its average computational time (including the time taken for intractable instances) is the worst among the four `Modulo` strategies. Further, arg $= 5$ does not outperform the other three strategies for any instance for $n = 8, 9, 10$. This pattern is particularly visible for $n = 10$ where most of the instances are intractable: the best instances of both `Modulo` with arg $= 1$ and arg $= 11$ obtain feasible solutions in under half a minute while the best for arg $= 5$ and arg $= 7$ take about four to five minutes (see, Table A.6 in Appendix A.4). However, the under-performing `Modulo` with arg $= 5$ strategy becomes useful for $n = 11$: arg $= 5$ and arg $= 7$ are the only two strategies that obtain a feasible solution for $n = 11$ (results not shown). These varied results provide empirical evidence in support of having multiple strategies as heuristics to solve PPs.

3.6 Conclusion

Employing ideas from number theory and mathematical optimization, we introduce a novel conceptual framework of optimizing over prime numbers. We describe mathematical programming formulations for several well-known problems in number theory that are traditionally studied using disciplines other than optimization. The framework we propose then facilitates numerical solutions by easily employing commercially available solvers; we do so by extending the B&B algorithm of LP into prime programming. Such an algorithm is generic and capable of solving any linear prime program.

We then further extend the branching rules of this method by developing additional strategies for variables to branch on as well as fixing certain variables. Our computational results, conducted on a classical conjecture to find a sequence of primes where the average of any two numbers is also a prime, shows the advantage of such methods. Future work could explore priority-based branching methods for optimization problems with integer and prime number constraints. In such problems, prioritizing branching on prime numbers could be advantageous. This preference

stems from the fact that when a feasible solution satisfies the prime number restrictions, the branching process results in a larger reduction of the feasible region compared to that by branching on integer variables alone. Finally, by employing the notion of an inference dual, we extend the concept of sensitivity analysis which is well-studied in linear programming into that for prime programming.

All our data, models, and code are available at:
https://github.com/montreeklim/PrimeNumberProgramming.

Chapter 4

Iterative Polyhedron Approximation to k-Hyperplane Clustering Problem

In this chapter, we consider the k-hyperplane clustering (k-HC) problem, which features a different constraint structure than the prime programming problems discussed in the previous chapter. This problem partitions data into k clusters, each represented by a hyperplane, by minimizing the sum of squared orthogonal distances between each point and its assigned hyperplane. The formulation includes unit Euclidean norm constraints on the hyperplane normal vectors, rendering the problem nonconvex and difficult to solve to global optimality. This work introduces an iterative polyhedron approximation with a spatial branch-and-bound (IPA-SBB) framework that provides a global solution to optimization problems constrained by the unit Euclidean norm. The approach iteratively refines a polyhedral relaxation of the nonconvex feasible region within a spatial branch-and-bound scheme, preserving valid bounds at each node while offering theoretical approximation guarantees. Computational experiments show that IPA-SBB performs more efficiently than the default Xpress solver, with speeds up to 72 times faster and almost 123 times fewer nodes explored on average. The approach also applies readily to other machine learning problems with norm-based restrictions, such as linear classification, demonstrating its efficacy and versatility.

4.1 Introduction

Clustering is an important unsupervised learning task that combines data into groups or clusters based on similarity of data points. The main objective is to minimize the sum of distances between the points and the center of each cluster. Many classical clustering algorithms, such as k-means, are based on defining a central point for each cluster. Similarly, k-medians, k-medoids, and fuzzy c-means are all clustering methods that rely on a single point center. Application examples for these algorithm are customer segmentation (Kansal et al., 2018; Tabianan et al., 2022), image segmentation (Dhanachandra et al., 2015; Mittal et al., 2021), and anomaly detection (Kumari et al., 2016; Pu et al., 2021).

Nonetheless, not all datasets fit to this point-centric paradigm. In many practical scenarios, data may lie near multiple low-dimensional subspaces rather than being concentrated around distinct point centers. In such cases, clustering by fitting multiple hyperplanes becomes essential. This approach is particularly valuable in applications where traditional point-based models such as k-means fail to capture the underlying geometric structure of the data, including subspace segmentation, anomaly detection, and piecewise-linear regression. *Hyperplane-based clustering*, first introduced by Bradley and Mangasarian (2000), addresses this limitation by grouping data points that lie close to a common hyperplane, thereby providing a more accurate and interpretable representation of complex, structured datasets than algorithms that rely on a single central point. We refer to this problem as a k-hyperplane clustering (k-HC) problem.

There is a family of clustering problems based on hyperplanes. Projected clustering, also called subspace clustering, was introduced by Aggarwal et al. (1999). It seeks a subset of the original high-dimensional axes along which to group points that share similar values, and distances are computed only within this selected subspace. Projective clustering later extended this idea to arbitrarily oriented subspaces, which is equivalent to finding low-dimensional hyperplanes that best fit the data points. This problem was proposed by Aggarwal and Yu (2000) at about the same time as k-HC and can be considered a generalization of the k-HC problem.

In formal terms, the k-HC problem aims to partition a dataset $\mathcal{A} = \{a_1, \ldots, a_m\}$ in real n-dimensional space $\mathbb{R}^n$ into k clusters. Each cluster is grouped by a hyperplane $H_j = \{x \in \mathbb{R}^n \mid w_j^\top x = \gamma_j\}$, where $w_j \in \mathbb{R}^n$ and $\gamma_j \in \mathbb{R}$. The objective of this nonconvex minimization problem is to minimize the total sum of squared Euclidean (orthogonal) distances from each point to its corresponding closest hyperplane. Megiddo and Tamir (1982), addressing k-HC in the context of facility location in the plane, prove that k-HC is $\mathcal{NP}$-hard even for the case where all the data points can be fit with zero error.

To make the measurement of distances between a hyperplane and points scale-invariant, a unit norm constraint is applied for each hyperplane. When a unit ℓ_2-norm constraint is imposed on each variable w defining a hyperplane, the constraint can be relaxed to $\|w\|_2 \geq 1$, since the optimal solution will lie on the surface of the ball (details in Section 4.3). Then, the geometry of the problem leads to infeasible balls, and so the feasible region for each hyperplane is a concave set, specifically $\mathbb{R}^n \setminus \{x : \|x\|_2 < 1\}$. This problem has a structure where directly convexifying the feasible region would result in the entire space. Therefore, this problem requires a specific algorithm to solve it.

Solving the hyperplane clustering problem heuristically can lead to local solutions that may not accurately capture the underlying data structure, resulting in misleading interpretations of clusters and less meaningful outcomes. In this work, we integrate an *IPA-SBB* framework into the MIQCQP for k-HC. For each unit ℓ_2-norm constraint, we relax the feasible region with a polyhedron that iteratively improves close to the feasible region of the unit ℓ_2-norm ball constraint. This strategy yields provable optimality gaps—and in many cases exact solutions—whereas existing continuous relaxations and heuristics cannot guarantee global performance.

With this background, the main contributions of this work are summarized as follows:

(i) We introduce an iterative polyhedron approximation with a spatial branch-and-bound (IPA-SBB) framework for solving optimization models with unit ℓ_2-norm constraints to global optimality.

(ii) We develop an implementation of the IPA-SBB method within the Xpress optimization solver for both k-HC and linear classification problems. The approach employs multi-way branching and achieves significant performance improvements over the default Xpress solver.

(iii) We derive an approximation factor for replacing the ℓ_2-norm ball with an n-simplex, which is an initial polyhedron approximation for the IPA-SBB method.

The remainder of the work is organized as follows. Section 4.2 reviews previous literature on continuous and polyhedral methods for unit ℓ_2-norm constrained optimization and provides problem formulation and heuristic methods for the k-HC problem. In Section 4.3, we present the IPA-SBB algorithm in detail, including the construction of the polyhedral approximations and branching rules. Section 4.4 reports comprehensive computational experiments comparing IPA-SBB against an Xpress solver. We further examine the efficiency of this method on linear classification problems in Section 4.5. Finally, Section 4.6 concludes with a discussion of practical trade-offs and potential extensions.

4.2 Preliminaries

4.2.1 Linear Algebra Preliminaries

This subsection reviews several fundamental concepts from linear algebra that are used throughout this work. These results form the theoretical foundation for the algorithms and numerical methods developed in the following sections.

4.2.1.1 Gram-Schmidt Orthogonalization

The Gram-Schmidt process constructs an orthogonal (or orthonormal) basis from a set of linearly independent vectors. Given $\{v_1, v_2, \ldots, v_n\} \subset \mathbb{R}^m$, the algorithm successively removes the components of each vector that lie in the span of the previous ones, producing orthogonal vectors $\{u_1, u_2, \ldots, u_n\}$ defined by

$$u_k = v_k - \sum_{j=1}^{k-1} \frac{u_j^T v_k}{u_j^T u_j} u_j, \quad k = 1, 2, \ldots, n. \tag{4.1}$$

Normalizing these vectors yields an orthonormal basis $\{q_k\}$ with $q_k = u_k / \|u_k\|$.

4.2.1.2 Inverse Power Method

The inverse power method is an iterative technique for computing the eigenvector associated with the smallest-magnitude eigenvalue of a matrix. For a nonsingular matrix $A \in \mathbb{R}^{n \times n}$ with eigenvalues $|\lambda_1| < |\lambda_2| \leq \cdots \leq |\lambda_n|$, the iteration

$$Ay_{k+1} = x_k, \quad x_{k+1} = \frac{y_{k+1}}{\|y_{k+1}\|}, \tag{4.2}$$

converges to the eigenvector corresponding to λ_1.

4.2.1.3 Condition Number

The condition number of a nonsingular matrix A quantifies the sensitivity of the solution of a linear system $Ax = b$ to perturbations in either A or b. It is defined, with respect to the 2-norm, as

$$\kappa_2(A) = \|A\|_2 \|A^{-1}\|_2 = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)}, \tag{4.3}$$

where $\sigma_{\max}(A)$ and $\sigma_{\min}(A)$ denote the largest and smallest singular values of A, respectively. A large condition number indicates that A is ill-conditioned, meaning

that small numerical errors or perturbations can lead to large deviations in the computed solution.

4.2.2 Previous works

In addition to introducing the k-HC problem, Bradley and Mangasarian (2000) derived a closed-form procedure for determining the optimal hyperplanes once all data points have been assigned to clusters. For each cluster $\ell = 1, \ldots, k$, let $A(\ell)$ denote the $m(\ell) \times n$ matrix whose rows are the vectors a_i belonging to cluster ℓ. Define

$$B(\ell) \;=\; A(\ell)^\top \left(I - \frac{ee^\top}{m(\ell)} \right) A(\ell), \tag{4.4}$$

where e is the vector of ones. The optimal hyperplane, written as $w_\ell^\top x = \gamma_\ell$, is obtained by setting w_ℓ to be the eigenvector associated with the smallest eigenvalue of $B(\ell)$, and

$$\gamma_\ell \;=\; \frac{e^\top A(\ell) w_\ell}{m(\ell)}. \tag{4.5}$$

A heuristic procedure, analogous to the k-means algorithm, then reassigns each data point a_i to its nearest hyperplane according to

$$\ell \;=\; \arg\min_{1 \le j \le k} \left| a_i^\top w_j - \gamma_j \right|. \tag{4.6}$$

This process repeats until the cluster assignments no longer change or the objective function no longer decreases. Because of its simplicity, we will use this heuristic to generate feasible solutions for the problem. In this case, we apply the inverse power method to find the eigenvector w corresponding to the smallest eigenvalue in order to deal with numerical issues.

The formulation of the k-HC requires a unit norm to ensure the distance measure is invariant (see Section 4.3 for details). Many techniques have been developed to address these non-convex ℓ_2-norm constraints. In the related field of subspace clustering, Qiu and Sapiro (2015) introduced a framework that applies a linear transformation to the data to restore a low-rank structure, using the nuclear norm as a convex surrogate for the rank function in its optimization criteria. In signal processing, gradient-based techniques, such as projections on a manifold (Douglas et al., 2000; Filho and Romano, 2014), enforce equivalent unit-norm constraints.

To enhance cluster separability, the k-Proximal Plane Clustering method (Liu et al., 2016) modifies the objective so that each hyperplane is not only close to its own data points but also distant from those of other clusters. This formulation can be efficiently solved by computing the eigenvector associated with the smallest eigenvalue of a suitably constructed matrix, leading to locally optimal solutions (Shao et al., 2013). The

problem has also been generalized from hyperplanes to affine subspaces of arbitrary dimensions, giving rise to k-flat clustering (Tseng, 2000; Zhang et al., 2009) and the broader k-subspace clustering framework (Guo and Bai, 2023; Wang et al., 2009).

In contrast to these related approaches, our proposed IPA-SBB framework addresses the nonconvex unit ℓ_2-norm constraints within a global optimization setting. Rather than relying on convex surrogates or locally optimal eigenvalue-based updates, IPA-SBB iteratively refines a polyhedral relaxation integrated into a sB&B scheme, providing provable optimality bounds and improved computational performance.

4.3 Method

In this section, we present an algorithm based on polyhedron approximation for the unit ℓ_2-norm ball that is iteratively enlarged to approximate the ball. We refer to this method as the IPA-SBB framework.

We first formally state the k-HC problem with the following model:

$$\min_{x,w,\gamma} \quad \sum_{i=1}^{m}\sum_{j=1}^{k}\left(\frac{w_j^T a_i - \gamma_j}{\|w_j\|_2}\right)^2 x_{ij} \tag{4.7a}$$

$$\text{s.t.} \quad \sum_{j=1}^{k} x_{ij} = 1, \qquad i = 1,\dots,m \tag{4.7b}$$

$$x_{ij} \in \{0,1\}, \qquad i = 1,\dots,m, \quad j = 1,\dots,k. \tag{4.7c}$$

To define a hyperplane in $\mathbb{R}^n$, we need n points to fix its orientation. However, a hyperplane in this model is defined by $n+1$ parameters (n for the weight vector w and one for the bias term γ). This gives us one degree of freedom. We can use this degree of freedom to impose the constraint $\|w_j\|_2 = 1$ for each $j = 1,\dots,k$, which makes the distance measurement scale-invariant. With the objective of minimizing $\sum_{i=1}^{m}\sum_{j=1}^{k}\left(w_j^T a_i - \gamma_j\right)^2 x_{ij}$, the constraint can be relaxed to $\|w_j\|_2 \geq 1$. This is because any optimal solution can be scaled down until the constraint becomes an equality, i.e., $\|w_j\|_2 = 1$.

Building upon this, we can write the QCQP formulation stated in Amaldi and Coniglio (2013) as follows:

$$\min \quad \sum_{i=1}^{m} y_i^2 \tag{4.8a}$$

$$\text{s.t.} \quad \sum_{j=1}^{k} x_{ij} = 1, \quad \forall i = 1, \dots, m, \tag{4.8b}$$

$$y_i \geq (w_j^T a_i - \gamma_j) - M(1 - x_{ij}), \quad \forall i = 1, \dots, m, \forall j = 1, \dots, k, \tag{4.8c}$$

$$y_i \geq (-w_j^T a_i + \gamma_j) - M(1 - x_{ij}), \quad \forall i = 1, \dots, m, \forall j = 1, \dots, k, \tag{4.8d}$$

$$\|w_j\|_2 \geq 1, \quad \forall j = 1, \dots, k, \tag{4.8e}$$

$$y_i \geq 0, \quad \forall i \in 1, \dots, m, \tag{4.8f}$$

$$w_j \in \mathbb{R}^n, \gamma_j \in \mathbb{R}, \quad \forall j = 1, \dots, k, \tag{4.8g}$$

$$x_{ij} \in \{0, 1\} \quad \forall i = 1, \dots, m, \forall j = 1, \dots, k, \tag{4.8h}$$

where $x_{ij} = 1$ if the point a_i is assigned to the cluster j, and 0 otherwise, y_i is the distance between point a_i and its corresponding hyperplane, and (w_j, γ_j) is the parameters of the hyperplane j for $i \in \{1, \dots, m\}$ and $j \in \{1, \dots, k\}$. The Big-M term in (4.8c) and (4.8d) is bounded by the length of the diagonal of the smallest hypercube that encapsulates all data points, i.e., $M = h\sqrt{n}$ where h is the edge length of the hypercube. This will be the basic formulation used throughout this work.

4.3.1 Polyhedral Disjunctive Program

We begin by denoting the unit ℓ_2-norm ball in $\mathbb{R}^n$ as

$$C \;=\; \{\, w \in \mathbb{R}^n \mid \|w\|_2 \leq 1\}, \tag{4.9}$$

and let P be a polyhedron satisfying $P \subset C$ and $0 \in P$. We write

$$C^{\complement} \;=\; \{\, w \in \mathbb{R}^n \mid \|w\|_2 \geq 1\}, \qquad P^{\complement} \;=\; \mathbb{R}^n \setminus \operatorname{int}(P) \tag{4.10}$$

for the complement of these sets. Because $P \subset C$, we have $C^{\complement} \subset P^{\complement}$, and our goal is to approximate $C^{\complement}$ ever more tightly by generating a sequence of polyhedral complements $P^{\complement}$.

To incorporate these approximations into our optimization, we replace the nonconvex ℓ_2-norm constraint, $\|w\|_2 = 1$, by the disjunctive constraint that $w \in P^{\complement}$. Let S be the set of all remaining (linear or convex) constraints, the feasible region becomes $P^{\complement} \cap S$, which we model via a disjunction on the facets of P. If P is given as

$$P = \{w : a_i^\top w \leq b_i,\ i = 1, \dots, m\}, \tag{4.11}$$

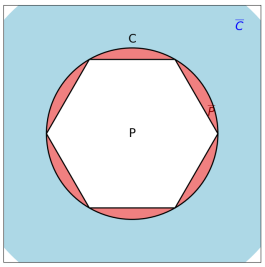


FIGURE 4.1: Visualization of a polyhedron (P) within a ball (C).

then

$$P^{\complement} \cap S \;=\; \big(a_1^\top w \geq b_1\big) \;\vee\; \cdots \;\vee\; \big(a_m^\top w \geq b_m\big), \quad (w,x) \in S. \tag{4.12}$$

This is equivalent to disjunctive program,

$$\big(a_1^\top w \geq b_1, (w,x) \in S\big) \;\vee\; \cdots \;\vee\; \big(a_m^\top w \geq b_m, (w,x) \in S\big), \tag{4.13}$$

which can be solved by solving each term of the disjunctive program.

We suppose that the vertices of polyhedron P belong to the border of the ball C, since we can trivially improve the polyhedron approximation if some vertices of P do not belong to the border by projecting those vertices onto the ball.

4.3.2 Iterative Polyhedron Refinement

Suppose we solve this relaxation to optimality and obtain w^*. If w^* does not lie on any facet F of P, we can then scale w^* to obtain a better solution. Therefore, w^* must lie on some facet F of P. To cut off w^* in the next iteration, we project it radially onto the unit sphere,

$$\pi(w^*) \;=\; \frac{w^*}{\|w^*\|_2}, \tag{4.14}$$

and then refine our polyhedron by setting

$$P' \;=\; \operatorname{conv}\big(P \;\cup\; \{\pi(w^*)\}\big). \tag{4.15}$$

If w^* belongs to exactly one facet F of P, the new region $P' \setminus P$ is the convex hull of F and $\pi(w^*)$, which introduces $f(F)+1$ new facets, where $f(F)$ is the number of facets of F. Moreover, the facet F is not a facet of P'.

A particular case can occur when $n > 2$, where the intersection of two sets $\{(w,x) \in S \mid a_i w \geq b_i\}$ and $\{(w,x) \in S \mid a_j w \geq b_j\}$ for some $i \neq j$ is not empty. Since w^* must belong to a facet of P, this is only the case if w^* belongs to the borders of the facets of P. Suppose that the optimal solution w^* to current relaxation belongs to the

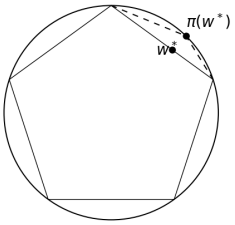


FIGURE 4.2: Refining a polyhedron by projecting w^* onto the ball and creating new facets, shown as dashed lines, to improve the approximation of the ball.

relative border of F. We let F' be the relative facet of F that contains w^*, and let $F'' := conv(F' \cup \pi(w^*))$. Since $w^* \in F'$, it will also be $w^* \in F''$, thus making the refinement operation degenerate, being unable to cut the previous solution by making it infeasible. Thus, we avoid the introduction of the term of the disjunction that corresponds to the new facet F'' because there exists another node whose feasible region contains the point w^* in our implementation. By construction, $w^*, \pi(w^*) \in F''$, thus its affine hull contains the line through w^* and $\pi(w^*)$, which contains the origin.

We handle this case with an *a posteriori* procedure: during the branching phase, we generate a node for each facet without initially verifying if w^* is contained within any of them. When we solve a node corresponding to a facet F, we check whether $0 \in aff(F)$ by examining the rank of matrix E, where each column of E represents an extreme point of F. If E is not of full rank, then there exists a non-zero vector in its null space. This implies that the origin can be expressed as a non-zero linear combination of E's columns. We also set the relative condition number to 10^{-4} to treat any value relatively smaller than this threshold as zero. This approach improves numerical stability when calculating the interactions between two close facets.

4.3.3 Integration with Spatial Branch-and-Bound

In a sB&B perspective, the initial polyhedron P is created at a dummy node at the root. The branching operation is implemented similarly, generating a son for each facet of P that contains the inequality $a_i x \geq b_i$, for $i \in \{1, \ldots, m\}$. Since $0 \in P$ but $0 \notin P^{\mathbb{C}}$, the trivial solution $w = 0$ is infeasible below the root, which improves dual bound in the branch-and-bound algorithm.

We configure the branching settings to control the algorithm's search behavior. Specifically, we've set `NODESELECTION = 4`, `BACKTRACK = 5`, and `BACKTRACKTIE = 5`. We selected these values to emulate a breadth-first search strategy. The idea behind this is to first create an exponential number of nodes, $f(P)^k$, after the dummy root node, where P is the initial polyhedron. We then solve all nodes in this layer to obtain

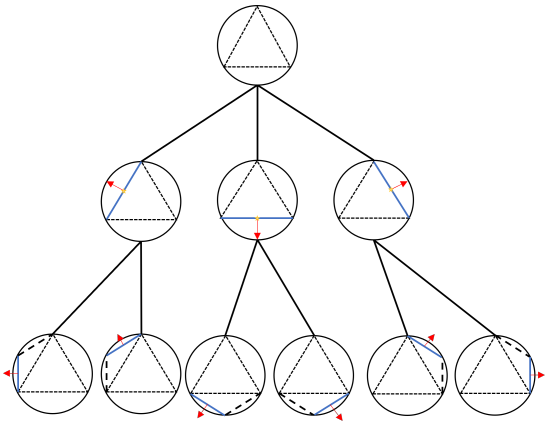


FIGURE 4.3: Example of an enumeration tree using iterative polyhedral approximation in $\mathbb{R}^2$ with one hyperplane. The two-simplex centered at the origin serves as the constraint in the first layer and is iteratively refined to more closely approximate the ball constraint.

a non-zero w solution. For more detailed information on these and other branching parameters, see Chapter 7 of FICO (2025).

We alternatively select to branch on continuous variable w with probability $\max\{\text{Gap}, 1 - \text{Gap}\}$, where Gap is the MIPGAP reported as $\frac{z^{\text{UB}} - z^{\text{LB}}}{z^{\text{UB}}}$. Here z^{LB} and z^{UB} are the current lower and upper bounds, respectively. Otherwise, we branch on binary variable x as usual. This is because branching on variable w in the early stages often yields a poor upper bound, as feasible solutions are rarely obtained. Therefore, we first branch on x until we get a non-zero lower bound. Moreover, when the node contains all integer values for x, we apply the closed-form solution described in Section 4.2 to improve the upper bound. In this case, we obtain the optimal hyperplanes given a fixed integer solution for x; therefore, we do not need to branch on w to improve the polyhedrons, and we can prune this node due to its optimality. We also generate 100 random heuristic solutions from Bradley and Mangasarian (2000) to the problem to further improve the initial upper bound.

4.3.4 Choice of Initial Polyhedron

Choosing a good starting polyhedron is critical. A large number of facets at the root creates too many branches, while facets with many relative facets bloat the refinement step. Here we first show that the smallest number of facets must be at least n in $\mathbb{R}^n$ to exclude the trivial solution $w = 0$.

Lemma 4.1. *Let $P \subset \mathbb{R}^n$ be a non-empty, bounded polyhedron. If the origin is in the interior of P, then P must have at least n facets.*

Proof. Let the polyhedron be defined as $P = \{w \in \mathbb{R}^n \mid a_i^\top w \leq b_i,\ i = 1, \ldots, m\}$. The condition $0 \in \text{int}(P)$ implies that $a_i^\top \cdot 0 < b_i$, and thus $b_i > 0$ for all i. Assume for the

sake of contradiction that P has fewer than n facets, i.e., $m < n$. The set of m normal vectors $\{a_i\}_{i=1}^m$ is linearly dependent and cannot span $\mathbb{R}^n$. Consequently, there exists a non-zero vector $d \in \mathbb{R}^n$ in the orthogonal complement of their span, such that $a_i^\top d = 0$ for all i. Now, let w_0 be any point in P. Consider the line $L = \{w_0 + \lambda d \mid \lambda \in \mathbb{R}\}$. For any point $w \in L$, we have:

$$a_i^\top w = a_i^\top (w_0 + \lambda d) = a_i^\top w_0 + \lambda (a_i^\top d) = a_i^\top w_0 \leq b_i$$

This shows that the entire infinite line L is contained within P. However, this contradicts the given condition that P is bounded. Therefore, our initial assumption that $m < n$ must be false. It follows that $m \geq n$. □

An n-simplex centered at the origin, with its $n+1$ vertices on the unit sphere, offers an ideal compromise: it has only $n+1$ facets, each of which is an $(n-1)$-simplex. To construct such a simplex in $\mathbb{R}^n$, we begin with a regular $(n+1)$-simplex in $\mathbb{R}^{n+1}$, defined as the convex hull of the standard basis vectors $e_1, \ldots, e_{n+1}$ and the origin $P = \operatorname{conv}\{\, 0, e_1, e_2, \ldots, e_{n+1} \,\}$. Among its facets, the one opposite the origin is $F = \operatorname{conv}\{\, e_1, e_2, \ldots, e_{n+1} \,\}$, which is an n-simplex lying in the affine hyperplane $A = \{\, x \in \mathbb{R}^{n+1} : \mathbf{1}^\top x = 1 \,\}$, where $\mathbf{1} = (1, \ldots, 1)^\top$. The centroid of this facet is $c = \frac{1}{n+1}\mathbf{1}$. To center the simplex at the origin, we translate its vertices by $-c$, defining

$$v_i = e_i - c, \quad i = 1, \ldots, n+1. \tag{4.16}$$

These translated points satisfy $\sum_{i=1}^{n+1} v_i = 0$, and they now lie in the linear subspace

$$A_0 = \{\, y \in \mathbb{R}^{n+1} : \mathbf{1}^\top y = 0 \,\}, \tag{4.17}$$

the subspace parallel to A.

Next, we construct a basis for A_0. The vectors $\{v_1, \ldots, v_n\}$ are linearly independent and span A_0. To obtain an orthonormal basis, we apply the Gram–Schmidt process to these vectors and denote by $U \in \mathbb{R}^{(n+1)\times n}$ the matrix whose columns are the resulting orthonormal vectors $u_1, \ldots, u_n$. This matrix defines an orthogonal transformation that maps points from $\mathbb{R}^{n+1}$ to $\mathbb{R}^n$. For each vertex e_i of the original simplex, we compute its coordinates in $\mathbb{R}^n$ by $e_i' = U^\top (e_i - c)$, yielding the projected coordinates $e_i' \in \mathbb{R}^n$. These e_i' form the vertices of an n-simplex centered at the origin. Finally, we scale the simplex so that all vertices have unit ℓ_2-norm.

We record two basic properties of these vertices in the following Lemma.

Lemma 4.2. *Let $e_1', \ldots, e_{n+1}' \in \mathbb{R}^n$ be the vertices obtained by the construction above. Then, for all $i \neq j$,*

$$e_i' \cdot e_j' = -\frac{1}{n} \qquad \text{and} \qquad \sum_{i=1}^{n+1} e_i' = 0. \tag{4.18}$$

Proof. For $i \neq j$,

$$v_i \cdot v_j = (e_i - c)^\top (e_j - c) = e_i^\top e_j - e_i^\top c - c^\top e_j + c^\top c = 0 - \tfrac{1}{n+1} - \tfrac{1}{n+1} + \tfrac{1}{n+1} = -\tfrac{1}{n+1}, \tag{4.19}$$

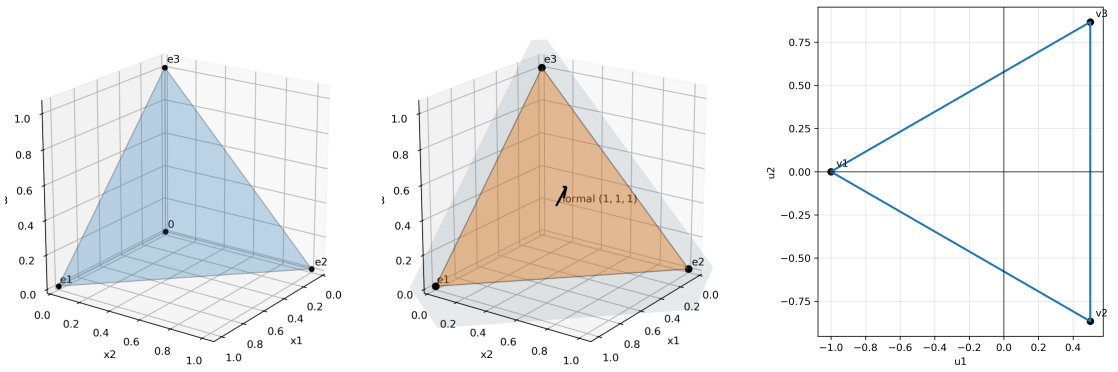
and $\|v_i\|_2^2 = v_i \cdot v_i = 1 - \frac{1}{n+1} = \frac{n}{n+1}$. Since U has orthonormal columns spanning A_0, $U^\top$ is an isometry from A_0 onto $\mathbb{R}^n$, hence

$$\tilde{e}_i := U^\top v_i \quad \text{satisfy} \quad \tilde{e}_i \cdot \tilde{e}_j = v_i \cdot v_j \quad \text{and} \quad \|\tilde{e}_i\|_2 = \|v_i\|_2 = \sqrt{\tfrac{n}{n+1}}. \tag{4.20}$$

After the common normalization $e_i' = \sqrt{\frac{n+1}{n}}\,\tilde{e}_i$, we obtain, for $i \neq j$,

$$e_i' \cdot e_j' = \frac{\tilde{e}_i \cdot \tilde{e}_j}{\|\tilde{e}_i\|_2 \, \|\tilde{e}_j\|_2} = \frac{-\frac{1}{n+1}}{\frac{n}{n+1}} = -\frac{1}{n}. \tag{4.21}$$

Since $\sum_{i=1}^{n+1} v_i = 0$ we have $\sum_{i=1}^{n+1} e_i' = \sqrt{\frac{n+1}{n}} \sum_{i=1}^{n+1} \tilde{e}_i = \sqrt{\frac{n+1}{n}} U^\top \sum_{i=1}^{n+1} v_i = 0$. □



(A) 3-simplex in $\mathbb{R}^3$ (B) Subspace A (C) Projected 2-simplex

FIGURE 4.4: Visualization of the 3-simplex, its subspace A, and the projected origin-centered 2-simplex.

4.3.5 Deriving Linear Relaxations

While it might seem intuitive to define each relaxation's feasible region by intersecting the original problem's feasible set with the current node's corresponding facet F, this strategy fails. The main problem is that the branching process makes this facet F infeasible for the sons of the current node.

To derive linear relaxations at each node, we define *down-extension* and *up-extension* of a facet F as,

$$D(F) \;=\; \operatorname{conv}(F \cup \{0\}), \qquad U(F) \;=\; \operatorname{cl}\big(\operatorname{cone}(F) \setminus D(F)\big), \tag{4.22}$$

where $\operatorname{cl}(\cdot)$ is the closure operation. For any two facets F and F' where the node

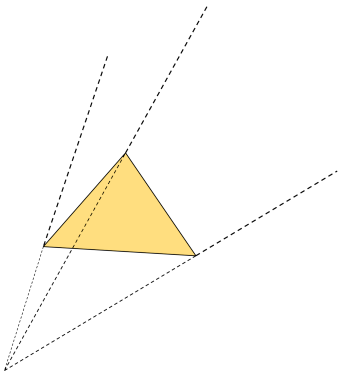


FIGURE 4.5: Illustration of a polytope (filled region), showing its down-extension (dotted border) and up-extension (dashed border).

corresponding to F is an ancestor of the node corresponding to F', we have $F' \subset U(F)$ due to the definition of up-extension and our branching mechanism. Therefore, $U(F)$ is feasible for all descendants of the node corresponding to F.

We describe $U(F)$ using linear inequalities. We can see that $D(F)$ is simply $U(F)$ with the opposite sign of inequalities, where the origin is contained in $D(F)$. Let $E \in \mathbb{R}^{n \times n}$ be a matrix where each row corresponds to an extreme point of F. By construction, the affine space of F does not contain the origin; thus, we look for an inequality of the form $aw \geq 1$. The normal vector a is found by solving the linear system $Ea = e$. The other inequalities correspond to hyperplanes that contain any set of $n-1$ extreme points of F plus the origin. Let $E' \in \mathbb{R}^{(n-1) \times n}$ be a matrix containing $n-1$ extreme points of F. We are looking for a hyperplane of the form $aw \geq 0$, which can be obtained by solving for a nonzero vector a that belongs to the null space of E'.

The drawback of this representation is its susceptibility to numerical errors as the algorithm approaches convergence. This issue is particularly pronounced when the polyhedron's facets become small and its extreme points are clustered closely together, leading to a geometrically degenerate and mathematically ill-conditioned system. The matrix E representing this geometry becomes nearly singular, indicated by a large condition number, which means small floating-point errors in the input vector e can be greatly amplified in the solution vector a. To mitigate this, we employ a two-part strategy. First, we avoid further branching when the geometric improvement is negligible, which we detect by checking if the rank of the new matrix E' is unchanged from the parent matrix E, or if the matrices are otherwise close within a numerical tolerance. This prevents further degradation of precision for minimal gain. Second, we adapt our method for solving the linear system $Ea = e$ based on its stability. If the condition number of E is below a threshold like 10^7 (Berthold and Hendel, 2021), we use a standard, efficient direct solver based on LU decomposition. Otherwise, for an ill-conditioned system, we avoid numerically unstable methods like calculating the explicit inverse E^{-1}. Instead, we employ a more robust algorithm, such as one based

on Singular Value Decomposition (SVD) to find a stable, least-squares solution via the pseudoinverse, a standard practice in robust numerical solvers.

4.3.6 Approximation Factors

In this subsection we replace the ℓ_2 ball constraint by a simplex disjunction constraint. We first define the *simplex gauge* which its level set is the n-simplex centered at the origin. We then show the relationship between the simplex gauge and ℓ_2 norm constraint, and derive the approximation factor for this replacement.

Lemma 4.3 (Simplex gauge). *Let $K = \operatorname{conv}\{p_0, \dots, p_n\}$ be a regular n-simplex centered at the origin. Then*

$$K \;=\; \bigcap_{i=0}^{n} \{x \in \mathbb{R}^n :\; p_i^\top x \geq -1/n\}, \tag{4.23}$$

and the Minkowski functional gauge of K is

$$\gamma_K(x) \;:=\; \inf\{t > 0 :\; x \in tK\} \;=\; -\,n \min_i p_i^\top x \;=:\; \|x\|_\Delta. \tag{4.24}$$

Proof. First, $\{p_0, \dots, p_n\}$ span $\mathbb{R}^n$ and $\sum_{i=0}^n p_i = 0$. Let $x \in K$ so $x = \sum_{j=0}^n \lambda_j p_j$ with $\lambda_j \geq 0, \sum_j \lambda_j = 1$. Then for any $i \in \{0, 1, \dots, n\}$,

$$p_i^\top x = \sum_{j=0}^{n} \lambda_j\, p_i^\top p_j = \lambda_i \cdot 1 + \sum_{j \neq i} \lambda_j \big(-\tfrac{1}{n}\big) = -\tfrac{1}{n} + \Big(1 + \tfrac{1}{n}\Big)\lambda_i \;\geq\; -\tfrac{1}{n}, \tag{4.25}$$

so x satisfies all $n+1$ inequalities. Conversely, suppose x satisfies $p_i^\top x \geq -\frac{1}{n}$ for all $i \in \{0, 1, \dots, n\}$. Define $\lambda_i := \dfrac{1 + n\, p_i^\top x}{n+1}$ (≥ 0) and set $y := \sum_{i=0}^n \lambda_i p_i$. Then $\sum_{i=0}^n \lambda_i = \frac{(n+1)+n\sum_i p_i^\top x}{n+1} = \frac{(n+1)+n\,(\sum_i p_i)^\top x}{n+1} = 1$, and for each $k \in \{0, 1, \dots, n\}$,

$$p_k^\top y = \sum_{i=0}^{n} \lambda_i\, p_k^\top p_i = -\tfrac{1}{n} + \Big(1 + \tfrac{1}{n}\Big)\lambda_k = -\tfrac{1}{n} + \Big(1 + \tfrac{1}{n}\Big)\frac{1 + n\, p_k^\top x}{n+1} = p_k^\top x. \tag{4.26}$$

Thus $p_k^\top (y - x) = 0$ for all k. Since $\{p_0, \dots, p_n\}$ span $\mathbb{R}^n$, it follows that $y = x$, so $x = \sum_{i=0}^n \lambda_i p_i \in K$. This proves $K = \bigcap_{i=0}^n \{x :\; p_i^\top x \geq -1/n\}$.

For the gauge, $x \in tK$ iff $p_i^\top (x/t) \geq -1/n$ for all i, i.e., $t \geq -\,n\, \min_i p_i^\top x$. Taking the infimum over $t > 0$ gives $\gamma_K(x) = -\,n \min_i p_i^\top x$. □

Lemma 4.4. *The minimum value of $\|x\|_2^2$ subject to $\|x\|_\Delta = 1$ equals to $1/n^2$. Consequently, for all $x \in \mathbb{R}^n$,*

$$\|x\|_2 \;\leq\; \|x\|_\Delta \;\leq\; n\, \|x\|_2. \tag{4.27}$$

Proof. Consider the Lagrangian

$$L(x,\lambda) \;=\; \tfrac{1}{2}\|x\|_2^2 + \lambda\big(\|x\|_\Delta - 1\big) \;=\; \tfrac{1}{2}x^\top x + \lambda\big(-n\,p_j^\top x - 1\big), \tag{4.28}$$

where j is chosen so that $p_j^\top x = \min_i p_i^\top x$; on the boundary $\|x\|_\Delta = 1$ this means $-n\,p_j^\top x = 1$. The KKT stationarity condition is

$$\nabla_x L \;=\; x - \lambda n\,p_j \;=\; 0 \quad\Longrightarrow\quad x = \lambda n\,p_j. \tag{4.29}$$

Enforcing the active constraint gives

$$1 \;=\; -\,n\,p_j^\top x \;=\; -\,n\,p_j^\top(\lambda n\,p_j) \;=\; -\,\lambda n^2\,\|p_j\|_2^2 \;=\; -\,\lambda n^2, \tag{4.30}$$

so $\lambda = -\frac{1}{n^2}$ and therefore $x = -\frac{1}{n}\,p_j$. Consequently,

$$\|x\|_2^2 \;=\; \Big\|-\tfrac{1}{n}p_j\Big\|_2^2 \;=\; \tfrac{1}{n^2}\|p_j\|_2^2 \;=\; \tfrac{1}{n^2}. \tag{4.31}$$

From the KKT argument above, the minimum Euclidean norm on the boundary $\{x : \|x\|_\Delta = 1\}$ is $1/n$, attained at $-\frac{1}{n}p_j$, and the maximum is 1, attained at the vertices p_i. Thus $\|x\|_2 \le \|x\|_\Delta \le n\,\|x\|_2$. □

Finally, we prove this relationship for the optimal objective values of these two settings.

Theorem 4.5. *Let $OPT(k-HC)_2$ be the optimal objective value of the k-HC problem and $OPT(k-HC)_\Delta$ be the optimal value of the relaxed problem using the n-simplex distance. The following relationship holds:*

$$\frac{1}{n^2}OPT(k-HC)_2 \le OPT(k-HC)_\Delta \le OPT(k-HC)_2.$$

Proof. From (4.27), squaring and taking reciprocals gives

$$\frac{1}{n^2\|u\|_2^2} \;\le\; \frac{1}{\|u\|_\Delta^2} \;\le\; \frac{1}{\|u\|_2^2}. \tag{4.32}$$

Applying (4.32) with $u = w_j$ yields

$$\frac{|w_j^\top x - \gamma_j|}{n^2\|w_j\|_2^2} \;\le\; \frac{|w_j^\top x - \gamma_j|}{\|w_j\|_\Delta^2} \;\le\; \frac{|w_j^\top x - \gamma_j|}{\|w_j\|_2^2}, \tag{4.33}$$

for every j and x. Taking the minimum over $j = 1,\dots,k$ preserves the inequalities, and summing over $i = 1,\dots,m$ preserves them as well:

$$\frac{1}{n^2}\sum_{i=1}^{m}\min_j \frac{|w_j^\top x_i - \gamma_j|}{\|w_j\|_2^2} \;\le\; \sum_{i=1}^{m}\min_j \frac{|w_j^\top x_i - \gamma_j|}{\|w_j\|_\Delta^2} \;\le\; \sum_{i=1}^{m}\min_j \frac{|w_j^\top x_i - \gamma_j|}{\|w_j\|_2^2}. \tag{4.34}$$

Finally, taking the infimum over all choices of $\{(w_j, \gamma_j)\}_{j=1}^{k}$ gives

$$\frac{1}{n^2}\,\mathrm{OPT}(k\text{-HC})_2 \;\le\; \mathrm{OPT}(k\text{-HC})_\Delta \;\le\; \mathrm{OPT}(k\text{-HC})_2. \tag{4.35}$$

□

Theorem 4.5 provides provable bounds for the k-HC models when the ℓ_2-norm constraints are replaced with the disjunctive constraints defined by the n-simplex. The lower bound factor is equal to $1/n^2$; this factor will be further tightened as the approximating polyhedron is refined throughout the algorithm.

4.4 Computational Experiments

This section evaluates the proposed IPA-SBB framework through a comprehensive set of computational experiments. All k-HC models are implemented in Python and solved using the Xpress optimization solver. Xpress is particularly well suited to this application because it supports multi-way branching, enabling the solver to generate more than two child nodes per split. All computations are conducted on a laptop equipped with an Intel Core i7–13620H processor (2.40 GHz) and 32 GB RAM, running Xpress Academic 9.5.4 with Python 3.8.3. We solve each instance within a 3,600 s time limit, applying an optimality tolerance of 10^{-4} for k-HC and 10^{-6} for linear classifiers.

Synthesis instances are generated by first constructing k random hyperplanes within the unit cube $[0, 1]^n$. A dataset of m points is then produced by repeatedly sampling a hyperplane according to predefined mixture weights, drawing the first $n-1$ coordinates uniformly from $[0, 1]$, and solving for the final coordinate so that the point lies exactly on the selected hyperplane. A small perturbation is optionally added in the direction of the hyperplane's normal vector to simulate noise. This procedure allows precise control over dimensionality and noise level while preserving known ground-truth cluster assignments.

We consider two categories of datasets. The `LowDim` group contains instances in which both the dimension n and the number of hyperplanes k are at most three, while the `HighDim` group includes instances where at least one of these parameters exceeds three. Each category is evaluated under three noise regimes: low, medium, and high. Table 4.1 summarizes the ranges of (m, n, k) used to generate all instances.

Dataset Group	m **(samples)**	n **(dimension)**	k **(hyperplanes)**
`LowDim` (all noise levels)	10–22	2–3	2–3
`HighDim` (all noise levels)	10–14	2–5	2–5

TABLE 4.1: Ranges of (m, n, k) parameters used for the `LowDim` and `HighDim` datasets.

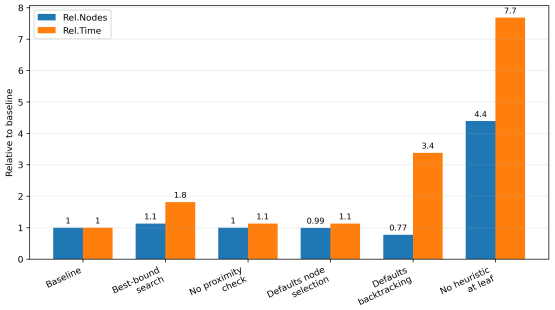


FIGURE 4.6: Comparison of IPA-SBB configurations averaged over ten k-HC instances. "Rel.Nodes" and "Rel.Time" are measured relative to the baseline configuration.

4.4.1 Parameter Configuration

To determine an effective configuration for the IPA-SBB algorithm, we conduct a parameter-sensitivity study on ten representative k-HC instances.

The *Baseline* configuration uses a *best-first search* strategy for node selection, avoids generating nodes whose affine facet F contains the origin, and incorporates the *closed-form heuristic at leaf nodes* proposed by Bradley and Mangasarian (2000). We compare this configuration against five variants: (i) best-bound search, (ii) no proximity check (i.e., nodes are created even if $\text{aff}(F)$ contains the origin), (iii) default node-selection rules, (iv) default backtracking strategy, and (v) no heuristic refinement at leaf nodes.

The results displayed in Figure 4.6 show that the *Baseline* configuration offers the strongest overall performance, solving all ten test instances with an average runtime of 145.6 s. Among all algorithmic choices, the *heuristic refinement at leaf nodes* is the most influential: disabling it increases the node count by a factor of 4.4 and the runtime by a factor of 7.7, and leads to two unsolved instances. This demonstrates the importance of generating high-quality feasible solutions to enable effective pruning.

Search strategy also plays a critical role. Replacing best-first search with *best-bound* substantially increases computation time (by 81%) and results in one unsolved instance. Similarly, disabling *backtracking* reduces the node count but more than triples runtime (Rel.Time $= 3.38$), indicating that bound revision is essential for efficiency. In contrast, omitting the *proximity check* or using *default node selection* only modestly degrades performance. Based on these observations, we adopt the baseline configuration for all subsequent experiments.

4.4.2 Experiments on Generated Instances

Table 4.2 summarizes the comparative performance of the IPA-SBB framework and the default Xpress solver across all generated datasets. Appendix B report, for every

instance, the average performance and corresponding standard deviation over ten independent runs.

Dataset	Noise	Solved		Geomean (IPA–SBB)	
		Xpress	IPA–SBB	Nodes	Time (s)
`LowDim`	Low	2 / 16	13 / 16	8.97E+04	34.6
	Medium	3 / 16	14 / 16	5.31E+04	22.0
	High	3 / 16	14 / 16	3.74E+04	18.3
`HighDim`	Low	0 / 21	19 / 21	2.45E+05	87.7
	Medium	0 / 21	20 / 21	1.24E+05	43.6
	High	0 / 21	21 / 21	6.54E+04	27.4
Overall		8 / 111	101 / 111	–	–

TABLE 4.2: Summary of computational performance across all datasets. The last row reports the overall solve rate and geometric–mean values of nodes and computation time for instances solved by both methods. Appendix B provides averages and standard deviations over ten runs per instance.

For the `LowDim` datasets, IPA-SBB reaches optimality in 13 of the 16 instances across the noise levels. Only one instance, $(m, n, k) = (18, 3, 3)$, fails to solve under low noise but solves successfully under medium and high noise. The performance improves consistently with increasing noise: the geometric mean of explored nodes decreases from 8.97×10^4 at low noise to 3.74×10^4 at high noise, and the average solution time drops from 34.6 s to 18.3 s. These results are due to the fact that when there is more noise, there are many possible optimal solutions within the given tolerance level that could improve the bound and so prune more, making the search easier.

For the `HighDim` datasets, IPA-SBB solves 19 of 21 instances at low noise, 20 at medium noise, and all instances can be solved at high noise. As in the low-dimensional case, the solve rate and geometric means improve monotonically with noise: time decreases from 87.7 s at low noise to 27.4 s at high noise, and nodes decrease from 2.45×10^5 to 6.54×10^4. This behavior again suggests that moderate perturbations simplify the geometry of the disjunctions, making the search more efficient.

Across all 111 generated instances, *IPA-SBB solves 101 (91%)* within the time limit, whereas the default Xpress solver solves only *8 (7%)*. All solvable Xpress cases correspond to instances with low dimension and a low number of hyperplanes, $(n, k) = (2, 2)$.

On the eight instances solved by both methods, IPA-SBB achieves a *geometric–mean speedup of 21×* and *uses roughly 28× fewer nodes*. When all instances are included—including those where Xpress reaches the time limit—the overall node ratio corresponds to an *approximate 123×* reduction in explored nodes and the reduction of computation time of 72×. This estimate is conservative, since Xpress may require substantially more nodes to close the optimality gap than the time-limited value recorded.

By dataset group, IPA-SBB solves 41 of 48 `LowDim` instances (85%) and 60 of 63 `HighDim` instances (95%), while Xpress solves only 8 of 48 (17%) `LowDim` instances and none of the `HighDim` instances.

Appendix B shows that, for most instances, the standard deviations of nodes and time remain within roughly 5–10% of their means, indicating stable performance across the ten independent runs. Variability is noticeably higher in the low-noise setting, particularly for difficult `HighDim` cases. In contrast, medium- and high-noise instances exhibit much smaller deviations, reflecting the stabilizing effect of noise on the geometry of the problem.

A closer look at Appendix B reveals that the relative influence of dimensionality and the number of hyperplanes is not uniform across the datasets. In the `LowDim` instances, increasing the dimension from $n = 2$ to $n = 3$ can have a stronger impact on difficulty than increasing the number of hyperplanes from $k = 2$ to $k = 3$. For example, instances of the form $(m, 3, 2)$ typically require more nodes and longer runtimes than their $(m, 2, 3)$ counterparts. In the `HighDim` datasets, however, the pattern shifts markedly. Once n exceeds three, increases in k become the predominant source of computational difficulty. Instances with $k \geq 4$ exhibit sharp growth in both mean runtime and standard deviation, even when n remains fixed. This transition reflects the fact that the size of the root partition, which scales as $(n + 1)^k$, grows explosively with k in higher dimensions. Consequently, while dimensionality can dominate in the low-dimensional regime, the number of hyperplanes becomes the primary driver of complexity as the problem moves into higher dimensions.

4.4.3 Experiments on Vision Instances

In this subsection, we compare the performance of IPA-SBB and the Xpress defaults on instances derived from two-dimensional images. For each source image, we apply an edge-detection routine to extract boundary pixels, from which m sample points are generated. Three instances are created per image from Figure 4.7 by varying the number of sampled points, and each instance is solved under a 3,600-second time limit.

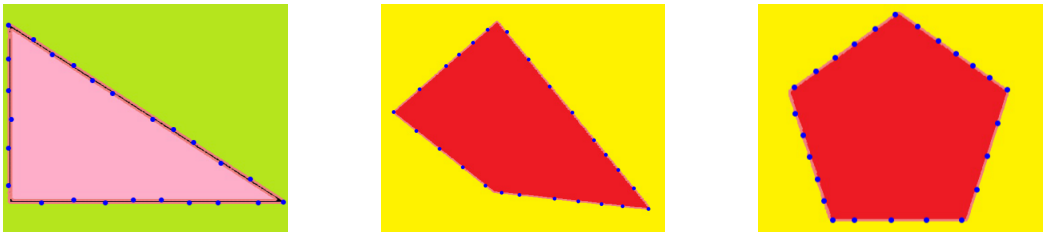
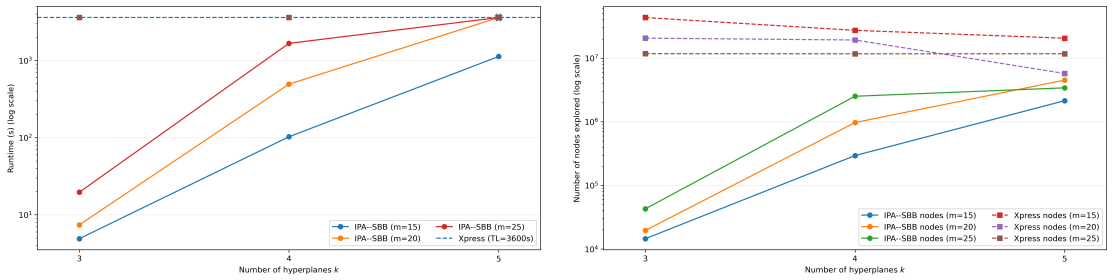


FIGURE 4.7: Examples of original images and their edge-detection points for three, four, and five hyperplanes, respectively.



(A) Runtime (TL at 3600 s). (B) Nodes explored.

FIGURE 4.8: Comparison of the Xpress default solver and IPA-SBB on two-dimensional edge-detection instances, solved with a 3,600-second time limit. Three instances are generated per source image by sampling m points from detected edges.

Figure 4.8 shows that problem difficulty increases substantially with both the number of sampled points m and the number of hyperplanes k. Even in two dimensions, the number of explored nodes and the solution time grow rapidly as k increases. For example, when $m = 20$, increasing the number of fitted hyperplanes from $k = 4$ to $k = 5$ pushes the runtime beyond the 3,600-second limit, indicating a sharp rise in search complexity. The baseline solver could not solve any of these instances within the time limit.

A key observation is that the computational bottleneck arises primarily from the value of k, rather than the number of sampled points alone. This sensitivity stems from the exponential growth of the root node layer, which scales as $(n+1)^k$. As a result, even moderate increases in k result in a significantly larger branch-and-bound tree. In practice, instances with $k \geq 6$ occasionally trigger memory exhaustion during initialization, causing Xpress to terminate before the search process can fully start.

Overall, these results highlight that, while IPA-SBB handles two-dimensional geometric data effectively for small k, the number of fitted hyperplanes remains the dominant factor limiting scalability.

4.5 Application of IPA–SBB to Linear Classification

In this section, we demonstrate the applicability of the IPA-SBB framework to a classical machine–learning task: two-class linear classification with an ℓ_2-norm constraint.

4.5.1 Problem Formulation

Consider two finite point sets $\mathcal{A}$ and $\mathcal{B}$ in $\mathbb{R}^n$, represented respectively by matrices $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{k\times n}$. The objective is to construct a hyperplane

$P = \{x \in \mathbb{R}^n : w^\top x = \gamma\}$ that separates the two classes so that each point lying on the misclassified side of the hyperplane is penalized by its distance to P. Following the formulation of Mangasarian (1999), the problem seeks a unit-norm normal vector w and a bias term γ that minimize the sum of distances of misclassified points:

$$\min_{(w,\gamma)\in\mathbb{R}^{n+1}} \left\{ \left\| \begin{bmatrix} -Aw + e\gamma \\ Bw - e\gamma \end{bmatrix}_+ \right\|_1 \text{ s.t. } \|w\|_2 = 1 \right\}, \tag{4.36}$$

where $[x]_+ = \max\{0, x\}$ is applied elementwise. A more explicit reformulation, given in Audet et al. (2008), introduces nonnegative slack variables y and z to measure misclassification distances:

$$\min_{w,\gamma,y,z} \quad \sum_{i=1}^{m} y_i + \sum_{j=1}^{k} z_j \tag{4.37a}$$

$$\text{s.t.} \quad y_i \geq -w^\top A_i + \gamma \qquad \forall i = 1, \ldots, m, \tag{4.37b}$$

$$z_j \geq \; w^\top B_j - \gamma \qquad \forall j = 1, \ldots, k, \tag{4.37c}$$

$$\|w\|_2 = 1, \tag{4.37d}$$

$$y \geq 0, \; z \geq 0. \tag{4.37e}$$

Here, y_i is the distance of point A_i to the hyperplane whenever it is misclassified (and zero otherwise); z_j plays the analogous role for points in $\mathcal{B}$. This formulation is an exact reformulation of the original classification objective, and the unit ℓ_2-norm constraint ensures scale invariance of the separating hyperplane.

4.5.2 Experiments on Linear Classification Problems

To assess the generality of IPA-SBB beyond k-HC, we evaluate it on two-class classification tasks using numerical datasets from the UCI Machine Learning Repository (Kelly et al., 2025). For each dataset, only numerical features are retained to ensure compatibility with the linear model. Table 4.3 summarizes the number of features (n), the number of samples (m), and the performance of both the Xpress default solver and IPA-SBB.

The IPA-SBB method performs favorably on low-dimensional problems with large sample sizes—such as `rice`, `occupancy_detection`, and `skin_segmentation`—where it either outperforms or matches Xpress in both nodes explored and runtime. Performance deteriorates as the number of features increases. Datasets such as `pima_diabetes`, `spambase`, and `online_shoppers`, which exhibit high dimensionality relative to the number of samples, cause IPA-SBB to explore substantially larger search trees and frequently reach the time limit. Overall, these classification experiments show that IPA-SBB remains an effective approach for large-scale, low-dimensional

Dataset	n	m	Xpress Defaults				IPA–SBB			
			Obj	Nds	Time (s)	Acc (%)	Obj	Nds	Time (s)	Acc (%)
pima_diabetes	32	198	34.0	5,367	1.6	70.7	34.5	5,963	TL	69.2
breast_cancer_wisconsin	8	699	11.4	1,907	0.7	96.7	11.4	3,674,754	TL	96.3
blood_transfusion	30	569	1.8	275	1.1	98.1	1.8	8,813	TL	98.1
spambase	44	267	92.8	66,317	10.9	59.2	92.8	1,624	TL	59.2
rice	7	3,810	1,419.2	15	6.1	52.0	1,419.2	2,949	4.1	52.0
magic_gamma	10	19,020	1,481.2	1,977	247.2	74.1	1,496.6	36,782	TL	73.4
occupancy_detection	5	20,560	109.0	15	13.9	98.3	109.0	292	2.8	98.3
adult	6	37,155	5,625.2	104	247.0	65.2	5,625.2	52,370	1,191.3	65.2
bank_marketing	7	45,211	5,186.7	10,009	932.2	86.9	5,213.8	21,594	TL	86.8
skin_segmentation	3	245,057	96,336.2	3	124.7	60.2	96,336.2	20	2.1	60.2
online_shoppers	14	12,330	881.9	339	62.6	88.4	884.5	120,446	TL	88.5

TABLE 4.3: Computational results for two-class linear classification on selected UCI datasets. Here, n is the number of features, m is the number of samples, "Obj" denotes the objective value, "Nds" the number of explored nodes, "Time" the total runtime, and "Acc (%)" the classification accuracy. TL indicates a time limit of 3,600 seconds.

datasets, while also highlighting the tradeoffs between dimensionality, geometric structure, and computational effort inherent in exact nonconvex classification models.

To evaluate separation capacity, we recorded the number of correctly labeled points and converted these values into an accuracy percentage. For both methods, the accuracy across all instances is relatively similar, suggesting that both approaches identified high-quality feasible solutions. Specifically, the Xpress solver reached proven optimality for all UCI instances.

4.6 Conclusion

The present work proposes the *iterative polyhedron approximation with spatial branch-and-bound (IPA-SBB)* framework for addressing optimization problems that incorporate unit ℓ_2-norm constraints, specifically focusing on the k-HC problem. The approach systematically improves polyhedral relaxations of the nonconvex feasible region and integrates them into a spatial branch-and-bound framework, resulting in a globally convergent algorithm underpinned by valid bounding guarantees.

Comprehensive computational experiments indicate that IPA-SBB consistently outperforms the default configuration of the Xpress solver in terms of performance and robustness. The framework consistently delivers substantial reductions in explored nodes and computation time, often resulting in speedups by an order of magnitude or greater. The results demonstrate that the method excels in low-dimensional contexts and moderate noise conditions, while difficulties with scalability emerge primarily with an increase in the number of fitted hyperplanes (k), reflecting the exponential growth in branching complexity.

In addition to the k-HC problem, we employed IPA-SBB for two-class linear classification problems, demonstrating its adaptability in handling a wider range of nonconvex optimization models with unit-norm constraints. In a few benchmark datasets, IPA-SBB either matches or surpasses the performance of Xpress, validating its potential as a versatile framework for exact geometric and machine-learning applications.

Overall, the proposed IPA-SBB method provides a principled, adaptable, and computationally efficient approach for solving a wide range of nonconvex problems with quadratic normalization constraints. Possible future work includes studying a larger number of points by first employing down-sampling methods, such as coresets, to reduce the dataset. These representative points can then be solved exactly using the IPA-SBB method.

Chapter 5

Optimizing Household Waste Recycling Centre Network Rationalization in Hampshire

In the previous chapter, we considered the k-hyperplane clustering problem, which features an MIQCQP structure, and provided a global optimization solution method. In this chapter, we examine a similar model with an MIQP structure, applied to the real-world application of rationalizing consolidation networks. Local councils across the UK are facing sustained financial pressures, and HWRCs are being increasingly considered for closure to reduce expenditure. In 2024, Hampshire County Council—which operates the largest HWRC network in the UK—proposed closing either five or twelve existing sites, representing reductions of ≈£0.5 million and £1.6 million per year, respectively, to meet medium-term budget targets. We analyze these proposals using a quadratic assignment framework originally developed for Bavaria, applying it to Hampshire's network with postcode-level population, travel-pattern, and capacity estimates. Across three closure scenarios considered by the Council, the optimization identifies closure sets that differ by only one facility, yet yield substantially improved performance: peak utilization falls by 40–60 percentage points, overloads are eliminated, and 90% of residents remain within 6.2 miles (five closures) or 9.6 miles (twelve closures) of an HWRC. Notably, the optimal closure sets form a strictly nested sequence across all feasible closure counts, enabling a stable multi-year closure path without policy reversals. We also find that capacity constraints play a structural role: relaxing them produces extreme overloads at a few urban sites, whereas enforcing them distributes demand more evenly with only marginal increases in resident travel distances. Finally, we identify three regimes of network behavior—low-impact (0–4 closures), manageable (5–12), and fragile (beyond 12)—quantifying when resident access and facility utilization remain resilient and when the network becomes structurally unstable. Overall, our results demonstrate

how established optimization methods can provide rigorous, data-driven support for ongoing HWRC rationalization under UK-wide fiscal pressure.

Note: a preprint of this work is available at
https://www.researchgate.net/publication/398638906_Optimizing_Household_Waste_Recycling_Centre_Network_Rationalization_in_Hampshire.

5.1 Introduction

5.1.1 Background

In 2019, the United Kingdom became the first major economy to legislate for Net Zero greenhouse gas emissions by 2050[1]. This commitment accelerated a structural shift in national waste policy, moving from land-filling towards circular resource management and placing greater emphasis on reuse, repair, and recycling (Department for Environment, Food & Rural Affairs, 2024). Achieving this transition depends on a resilient infrastructure of public-access facilities capable of handling bulky, hazardous, and recyclable household materials that cannot regularly be processed through curbside collection. Within this system, *Household Waste Recycling Centres (HWRCs)* serve as critical service nodes. They divert significant waste mass from landfill and incineration, delivering measurable environmental benefits. Such a recycling system is crucially dependent on the public separating recyclable and general waste at home and *bringing* the latter to the HWRCs (Williams and Taylor, 2004). Thus, the importance of HWRCs is well-beyond their function; e.g., they also serve as a tangible facility of local environmental commitment and a symbolism of public participation in recycling programs (Martin et al., 2006).

Despite their value, HWRCs are costly to maintain and operate. They rely on two schemes that have been under sustained financial pressure in the last decade: local taxation and central government grants. Between 2010 and 2020, English councils experienced a 55% reduction in central government funding[2]. Under these fiscal pressures, recycling centers—capital- and labor-intensive as well as politically sensitive—have repeatedly been targeted. Thus, rationalization in local savings plans has resulted in closures, reduced opening hours, and consolidation to statutory minimum provision.

Recent examples illustrate a nationwide pattern. In 2024, Bradford Council closed three HWRCs—Ford Hill (Queensbury), Sugden End (Keighley), and Golden Butts (Ilkley)—projecting annual savings of ≈£900,000[3]. Cheshire East Council shut three facilities as part of an ≈£100 million, four-year spending reduction program[4], and Cornwall Council proposed reducing service levels at nine of its 14 HWRCs to save ≈£200,000[5]. Kirklees Council closed its Nab Lane HWRC in Birstall to save ≈£200,000

[1] See: https://www.gov.uk/government/news/uk-becomes-first-major-economy-to-pass-net-zero-emissions-law.
[2] See: https://ifs.org.uk/publications/what-outlook-english-councils-funding.
[3] See: https://www.bbc.co.uk/news/uk-england-bradford-west-yorkshire-68628611.
[4] See: https://www.alderleyedge.com/news/article/23991/three-waste-recycling-centres-could-close-permanently.
[5] See: https://www.bbc.co.uk/news/articles/cn4zed9w4wko.

over two years[6] while Reigate & Banstead Borough Council permanently closed the Stubbs Lane HWRC in Kingswood in 2025 after recurrent fly-tipping incidents[7]. These decisions are often financially rational yet socially contentious, reflecting tension between fiscal sustainability and equitable access to environmental services; see Maynard and Cherrett (2006) for an overview of the UK's HWRC network and regulatory context.

As of 2024, Hampshire County Council (HCC) operated the largest HWRC network in England with 24 facilities, complemented by two city-operated sites in Southampton and Portsmouth; see Figure 5.4a. Many of these sites date from the 1990s and were located to balance population coverage, road accessibility, and operational capacity. This network serves over 1.4 million residents and handled approximately 120,000 tonnes of waste across more than two million bookings in 2022–23[8]. Thus, HWRCs function both as *service nodes* (by being public access points) and *load nodes* (by being points of concentrated waste material flow).

By 2023, escalating fiscal pressures led HCC to launch a county-wide *Future Services Consultation* to identify areas of cost reductions. Of the ≈£132 million sought by April 2025 in total savings, at least £1.2 million were expected from reducing HWRC operations (Hampshire County Council, 2024). However, closing even a single site alters flows across the network: residents shift to alternative HWRCs thereby increasing travel distances and raising utilization at remaining sites. Improper closure choices risk violating national guidelines requiring facilities to be "reasonably accessible" (Hampshire County Council, 2024). HCC's interpretation of "reasonably accessible", consistent with benchmarks from the Waste and Resources Action Programme (WRAP), defines accessibility as at least one site within 10 miles for most residents. WRAP further recommends that households be located within seven miles (≈20–30 minutes by car) of a facility and that no HWRC should serve more than 120,000 residents or 17,250 tonnes of waste annually. These benchmarks build on the 2004 *National Assessment of Civic Amenity Sites*, a seminal report that shaped modern UK HWRC policy.

5.1.2 Related Work

With this background, we view Hampshire's challenge as a facility-network *rationalization* problem within the class of undesirable or obnoxious facility location. These problems cater to facilities that serve essential functions (e.g, waste centers, treatment plants, or airports) while also generating negative externalities (e.g., traffic

[6]See: https://kirkleestogether.co.uk/2024/10/24/upcoming-changes-to-household-waste-recycling-centres/.
[7]See: https://www.circularonline.co.uk/news/council-closes-recycling-centre-after-persistent-fly-tipping/.
[8]See: https://www.hants.gov.uk/wasteandrecycling/recyclingcentres.

or emissions). Classical work in this field (see, e.g., Erkut and Neuman (1989)) highlights the central trade-off between residential accessibility and the burdens imposed on host communities surrounding the facilities. Modern research extends this framework by recognizing that accessibility depends not only on distance but also on population geography, commuting patterns, and spatial interaction effects, see, e.g., Salze et al. (2011). Parallel literature on public-service facility siting (e.g., hospitals, schools, emergency services) emphasize the importance of balancing system utilization with equitable access; these ideas have motivated balanced approaches that extend beyond coverage maximization alone (Xu et al., 2023). Thus, these different strands underscore that rationalization is not just a cost-minimization exercise; instead, fairness, load distribution, and public behavioral responses are also relevant.

Optimization-based methods are also being increasingly studied within the waste-management community. A substantial body of work addresses strategic planning of waste management infrastructure under uncertainty. For instance, robust municipal solid waste (MSW) network designs account for uncertain amounts of waste generation while minimizing economic, environmental, and social impacts (Yousefloo et al., 2023). Robust optimization has also been used to determine sorting-center construction and operational allocations under uncertain recycling rates and facility performance in Baoding, China (Wang et al., 2024). Interval-based location models similarly handle parametric uncertainty when identifying optimal transfer-station sites (Yadav et al., 2017). A stochastic extension for determining recycling-center capacities and locations under uncertain demand for recycled materials has also been developed (Xie et al., 2024). At the operational level, stochastic planning arises in multi-stage waste-collection models addressed through rolling-horizon heuristics to navigate uncertain waste-generation patterns (Spinelli et al., 2025), and in multi-period inventory-routing strategies for recyclable-material collection under uncertain accretion rates (Elbek and Wøhlk, 2016). Fuzzy and credibility-based formulations enhance the modeling of uncertainty regarding municipal waste volumes and service-time constraints within bi-objective vehicle routing frameworks as well (Aliahmadi et al., 2021).

Similar multi-objective location–allocation approaches aim to reduce system costs, greenhouse gas emissions, and environmental burdens in sustainable MSW systems (Yu and Solvang, 2017). There is also work on district-based facility-location models that balance economic costs, environmental impacts, and social dissatisfaction by enforcing contiguous and workload-balanced collection zones (Mostafayi Darmian et al., 2020). For hazardous or infectious waste, multi-objective formulations collectively reduce transport risk, cost, and environmental impacts when identifying disposal or transshipment facilities and planning transport routes, see, e.g., Wichapa and Khokhajaikiat (2017a,b); Kailomsom and Khompatraporn (2023); Wang et al. (2021). Location-assignment-routing models have been proposed for

agricultural-waste collection systems, determining optimal storage sites and transport routes (Tran et al., 2024). Further, there is work on locating depots or disposal sites while defining operational routes, such as the design of a depot plus three disposal-site types (compost, disposal, and recycling) in a London borough (Hirbod et al., 2023). In more recent work, exact branch-and-price methods for jointly determining transfer-station locations, waste allocation, and vehicle routes in household waste-collection systems are proposed (Han et al., 2024).

There are also empirical studies that complement the above-mentioned modeling works by examining actual HWRC performance. Nationwide studies of Danish HWRCs show that the amount of waste delivered and the amount that can be recycled vary greatly over time and across different parts of the country due to user behavior, seasonal patterns, and demographic factors (Edjabou et al., 2019). Within the UK, a case study from Sussex, UK examines user behavior, waste-stream composition, and congestion, suggesting implementable enhancements—such as restructured traffic flows, modified site layouts, and satellite green-waste depots—to improve operational efficiency (Woodard et al., 2004). Aligning with WRAP's broader efforts to support equitable, practicable recycling services in the UK, there have been other proposals for performance-indicator frameworks that build on the now-retired BVPI-91 access indicator (Harder et al., 2008).

Three works, all from Sheffield in South Yorkshire in the UK, are especially relevant to ours. In Sgalambro et al. (2025), the authors study alternate configurations of the existing HWRC network to minimize overall costs and maximize service quality. In Zaharudin et al. (2021), a spatial interaction model is developed that determines how residents choose their go-to recycling centers. In Zaharudin et al. (2022), a multi-period optimization model is introduced to provide a new operating schedule for Sheffield's HWRC network, indicating reduced evening hours and the closure of two HWRC facilities. All three studies also illustrate the financial pressure on the Sheffield City Council. In a similar spirit, our work adapts the quadratic optimization framework proposed in Schmidt and Singh (2024) that seeks to balance large accessibility and unequal burdens for recycling-center closures in Bavaria.

With this background, our work makes three key contributions as follows.

(i) We provide a data-driven optimization-based evaluation of ongoing HWRC rationalization in the UK. We show that under a 12 HWRC closure scenario in Hampshire, our model achieves substantially more balanced utilization while maintaining access within WRAP guidelines. Compared to the Council's configuration, where three sites are expected to operate over capacity with a maximum utilization of ≈153%, our proposed configuration keeps all facilities within capacity and reduces the maximum utilization by ≈61 percentage points.

(ii) We compare Hampshire County Council's proposed closure options against the model-optimal closure sets. We show that both plans differ by only one site in each scenario, yet the optimization delivers markedly better service outcomes. For instance, with five closures, the Council's plan yields two facilities operating over 100% capacity (up to ≈136%), whereas our model ensures that no site exceeds 85%. With 17 closures, the difference is even more pronounced and the Council's configuration produces extreme overloads, such as ≈220% utilization at Basingstoke and ≈160% at Southampton, while our model shows that the maximum utilization can be kept below 93%. These results demonstrate that a small structural change in the closure set can avoid severe congestion at key sites while preserving nearly identical financial savings.

(iii) We validate an existing quadratic optimization model that was originally developed for Bavaria in a new national context. Despite differing policy motivations (closures due to emissions reduction in Bavaria versus fiscal consolidation in Hampshire), the underlying model exhibits identical data-driven behavior: balanced assignments, disciplined utilization, and—importantly—a strictly nested sequence of optimal closures. Such nestedness is rare in discrete optimization and has substantial practical value. For Hampshire, the 19-step optimal sequence (Table 5.4) ensures that if budget constraints tighten over successive years, policymakers can progress through the sequence without ever needing to reverse earlier decisions.

The structure of the rest of this article is as follows. In Section 5.2, we summarize the HCC's proposed closure plans. In Section 5.3, we present our modeling framework and data estimation procedures. Section 5.4 reports our results and analysis of our model and the Council's plans, as well as a sensitivity analysis by modifying our central model. We conclude in Section 5.5.

5.2 Problem Setting

Following the 2023–24 Future Services Consultation, the HCC conducted a structured, multi-criteria assessment of all 24 HWRCs based on seven quantitative indicators:

(i) population within seven miles of each site,
(ii) tonnage handled and diversion performance,
(iii) capacity and demand balance,
(iv) distance to alternative sites,
(v) flood risk and environmental resilience,
(vi) operational performance and cost, and
(vii) site layout and traffic safety.

Based on these metrics, HCC classified the 24 HWRCs into four tiers, with Tier 1 indicating large, new, and efficient facilities and Tier 4 marking smaller facilities with lower operational value. The tiers were as follows:

(i) Tier 1: Andover, Basingstoke, Eastleigh, Gosport, Segensworth, Waterlooville, Winchester.
(ii) Tier 2: Alton, Efford, Farnborough, Havant, Netley.
(iii) Tier 3: Aldershot, Bordon, Casbrook, Hedge End, Marchwood, Petersfield, Somerley.
(iv) Tier 4: Alresford, Bishops Waltham, Fair Oak, Hartley Wintney, Hayling Island.

HCC then evaluated two closure scenarios that we refer to as Scenario I and Scenario II:

(a) Scenario I: closure of all five Tier 4 sites (≈£0.5 million annual saving), and
(b) Scenario II: closure of all twelve Tier 3 and 4 sites (≈ £1.6 million annual saving).

Both scenarios were confirmed to meet WRAP accessibility standards and were supported by detailed drive-time maps, sensitivity analyses and catchment population transfers. In addition, the HCC evaluated a more extensive configuration involving the closure of all Tier 2, Tier 3 and Tier 4 sites (i.e., 17 closures). However, this configuration was analyzed solely for impact assessment purposes, and the HCC explicitly ruled it out on the grounds that it would surpass WRAP guidance on population served and site capacity. We refer to this scenario as Scenario III.

Despite this analysis, public response was overwhelmingly negative: over 10,000 responses were received with 87% opposing any closures[9]. Residents cited increased travel times, queuing, and a potential for fly-tipping (i.e., illegal dumping) as primary concerns. Parish councils highlighted that rural communities would face disproportionate burdens, alongside risks to Hampshire's Net Zero strategy. In October 2024, HCC's Cabinet elected to retain all 24 sites and instead focus on operational efficiencies. This outcome exemplifies the tension between analytical, data-driven recommendations and political feasibility.

The above discussion a broader national pattern in the UK, where fiscal sustainability and service equity increasingly conflict (see Section 5.1). Comparable rationalization efforts outside the UK show how optimization models can inform politically constrained decisions. In Germany, for instance, residual waste generation per capita fell by nearly half between 1985 and 2020[10] contributing to a municipal recycling rate

[9]See: https://www.hants.gov.uk/aboutthecouncil/haveyoursay/consultations/future-services-consultation/hwrcs-insight-summary.
[10]See: https://www.umweltbundesamt.de/en/press/pressinformation/residual-waste-in-germany-has-nearly-halved-in-35.

of 69% in 2022 — the highest in Europe[11] Motivated by this experience, Schmidt and Singh (2024) developed a quadratic optimization framework for Bavaria that balances accessibility with equitable utilization. Even with a 20% reduction in sites, the model increased the median travel distance by only 450 m (≈0.28 miles).

Although the spatial scale and political context differ between Bavaria and Hampshire, the underlying OR challenge is similar: balancing resident accessibility and facility utilization within a mature existing network. Hampshire's higher population density and smaller geographic scale provide a stringent test of the approach under UK's constraints[12]. In the next section, we summarize this existing model and describe how we adapt it to Hampshire's geography, utilization patterns, and travel behavior.

5.3 Modeling Framework

There are three decision variables considered in the framework of Schmidt and Singh (2024): $\mathbf{x}$ denoting whether a user is assigned to a facility, $\mathbf{y}$ denoting whether a facility is selected, and $\mathbf{u}$ denoting the utilization of a facility. Below we present the optimization model reproduced from Schmidt and Singh (2024).

$$\min_{\mathbf{x},\mathbf{y},\mathbf{u}} \quad \sum_{j\in J} C_j(1-u_j)^2 \tag{5.1a}$$

$$\text{s.t.} \quad \sum_{j\in J} y_j \le B, \tag{5.1b}$$

$$u_j = \frac{\sum_{i\in I} W_{ij}\, x_{ij}}{C_j}, \quad \forall j \in J, \tag{5.1c}$$

$$\sum_{j\in J} x_{ij} = 1, \quad \forall i \in I, \tag{5.1d}$$

$$x_{ij} \le y_j, \quad \forall i \in I,\ \forall j \in J, \tag{5.1e}$$

$$\mathbf{x} \in \{0,1\}^{|I|\times|J|}, \tag{5.1f}$$

$$\mathbf{y} \in [0,1]^{|J|}, \tag{5.1g}$$

$$\mathbf{u} \in [0,1]^{|J|}. \tag{5.1h}$$

The quadratic optimization model (5.1) jointly enforces high user accessibility and balanced facility utilization. The Karush-Kuhn-Tucker optimality conditions of this non-convex model seek to achieve a notion of fairness in the optimal closures Schmitt and Singh (2024); in model (5.1) the non-convexity arises solely due to the binary decision variable $\mathbf{x}$. We consider population groups indexed by $i \in I$ (postcode areas)

[11] See: https://www.eea.europa.eu/en/analysis/indicators/waste-recycling-in-europe.

[12] Hampshire's population density is approximately 2.5 times that of Bavaria, with more than fifty times fewer HWRCs.

and facilities indexed by $j \in J$ (HWRCs). Then, the binary decision variable $y_j = 1$ indicates that facility j remains open, while $y_j = 0$ indicates closure. The assignment variable $x_{ij} = 1$ specifies that population group i is served by facility j; further, each group is assigned to exactly one open facility. Capacity constraints ensure that the total assigned demand at a facility, $\sum_{i \in I} W_{ij} x_{ij}$, does not exceed its estimated service capacity, C_j, and the budget constraint limits the number of open sites to B. Let $\mathcal{F}$ denote the feasible region of model (5.1) given by constraints (5.1b)-(5.1h).

We use population data from the UK Office for National Statistics[13] for the $|I| = 266$ resident population postcodes in Hampshire, where each postcode is classified as urban or rural. As mentioned in Section 5.1, the HCC operates $|J| = 24$ HWRCs. For each j, we have the 2022–23 throughput (in tonnes) and booking utilization rates. Following the Council's method, we compute a theoretical tonnage capacity by dividing observed tonnage by the booking utilization rate, and then convert this to a service capacity C_j by dividing by the average annual household waste generation per person (which is 0.085 tonnes). For further details on this methodology, see Hampshire County Council (2024).

When closures of public facilities occur, residents may be reassigned to more distant or less familiar sites, reducing their likelihood of using the service, see, e.g., Romero et al. (2012); Li et al. (2020). Thus, participation in recycling depends not only the presence of HWRCs but also on the "willingness" of residents to travel for recycling. To incorporate this behavioral dimension, we adopt a distance-based discounting function describing the probability that a resident undertakes a recycling trip as a decreasing function of distance. Such decaying exponential-like models are pervasive in travel-behavior modeling, see, e.g., Fotheringham (1983); Sen and Smith (1995); Iacono et al. (2008). We follow a similar approach by calibrating a willingness-to-travel model using NTS9916d records from the UK's 2023 National Travel Survey[14]. This survey comprises 1.79 million trips classified by urban/rural residence across eight distance bins: $[0,1)$, $[1,2)$, $[2,5)$, $[5,10)$, $[10,25)$, $[25,50)$, $[50,100)$, $[100,\infty)$ miles. Among these, we filter for trips made by vehicle alone. For each bin, we compute a tail (or, survival) probability as the fraction of trips with distance greater than or equal to the bin's lower bound. As in Schmidt and Singh (2024), we fit $P(d) = a \exp(-b\, d^c)$ separately for urban and rural populations; both achieve $R^2 > 0.99$. We then obtain the following willingness-to-recycle models:

$$y_{\text{urban}}(x) = 1.02 \exp\big(-0.113\, x^{1.01}\big), \tag{5.2a}$$

$$y_{\text{rural}}(x) = 1.02 \exp\big(-0.0762\, x^{1.19}\big). \tag{5.2b}$$

[13] See: https://www.ons.gov.uk/.
[14] See: https://www.gov.uk/government/statistics/national-travel-survey-2023.

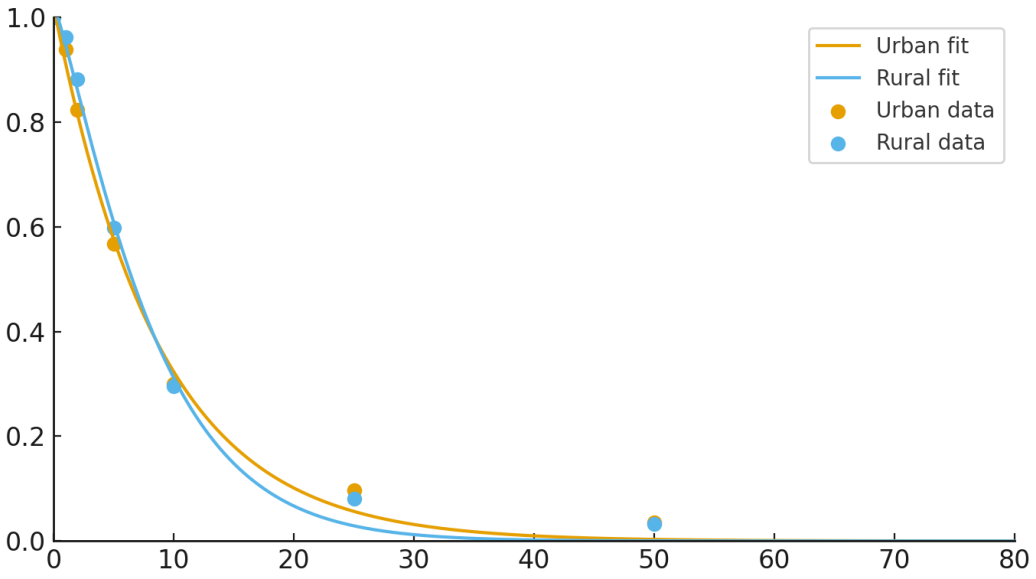


FIGURE 5.1: Empirical willingness-to-recycle fits for urban and rural respondents in Hampshire county. The y axis denotes the probability to travel from home to the closest recycling center, while the x axis denotes the distance traveled in miles. For details, see Section 5.3.

Figure 5.1 plots equation (5.2). We observe that for distances below roughly 10 miles, willingness is slightly higher among urban residents, whereas the trend reverses for longer distances. For each (i, j) pair, we then compute a discounting factor using the fitted curve corresponding to the urban/rural class of postcode i and the centroid-to-facility distance. Multiplying this factor by the population of postcode i yields the effective demand term W_{ij}.

Next, we estimate the savings associated with closing facility j. We calibrate against aggregated figures published by HCC (Hampshire County Council, 2024), which report that closing the five Tier 4 sites saves ≈£0.5 million annually and closing all Tier 3 and Tier 4 sites saves ≈£1.6 million annually, with an annual operating cost of ≈£2 million for the full 24-site network. Using per-tier throughput data, we decompose operating expenditures into:

(i) a fixed component (e.g., baseline staffing), and
(ii) a usage-dependent component proportional to throughput (i.e., tonnes of material waste handled).

We select per-tier usage weights to match the aggregated savings figures. Figure 5.2 shows the resulting cost estimates; our estimates for Scenario I and Scenario II are £521,610 (Council: ≈£500k) and £1,059,878 (Council: ≈£1.1M), respectively.

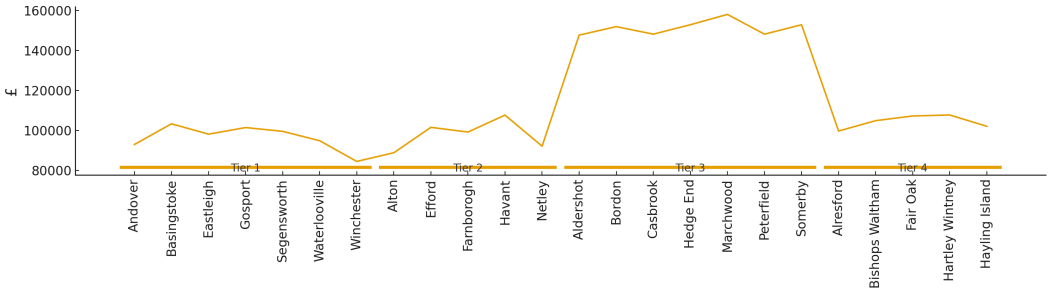


FIGURE 5.2: Estimated per-site savings by Council tier. For details, see Section 5.3.

With these data estimates, we apply model (5.1) to evaluate Hampshire's closure scenarios (see, Section 5.2) alongside several "what-if" configurations. We report results across three performance metrics:

(i) Resident accessibility: changes in percentile travel distances relative to the baseline (i.e., no closures);
(ii) Service equity: dispersion and excess utilization across open facilities; and
(iii) Financial savings: post-hoc estimates derived from operational cost data rather than included directly in the optimization objective.

5.4 Results

In this section, we compare the three closure scenarios proposed by Hampshire County Council (HCC) with the model-optimized alternatives obtained from model (5.1). We begin with Section 5.4.1 by presenting the current status quo, which serves as the baseline against which all closure scenarios are evaluated. In Section 5.4.2, we examine how the Council's proposals and the model-optimal configurations perform under identical closure counts, with particular attention to assignment patterns and resulting utilization levels. Here we also conduct a sensitivity analysis that replicates the Council's closure proposals while achieving improved accessibility through balanced assignments. We summarize practical implications for policy design in Section 5.4.3. All our computations are conducted on a laptop equipped with an Intel Core i7–13620H processor (2.40 GHz) and 32 GB RAM, running Pyomo version 6.8.2 and Gurobi version 13.0.0. Each model is solved with a time limit of 20,000 seconds and an optimality tolerance of 0.5%. All results reported in this work are globally optimal solutions which are achievable given the small scale of the model ($|I| = 266$ postcodes, $|J| = 24$ facilities).

5.4.1 Optimized versus Council Closure Scenarios

Table 5.1 summarizes our key results for this section. In the "Baseline" row, we report the resident travel distances to the existing HWRC network (10th, 50th, and 90th percentiles); the median and maximum utilization across facilities, and the number of HWRCs operating above 70% of their estimated capacity. Table 5.2 provides detailed utilization levels for each of the 26 HWRCs. This baseline—in which all 26 HWRCs remain open—serves as the reference point for the accessibility and utilization outcomes that we examine throughout this section.

The Council reports a wide utilization in the baseline, from 32.8% (Fair Oak) to 89.5% (Farnborough), with eight facilities operating above 70% of capacity. Although baseline accessibility is not reported by the Council, we compute these values using model (5.1) with $B = 26$. Then, we find travel distances are generally low: the median distance is 2.2 miles, and the 10th and 90th percentiles are 0.8 and 5.7 miles, respectively. These figures align with WRAP guidance that most residents should be within approximately seven miles of an HWRC (see Section 5.1). A small number of peripheral areas—notably those surrounding Andover, Somerley, and Efford—lie just beyond this threshold, reflecting geographic rather than structural limitations of the current HWRC network. Figure 5.4a visualizes these distances and the resulting utilization across the HWRCs.

A key distinction between the Council's baseline and the model-generated assignment is in how demand is allocated. Model (5.1) balances assignments across the Hampshire network, thereby reducing extremes in utilization. As a result, the median utilization falls from 62.6% to 58.3%, the minimum utilization rises by 7.3 percentage points (Alresford instead of Fair Oak), and the maximum drops by 10.2 percentage points (Hartley Wintney instead of Farnborough). Further, the number of HWRCs exceeding 70% utilization is reduced by half (from eight to four). Such balancing sometimes yields assignments that may appear unintuitive—for instance, residents in the BH23 1 postcode are assigned to Somerley, located 9.4 miles from the postcode centroid. Nonetheless, this contributes to a more even distribution of load.

Option	Setting	Closed HWRCs	Saving [£]	Resident Access [miles]		HWRC Burden	
				median	[10, 90] quantile	[median/maximum] [%]	$>70\%$ [#]
Baseline	Council	–	–	–	–	[62.6/89.5]	8
	Optimal	–	–	2.2	[0.8, 5.7]	[58.3/78.3]	4
Option I	Council	Alresford; Bishops Waltham; Fair Oak; Hartley Wintney; Hayling Island	522,000	–	–	[66.9/135.8]	8
	Optimal	Alresford; Bishops Waltham; Hedge End; Hartley Wintney; Hayling Island	567,000	2.4	[0.8, 6.2]	[61.9/84.8]	4
Option II	Council	Aldershot; Bordon; Casbrook; Hedge End; Marchwood; Petersfield; Somerley; Alresford; Bishops Waltham; Fair Oak; Hartley Wintney; Hayling Island;	1,581,000	–	–	[84.6/153.7]	10
	Optimal	Alton; Bordon; Casbrook; Hedge End; Marchwood; Petersfield; Somerley; Alresford; Bishops Waltham; Fair Oak; Hartley Wintney; Hayling Island;	1,523,000	2.7	[0.9, 9.6]	[74.0/92.7]	10
Option III	Council	Alton; Efford; Farnborough; Havant; Netley; Aldershot; Bordon; Casbrook; Hedge End; Marchwood; Petersfield; Somerley; Alresford; Bishops Waltham; Fair Oak; Hartley Wintney; Hayling Island	2,071,000	–	–	[101.7/219.7]	7
	Optimal	Alton; Efford; Farnborough; Waterlooville; Netley; Aldershot; Bordon; Casbrook; Hedge End; Marchwood; Petersfield; Somerley; Alresford; Bishops Waltham; Fair Oak; Hartley Wintney; Hayling Island	2,058,000	4.0	[1.1, 14.2]	[87.7/93.4]	8

TABLE 5.1: Summary comparison of HWRC closure options in Hampshire. The Council rows provide the impact on following the closure policy proposed by the Hampshire County Council, while the Optimal rows provide the analogous impact as determined optimal by model (5.1). For details, see Section 5.4.

With this background, we now evaluate the closure scenarios. In Option I (i.e., closure of Tier 4 HWRCs), our model's closure set differs from the Council's by only a single facility: the model recommends closing the HWRC in Hedge End (which is Tier 3) rather than Fair Oak. The estimated savings under this configuration are £567,000, ≈8.8% higher than the Council's estimate. The minimum and median utilizations are similar under both approaches (differences < 5 percentage points); however, the distribution of load differs substantially. For instance, the Council's configuration places eight facilities above 70% utilization, with two even exceeding their capacity (see, Table 5.2). Under the model's configuration, only four facilities (Basingstoke, Farnborough, Netley, and Aldershot) exceed 70% utilization, and none approaches its capacity limit; the maximum utilization is 84.8%. Thus, even after five closures, all remaining sites operate within feasible bounds and retain additional headroom. Accessibility also remains robust with 90% of residents being within 6.2 miles of their assigned HWRC (see, Table 5.1).

Tier	HWRC	Baseline		Option I		Option II		Option III	
		Council	Optimal	Council	Optimal	Council	Optimal	Council	Optimal
1	Andover	53.6%	48.8%	53.7%	49.8%	54.9%	55.5%	54.9%	64.1%
	Basingstoke	74.7%	77.5%	81.5%	82.7%	81.6%	87.9%	219.7%	93.4%
	Eastleigh	48.4%	54.5%	52.2%	60.1%	75.9%	77.0%	93.6%	87.8%
	Gosport	70.0%	63.6%	70.0%	67.7%	70.0%	74.0%	70.4%	86.5%
	Segensworth	71.0%	59.2%	73.6%	64.3%	86.3%	74.0%	127.7%	86.1%
	Waterlooville	69.9%	63.1%	71.9%	58.5%	86.0%	68.3%	157.1%	—
	Winchester	46.0%	50.0%	53.7%	53.8%	55.7%	67.2%	60.6%	81.2%
2	Alton	64.2%	59.2%	66.9%	61.0%	141.0%	—	—	—
	Efford	79.7%	51.1%	79.7%	51.1%	95.3%	67.1%	—	—
	Farnborough	89.5%	77.2%	117.0%	84.8%	153.7%	92.7%	—	—
	Havant	70.2%	62.3%	83.3%	63.1%	83.3%	70.5%	—	87.9%
	Netley	53.8%	57.4%	53.9%	70.6%	121.2%	81.6%	—	—
3	Aldershot	40.7%	75.2%	40.9%	80.0%	—	92.1%	—	—
	Bordon	74.9%	61.0%	74.9%	67.9%	—	—	—	—
	Casbrook	47.1%	47.2%	47.1%	47.2%	—	—	—	—
	Hedge End	71.8%	62.3%	135.8%	—	—	—	—	—
	Marchwood	59.2%	53.4%	59.2%	61.9%	—	—	—	—
	Petersfield	52.1%	44.1%	52.7%	53.6%	—	—	—	—
	Somerley	66.6%	42.6%	66.6%	42.6%	—	—	—	—
4	Alresford	39.9%	40.1%	—	—	—	—	—	—
	Bishops Waltham	47.5%	45.9%	—	—	—	—	—	—
	Fair Oak	32.8%	46.4%	—	59.5%	—	—	—	—
	Hartley Wintney	73.5%	78.4%	—	—	—	—	—	—
	Hayling Island	54.2%	49.7%	—	—	—	—	—	—
Ind.	Southampton	61.0%	59.7%	61.0%	62.7%	86.3%	80.7%	159.0%	90.0%
	Portsmouth	69.6%	66.8%	70.0%	68.4%	70.0%	72.3%	101.7%	87.9%

TABLE 5.2: Detailed comparison of HWRC utilization across the closure scenarios of Table 5.1. For details, see Section 5.4.

This difference arises from the underlying assignment mechanism that showcases the

importance of the optimization framework. The Council's approach implicitly redirects all users from a closed site to the *next-closest* facility. For instance, closing the Alresford HWRC sends all its residents to the Winchester HWRC, increasing travel distance by nearly eight miles. In contrast, model (5.1) allocated demand at the postcode level. In this example, roughly two-thirds of residents are reassigned to Winchester (median distance of 7.2 miles), while the remaining one-third are assigned to Andover of (median distance 12.7 miles). Although some residents travel farther, the load is more evenly distributed, thereby reducing congestion and potential booking delays at a single overly utilized facility. Figure 5.3 illustrates these assignment proportions and distances.

Under Option II, the model and Council closure sets again differ by only one facility: the model recommends closing the HWRC in Alton (which is Tier 2) rather than Aldershot. Table 5.1 shows that this change produces substantially more balanced utilization at only a modest reduction in expected savings (the Council's estimate is 3.7% higher). However, under the Council's plan, ten of the remaining twelve facilities operate above 70% utilization, and three exceed capacity (Alton 141.0%, Farnborough 153.7%, Netley 121.2%). Efford's utilization reaches 95.3% and Waterlooville's is at 86.0%. Under the model's configuration, ten facilities also exceed 70% utilization, however critically, *none* exceed capacity; the maximum utilization is 92.7% (Farnborough).

Table 5.2 shows the redistribution of load that produces these improvements. For instance, Farnborough's utilization falls by 61 percentage points and Efford's by 28.2. Here, Tier 1 facilities absorb some of the additional demand, e.g., Basingstoke increases by 6.3 percentage points, Winchester by 11.5, and Gosport by 4.0. Among independent facilities, Portsmouth increases by 2.3 points while Southampton's load declines by 5.6. Overall, model (5.1) trades a small reduction in financial savings for a large reduction in peak utilization. Essentially, this transforms an overly burdened network into a set of high, yet manageable, utilizations within capacity.

Travel distances under Option II rise relative to the baseline, as expected from such a large network contraction where nearly half of the sites have been removed. The median travel distance increases by 0.3 miles while the 90th percentile increases by 3.4 miles (see, Table 5.1). Figure 5.4c shows that longer distances concentrate in the eastern and western portions of the county. Although no direct comparison with the Council's distance estimates is possible, the model keeps the 90th percentile below 10 miles. Thus, despite considerable consolidation, accessibility remains broadly compatible with WRAP guidance; thus, the network still maintains a functional level of service.

The extreme scenario, Option III, tests the limits of HWRC's network contraction. Here, only nine facilities remain open, and the Council and optimized closure sets

again differ by a single site: the model retains the HWRC in Havant (Tier 2) while closing the Tier 1 facility in Waterlooville, the reverse of the Council's choice. Savings differ by less than 1% between the two configurations (both exceed £2 million annually). However, the Council's configuration produces severe overloads: the median utilization is over capacity with a maximum being over double the capacity, and seven of nine facilities exceed 70%. However, under the model's configuration, the median utilization is 87.7%, all facilities except that in Andover exceed 80% while remaining within capacity, and the maximum utilization is 93.4%. Large reductions occur at Basingstoke (219.7% to 93.4%), Segensworth (127.7% to 86.1%), Southampton (159% to 90%), and Portsmouth (101.7% to 87.9%). Again, the model spreads demand more evenly and avoids extreme overloads.

The resulting travel-distances increase markedly under this extreme contraction. For instance, the median distance reaches 4.0 miles, with 10th and 90th percentiles at 1.1 and 14.2 miles, respectively (see, Table 5.1); nearly a third of all residents travel more than 10 miles. Figure 5.4d shows pronounced distance increases across most parts of Hampshire; however, the central region still displays markedly lower travel distances than the outer boundaries. While these distances still broadly align with WRAP guidance, the combination of near-capacity utilization across the network and longer travel times suggests that such a configuration is challenging to justify in practice.

5.4.2 Sensitivity Analysis of Council-Implied Assignments

The analysis so far demonstrates that the Council's closure plans are strikingly close to the optimal selection determined by our model in all considered scenarios. However, as shown in Section 5.4.1, there is substantial variation in utilization levels. Further, a direct comparison of the corresponding travel distances between model (5.1) and the Council's proposal is not possible since the Council did not publish postcode-level travel patterns. In this section, we seek to approximate how the Council's assignment might behave. To this end, we construct three surrogate optimization models that replicate, as closely as possible, the utilization pattern implied by the Council's published figures; we let $\bar{u}_j$ denote these reported utilizations. All our surrogate models seek to largely retain the structure of the feasible region $\mathcal{F}$ and differ only in their objectives and in how the utilization bounds are enforced.

We consider the following three formulations:

1. Our first auxiliary model seeks to match the Council's utilization pattern by minimizing the squared deviation from $\bar{u}_j$. To this end, we relax the feasible region, $\mathcal{F}$, to allow $u_j > 1$ enabling us to replicate the Council's implied loads that include over-capacity operation (see, Table 5.2). We then have the following

model:

$$\min \sum_{j \in J} (u_j - \bar{u}_j)^2 \;\mid\; (\mathbf{x}, \mathbf{y}, \mathbf{u}) \in \mathcal{F}', \tag{5.3a}$$

where

$$\mathcal{F}' \;=\; \left\{ (\mathbf{x}, \mathbf{y}, \mathbf{u}) \,\middle|\, (\mathbf{x}, \mathbf{y}, \mathbf{u}) \in \mathcal{F},\; u_j \in [0, \infty)\; \forall j \in J \right\}. \tag{5.3b}$$

We refer to model (5.3) as the Utilization-Fit (`UF`) model.

2. In our second variant, we enforce the original utilization bounds and require all facilities to operate within capacity:

$$\min_{\mathbf{x},\mathbf{y},\mathbf{u}} \sum_{j \in J} (u_j - \bar{u}_j)^2 \qquad \text{s.t.} \qquad (\mathbf{x}, \mathbf{y}, \mathbf{u}) \in \mathcal{F}. \tag{5.4}$$

Model (5.4) examines how closely we may match the Council's utilization while respecting the capacity constraints. We refer to model (5.4) as the Utilization-Fit with Capacity (`UF-C`) model.

3. Our third model maximizes system-wide access (equivalently, minimizes aggregate resident travel distance) while restricting utilization to remain within a symmetric band around the Council-reported values:

$$\begin{aligned} \max_{\mathbf{x},\mathbf{y},\mathbf{u}} \quad & \sum_{i \in I} \sum_{j \in J} W_{ij} x_{ij} \\ \text{s.t.} \quad & (1 - \varepsilon)\, \bar{u}_j \;\le\; u_j \;\le\; (1 + \varepsilon)\, \bar{u}_j, \quad \forall j \in J, \\ & (\mathbf{x}, \mathbf{y}, \mathbf{u}) \in \mathcal{F}', \end{aligned} \tag{5.5}$$

where $\mathcal{F}'$ is as defined in equation (5.3b). Based on our computational experiments, we choose $\varepsilon = 0.2$ as a small enough number to ensure feasibility of model (5.5). We refer to model (5.5) as the Access-Maximizing Fit (`AMF`) model.

Models (5.3)-(5.5) also allow us to investigate the implicit assumption in Section 5.4.1 that nearby facilities can continue to absorb additional demand even after their expected capacity is reached. Table 5.3 reports the distance percentiles and utilization levels for the `UF`, `UF-C`, and `AMF` models alongside those produced by the central model (5.1).

In the Baseline and Option I scenarios, model (5.1) achieves the smallest 90th percentile distances (5.7 and 6.2 miles, respectively), while keeping all facilities below capacity, despite not optimizing directly for access. None of the auxiliary models achieves a lower 90th percentile in these scenarios: both `UF` and `UF-C` exceed 9 miles, and `AMF` remains between 7.4 and 7.9 miles; further, these latter values are close to or above the WRAP benchmark for acceptable resident travel. In Option II, model (5.1) matches `UF` at a 90th percentile of 9.6 miles and is only slightly worse than `AMF` (9.3

Option	Setting	Resident Access (miles)			HWRC Burden [median/maximum] [%]	Distance (miles)	
		p_{10}	p_{50}	p_{90}		Group A	Group B
Baseline	UF	0.8	2.6	9.3	53.9/73.9	5.2	2.3
	UF-C	0.8	2.6	9.3	54.2/73.9	5.2	2.3
	AMF	0.8	2.5	7.9	55.2/88.9	4.5	2.2
	Model (5.1)	0.8	2.2	5.7	58.3/78.3	3.4	1.9
Option I	UF	0.8	2.4	9.4	62.5/132.5	4.3	2.1
	UF-C	0.8	2.5	9.9	63.5/100.0	4.5	2.1
	AMF	0.8	2.3	7.4	58.3/130.7	3.8	2.1
	Model (5.1)	0.8	2.4	6.2	61.9/84.8	4.3	2.1
Option II	UF	0.9	2.7	9.6	75.4/149.1	4.5	2.2
	UF-C	0.9	2.7	11.2	78.9/100.0	6.2	2.3
	AMF	0.9	2.6	9.3	70.4/160.6	4.1	2.2
	Model (5.1)	0.9	2.7	9.6	74.0/92.7	4.4	2.3

TABLE 5.3: Comparison of model outcomes across optimization approaches. Resident access is measured by the 10th, 50th, and 90th percentile travel distances. HWRC burden reports median and maximum utilization (percent). Group-level distances correspond to median distances for postcode groups A and B. For details, see Section 5.4.2.

miles), however it does so with a markedly lower maximum utilization (at 92.7%, compared with 149.1% for UF and 160.6% for AMF). Thus, across all three scenarios, model (5.1) either improves on or remains close to the best achievable 90th percentile distance among the auxiliary models, while simultaneously avoiding the substantial overloads they permit.

The three auxiliary formulations help interpret the system behavior under Council-like assumptions. The UF model, that attempts to follow the Council-reported utilizations, consistently reproduces high or extreme peak loads (e.g., up to 149.1% compared to 141.0% reported by the Council in Option II, see Table 5.2), reflecting the assumption that several facilities can absorb large quantities of displaced demand. In contrast, imposing capacity limits in UF-C eliminates overloads with only modest effects on access: median travel distances remain unchanged across all settings, including Option II, and the 90th percentile distance increases by 1.6 miles only in the heaviest contraction scenario. This suggests that enforcing capacity is both a natural and practically benign constraint; we discuss this aspect below. Finally, the AMF model reduces distances slightly by allowing utilization to vary within a band around the Council's levels; however, the correspondingly necessary increases in peak load (reaching 160.6% in Option II) do not appear justified by the relatively small reductions in the 90th percentile distance.

To summarize, our comparisons comparisons validate that model (5.1) provides a balanced outcome as intended: it maintains feasible utilization levels across the Hampshire HWRC network and produces 90th percentile distances that are, either, the

best among all capacity-feasible assignments or close to the best attainable under Council-like patterns. In this sense, the performance of model (5.1) on the Hampshire HWRC network validates its use for achieving stable and publicly defensible closure solutions.

Optimizing for the entire county of Hampshire may, however, lead to spatial inequity in access. To examine this, we categorize the 266 sub-postcodes into Hampshire's six official postcode districts. A visual inspection of the HWRC network (see Figure 5.4a) indicates that most of the open facilities lie within or near the SO (Southampton) and PO (Portsmouth) postcode regions, which together form Hampshire's two largest conurbations. The remaining four postcodes—SP, GU, RG, and BH—account for approximately 35% of the relevant population and are much more spatially dispersed. Since travel distance is closely tied to population density (see, e.g., Figure 5.4a) and facility clustering (see, e.g., Levinson and Kumar (1994)), we split the postcode districts into two groups:

(i) Group A (SP, GU, RG, BH), representing the more dispersed regions, and
(ii) Group B (SO, PO), representing the more densely urban districts.

This grouping also aligns with UK's ONS classifications of rural and urban, and 85% of the population in these areas is urban. Hence, for this analysis alone, we restrict attention to the urban population in these two groups.

The last two columns of Table 5.3 show that median travel distances in Group A are consistently higher than those in Group B for all models and all scenarios; this reflects the underlying geographic distribution in Hampshire rather than any specific modeling choice. In the baseline, model (5.1) achieves the smallest median distances in both groups (as well as the smallest gap between them) suggesting that balanced assignments reduce spatial disparities even without any closures. Under Options I and II, the median distance in Group A increases by roughly one mile relative to the baseline, while Group B's travel distance remains nearly similar. Across these options, model (5.1) performs at least as well as the auxiliary models that enforce capacity (i.e., `UF-C`) and remains broadly comparable to the unrestricted `UF` model. The `AMF` produces only slightly lower travel distances for Group A and only in the two closure scenarios. Thus, model (5.1) remains competitive overall, even in the largely urban areas of Hampshire.

5.4.3 Policy Interpretation and Practical Guidance

In this section, we evaluate model (5.1) for varying closure counts with $B = 26, 25, \ldots, 7$; i.e., up to 19 closures beyond which model (5.1) becomes infeasible. We obtain a remarkably consistent pattern: the optimal closure sets are *nested*; i.e., the

optimal closure set of size k is always contained in the optimal set of size $k-1$. This structure is not imposed in our mathematical formulation, and it arises directly from both the geometry of the Hampshire network and the underlying optimization structure. In discrete optimization, such a nested structure is characteristic of submodular or supermodular sets; however, model (5.1) is, in general, neither submodular nor supermodular (Singh, 2025). For HWRC planning, this property is particularly valuable: once a facility is optimally selected for closure, it remains closed in all subsequent optimal solutions with larger closure sets. Table 5.4 summarizes the resulting sequence.

Closure Order	HWRC
1	Alresford
2	Hayling Island
3	Bishops Waltham
4	Hedge End
5	Hartley Wintney
6	Casbrook
7	Somerley
8	Bordon
9	Petersfield
10	Marchwood
11	Fair Oak
12	Alton
13	Aldershot
14	Netley
15	Farnborough
16	Efford
17	Waterlooville
18	Winchester
19	Havant

TABLE 5.4: The nested optimal sequence of HWRC closures as determined by model (5.1). A closure order of k denotes $B = 26 - k$. The k-th closure is included in the $k+1$-st closure. For details, see Section 5.4.3.

This nested structure has direct practical value in policy-making. As observed in the UK (see Section 5.1), and other European countries (Schmidt and Singh, 2024), HWRC rationalization rarely occurs in a single step; councils typically close a few facilities in a given financial year as budgets, policy guidance, and service contracts evolve. The nested property guarantees that if Hampshire implements the early closures in the sequence and further cuts become necessary later, it can simply proceed to the next closure on the list without needing to revisit or reverse prior decisions. In this sense, the sequence functions as a stable, multi-year decision pathway.

Figure 5.5 illustrates how accessibility changes as more closures occur. In the status quo with all 26 sites open, county-wide accessibility is just under 80%. This declines relatively smoothly to about 55% when 19 sites are closed (i.e., $B = 7$). The two postcode groups, however, evolve differently. Group B (SO, PO) experiences a

relatively modest decline of about 15 percentage points from 0 to 19 closures. This reflects both its dense population and the concentration of remaining HWRC capacity in the Southampton–Portsmouth region, which remains open in every optimal solution. In contrast, Group A (SP, GU, RG, BH) shows a sharper decline. Accessibility remains near 70% up to four closures, falls gradually to about 65% by twelve closures, and then drops steeply once the sequence closes the HWRC in Aldershot (closure 13) and subsequently in Farnborough (closure 15).

Figure 5.6 reports the 10th-percentile, median, and 90th-percentile travel distances for all values of B. We find that the overall median travel distance remains below seven miles for all closure counts, despite the substantial contraction of the network. In contrast, the median distance for Group A exceeds five miles after more than 12 closures and rises to approximately 7.5 miles after 13 are closed; it exceeds ten miles only after fifteen closures. The 90th percentile shows similar behavior: for up to five closures, 90% of Group A residents remain within 10 miles; between seven and twelve closures, this upper tail distance rises to about 12 miles; beyond twelve closures, the increase accelerates. This pattern highlights a typical feature of mature spatial networks: it absorbs early closures effectively, but loses resilience as contraction deepens.

These results identify practical planning ranges. For instance, the first four closures have almost no system-wide effect; then, accessibility, travel distances, and utilization patterns remain near their baseline levels. This suggests that these facilities can be removed with minimal disruption. Further, the fifth and sixth closures generate additional savings with still limited impact. Closures seven through twelve remain operationally manageable; however, inequity begins to emerge as travel distances for Groups A and B diverge. Beyond twelve closures, both accessibility and utilization indicators deteriorate rapidly (especially in Group A), suggesting that deeper rationalization is not viable. Although Hampshire is unlikely to pursue cuts of this magnitude, the analysis clarifies—through data-driven evidence—where the HWRC network becomes structurally fragile and why only the early portion of the sequence is practically relevant.

5.5 Conclusions

Our work examines Hampshire's HWRC network through a quadratic optimization framework originally developed for Bavaria. We find this framework to be both structurally transferable as well as practically informative. Despite the distinct policy motivation in Hampshire—fiscal consolidation rather than emissions mitigation—the model produces balanced assignments, feasible utilization levels, and, importantly, predictable system response to incremental closure. The challenge of managing public

services under severe financial pressure is a defining issue for local authorities across the UK, and this tension is especially acute for HWRCs, where cost-reduction pressures conflict with national ambitions to build a circular economy. As the recent Hampshire County Council consultation illustrates, even quantitatively robust proposals may fail — not only on technical grounds but also because they provoke strong public opposition rooted in concerns over fairness, accessibility, and the potential for negative environmental outcomes. In this context, our work provides a structured, data-driven alternative capable of informing strategic decisions that are both fiscally responsible and socially equitable.

Three of our findings particularly stand out. First, optimal closure sets from model (5.1) nearly match the Council's proposals differing by only one site in each considered scenario — however, model (5.1)'s closures consistently achieve more uniform utilization and more evenly distributed demand. Second, capacity constraints—requiring utilization not to exceed 100%—serve not merely as a modeling artifact but structural features of HWRC spatial networks. Enforcing them prevents over-concentration of re-assigned demand and produces realistic assignment patterns aligning with quality operational practice. Third, Hampshire exhibits an unexpectedly strong form of nested optimality: the closure sequence that minimizes system-wide imbalance is strictly monotone. This property is relatively rare in discrete optimization. In practice, it ensures policymakers can implement closures sequentially without risking reversals or inconsistent outcomes in later years.

Our analysis also highlights where the existing Hampshire HWRC network is most resilient and where it begins to strain. The network behaves almost elastically up to four HWRC closures: accessibility and utilization remain close to their baseline values, and changes are minimal for residents. Between five and twelve closures, the system remains operationally feasible but begins to exhibit spatial inequity, particularly for residents in the more dispersed postcode areas (Group A). Beyond twelve closures, the network becomes structurally fragile, with rapidly rising travel distances and reduced accessibility. These three closure bands —low-impact, manageable, and fragile—provide clear, evidence-based guidance for any future rationalization exercise.

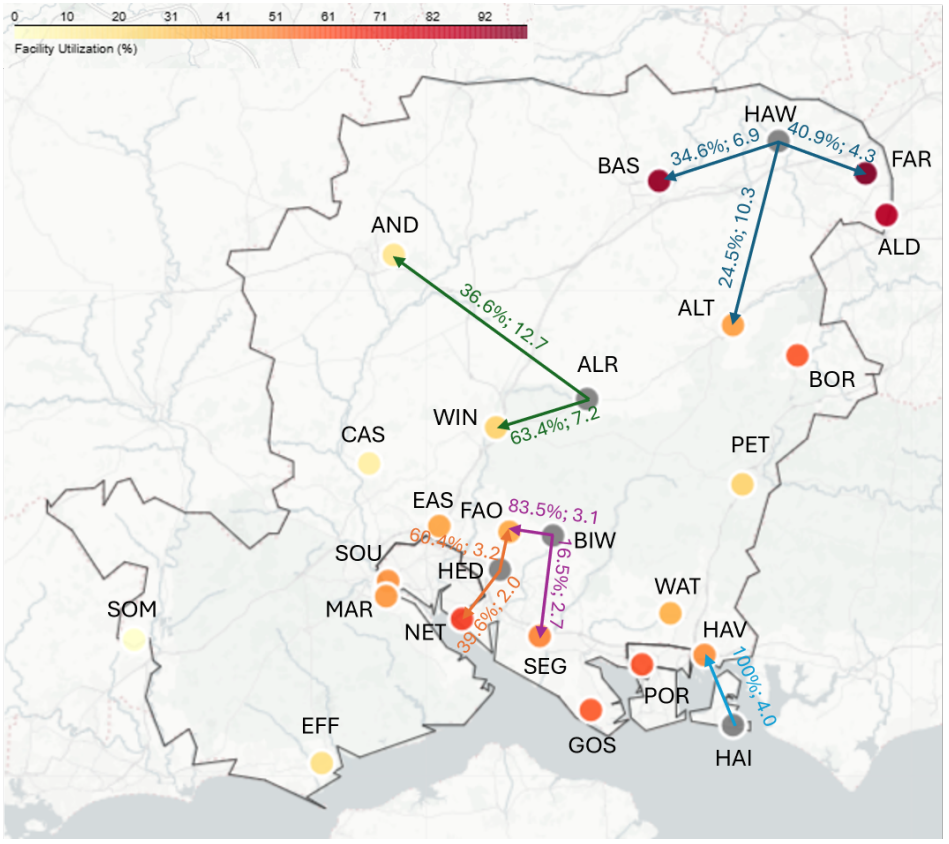
A broader implication of our results concerns the role of capacity in HWRC planning. The auxiliary models in Section 5.4.2 demonstrate that relaxing capacity constraints leads either to substantial overloads at a small number of HWRCs or to assignment patterns in which large volumes of displaced demand are absorbed by a few urban sites. In contrast, enforcing capacity yields reassignment patterns that distribute demand more evenly across the network producing travel distances only marginally different from those obtained without capacity limits. This behavior explains the transferability of model (5.1) outside the Bavarian context for which it was originally developed, and reflects the spatial structure of Hampshire's HWRC network. Thus,

capacity constraints act as a natural regularization mechanism that prevents unrealistic concentration of demand yielding solutions consistent with public-service expectations for booking management, queueing, and throughput.

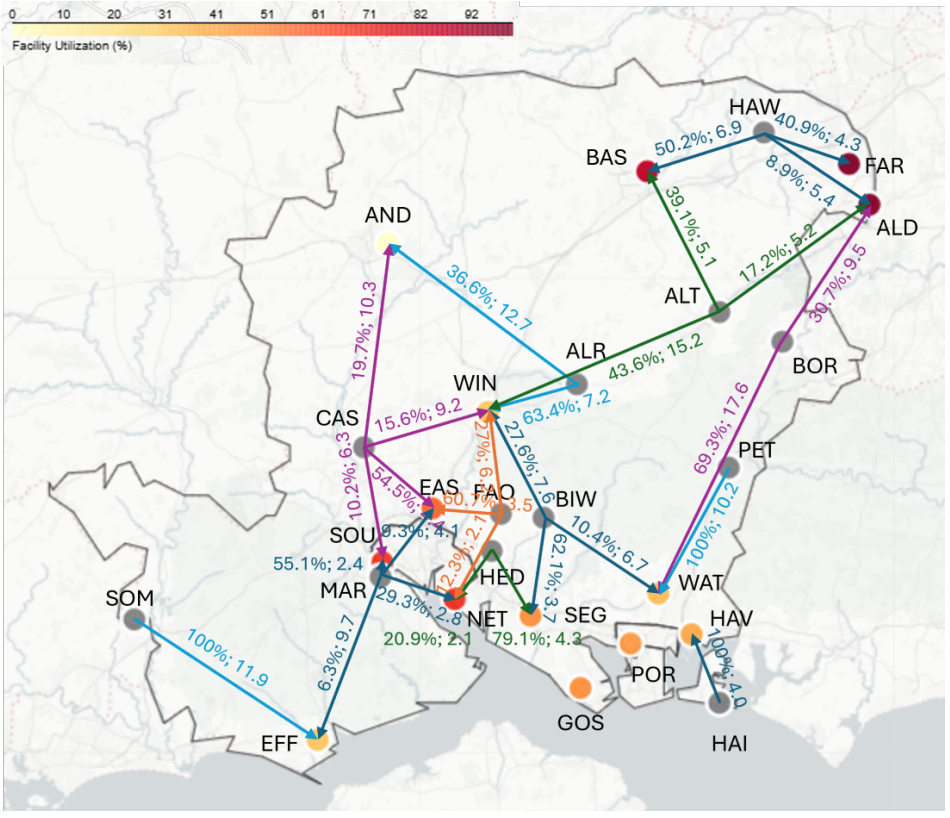
Our analysis also show that a principled optimization framework can significantly outperform intuitive or ad hoc approaches. Although the Council's modeling framework is surprisingly robust, the optimization identifies significant improvements in equity across Hampshire. Approximately a third of Hampshire's population—those in Group A postcodes—live in more dispersed areas. Under the optimized 12-closure configuration (i.e., Option II), the median travel distance for this group is nearly double that of the more urban Southampton and Portsmouth regions (i.e., Group B); see, Table 5.3. However, under a Council-like configuration that matches reported utilization while enforcing capacity (i.e., the `UF-C` model), the median travel distance for Group A nearly triples; see, Table 5.3. In contrast, model (5.1) reduces these extremes and bounds geographic disparity, even under substantial network contraction.

At a policy level, our findings argue for a shift from reactive, short-term cost-cutting towards a proactive paradigm of long-term network design. We find that the quadratic optimization framework of Schmitt and Singh (2024) is not merely an abstract mathematical construct, but a practical tool for modern governance. It provides data-driven, transparent, and publicly defensible strategies for multi-year facility planning, readily adaptable to evolving budget conditions, candidate-site evaluations, or, even, projected demographic changes. The two metrics of resident travel-distance and facility demand burden generated by the model additionally provide a principled basis for targeted Council investment, e.g., towards shrinking the population-density and infrastructure gaps within Hampshire.

As expected, several limitations apply to our work. We highlight some of these here. Our willingness-to-travel function assumes that resident recycling journeys in Hampshire resemble national travel patterns derived from the National Travel Survey (NTS). Our cost-benefit estimates rely on aggregated operational information rather than site-specific accounts. Further, representing postcode populations at their centroids is a geographic simplification. Thus, while our model offers strategic and quantitative insights, it is not a prescriptive operational tool. That being said, a framework that embeds equity and accessibility directly into a rationalization exercise enables local authorities to navigate difficult decisions without undermining public participation or compromising environmental obligations. As our results demonstrate, the UK's transition to a circular economy can indeed be supported via intelligent, data-driven and publicly defensible decision-making.

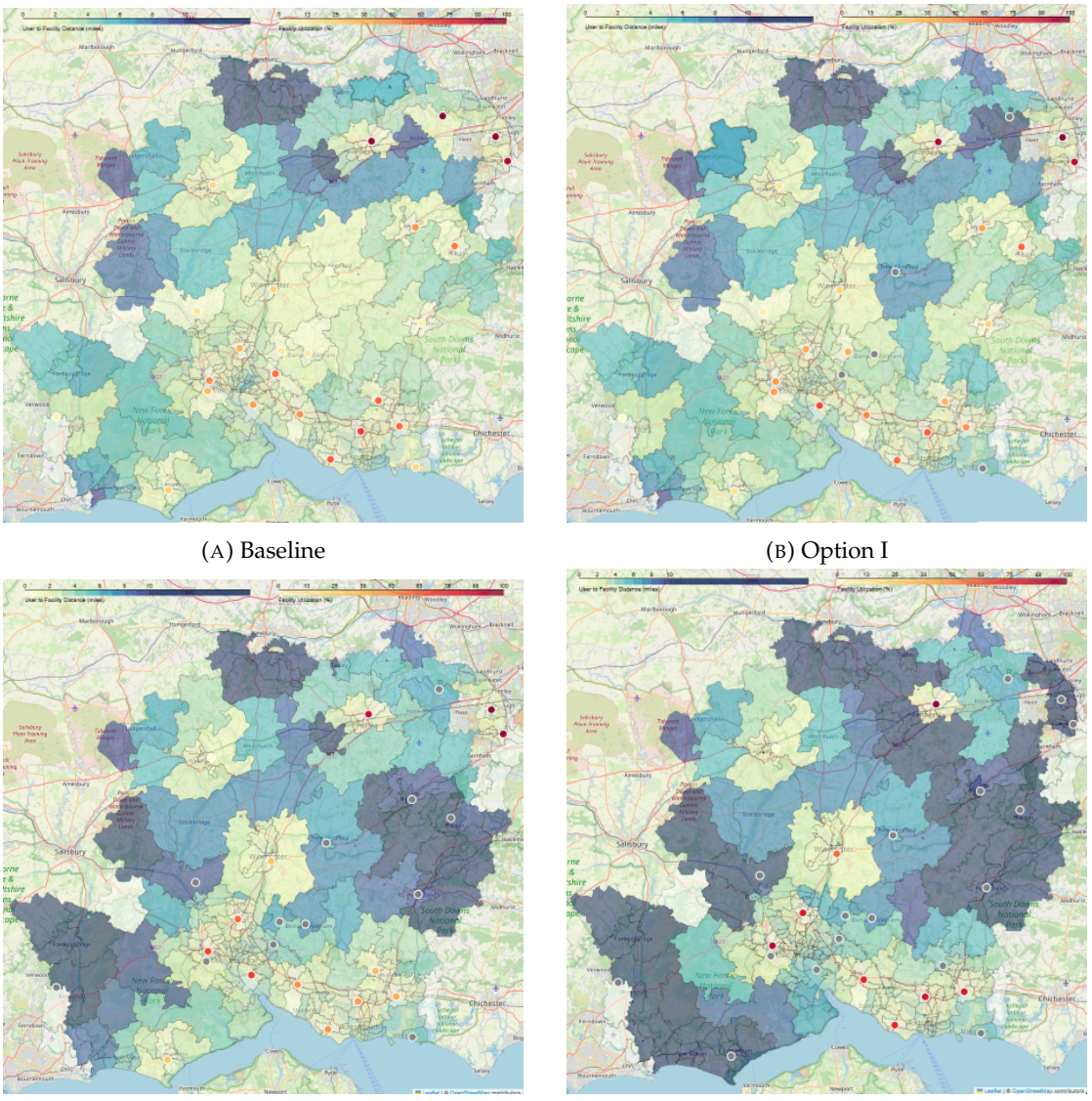


(A) Option I



(B) Option II

FIGURE 5.3: Re-assignment of closed HWRC facilities for the two closure scenarios considered by the Hampshire County Council. All facility names are represented with three characters and a circle dot (see Table 5.2 for full names). Grey indicates a closed facility, while the other colors represent utilization. The arrow labels show both the proportion of the population assigned to the new facility with the corresponding median travel distance. For details, see Section 5.4.



(A) Baseline

(B) Option I

(C) Option II

(D) Option III

FIGURE 5.4: Heatmaps showing the resulting utilization of each HWRC (circles) and the resident travel distances from each sub-region to its assigned facility. Results are based on the optimal solution of model (5.1) for the baseline and the three closure scenarios. For details, see Section 5.4.

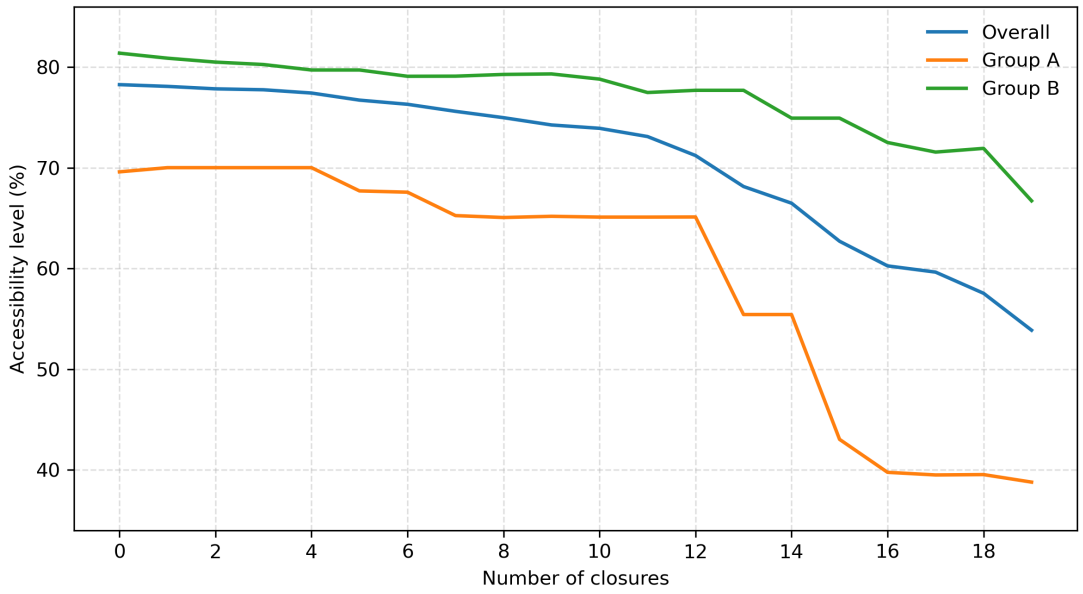


FIGURE 5.5: Resident accessibility for the entire population of Hampshire ("Overall"), Group A and B populations. The x-axis denotes the number of closures (i.e., $26 - B$) while the y-axis denotes the accessibility, $\sum_{i \in I, j \in J} W_{ij} x_{ij}$, computed from the optimal solution of model (5.1). For details, see Section 5.4.3.

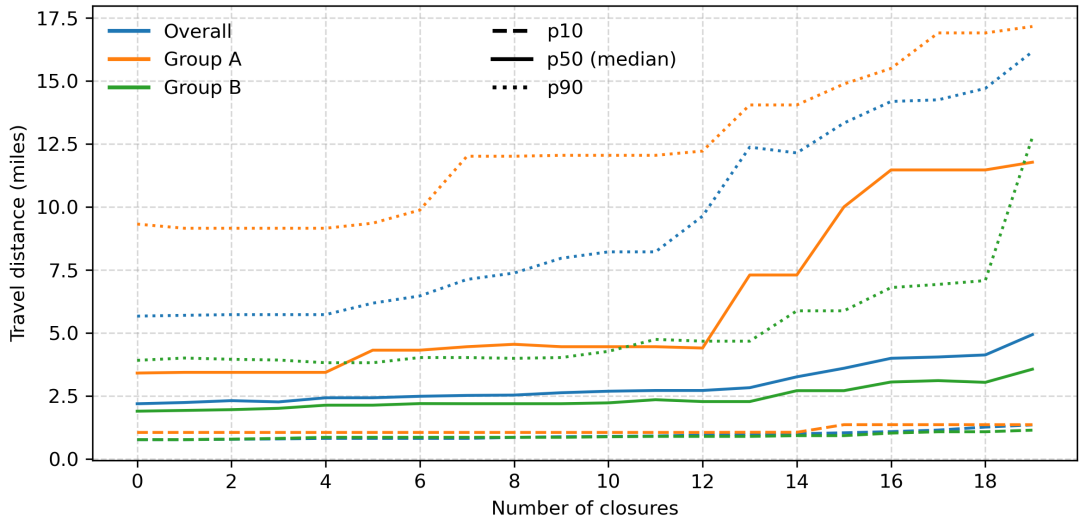


FIGURE 5.6: Quantiles of travel distances corresponding to the resident accessibility in Figure 5.5 for the entire population of Hampshire ("Overall"), Group A and B populations. For details, see Section 5.4.3.

Chapter 6

Risk-Seeking Problem Formulations and Their Approximations with Union Bounds

In this chapter, we shift our focus to the second major theme of this thesis: addressing optimization problems under uncertainty. We place particular emphasis on probabilistic constrained problems, where traditional decision-making methods typically prioritize risk-neutral and risk-averse approaches that strictly limit constraint violations. In contrast, this study explores a risk-seeking framework by utilizing probabilistic constraints with substantial violation tolerances. This approach enables decision-makers to accept constraint violations in "lucky" low-resource scenarios while exploiting "greedy" high-reward outcomes. We propose a family of approximation models based on binomial moment bounds (orders one through four) applied to the probability of the union of failure sets. We introduce a variable definition that enables the effective linear representation of higher-order joint probabilities. These approximation models are initially constructed as bilevel linear programs and subsequently reformulated into single-level mixed-integer linear programs using KKT conditions. Computational experiments on baseline and sparse datasets show that the first-order approximation is computationally efficient yet structurally imprecise, whereas the second-order approximation presents the most advantageous compromise, yielding improvements in bound tightness and attaining near-perfect structural fidelity to the true optimal solution. Additionally, we examine how resource distribution, risk appetite, and violation tolerance interact, demonstrating how a high risk appetite pushes the solution in the direction of jackpot scenarios.

6.1 Introduction

6.1.1 Motivation

Decision-making problems in existing literature are majorly formulated as risk-neutral or risk-averse mathematical optimization models where one seeks to limit losses against uncertain outcomes, see, e.g., Ruszczyński (2013). However, both anecdotal evidence and managerial reflection suggests individuals, as well as firms, often do exemplify risk-seeking tendencies (Cain and McKeon, 2012). An extreme example of such individual behavior is addictive gambling where an individual may lose large sums — possibly one's entire wealth — for the lure of considerably much smaller amounts of money. A more practical example is the choice of risky therapies in terminally ill patients, see, e.g., Rasiel et al. (2005). Investment decisions by firm executives in unexplored territories, driven by recognition and creation of new opportunities, present further evidence of risk-seeking behavior with a potential for ruin (Miller, 2007). Despite the widespread prevalence of such behavior, there are relatively limited studies in the mathematical optimization community that analyze this behavioral phenomenon. Our study is in this spirit.

The conceptual foundations of risk tolerance are based on utility theory drawing on the seminal work of Markowitz (1952) that has been extended over the years, see, e.g., Hakansson and Arrow (1972). Following such frameworks, rational decision-making choices in typical stochastic programming models involve one or a combination of maximizing expected returns and minimizing variability of returns. In contrast, Crum et al. (1981) provide evidence that decision-makers often switch between risk-averse and risk-seeking behavior depending on their outcome targets. Such behavior aligns with Kahneman and Tversky's prospect theory that demonstrates individuals tend to seek risk when attempting to recover from previous losses driven by a desire to return to an established reference point (Kahneman and Tversky, 1988). For a summary of classical risk-return associations, see (Fiegenbaum and Thomas, 1988, Table 1). These insights have influenced more modern concepts in stochastic programming, such as stochastic dominance, by incorporating both concave (risk-averse) and convex (risk-seeking) utility segments (Post and Levy, 2005).

Specifically, our work seeks to investigate risk-taking models using probabilistic constraints. These constraints find origin in work by Charnes and Cooper (1959) on chance-constrained programming models that allow a small fraction, ε, of constraint violations to support riskier decisions. This framework has since been extensively investigated with related concepts — two of the most notables ones being conditional value-at-risk (Rockafellar and Uryasev, 2000) and buffered failure probability (Rockafellar and Royset, 2010). Applications of probabilistic constraints to incorporate risk-preference span various fields. In finance, Lejeune and Shen (2016)

consider optimizing multiple portfolios together at shared varied levels of risk tolerance allowing up to 25% of unreliable outcomes. In renewable energy, Roald and Andersson (2018) apply probabilistic constraints to model AC optimal power flow, ensuring that generation, power flows, and voltage levels remain reliable with a violation probability of up to 15%. Similarly, Guo et al. (2018) propose a less-conservative coordination between energy storage units and uncertain renewable energy resources in a microgrid, accommodating up to a 10% violation. These examples serve to demonstrate how probabilistic constraints offer a framework to model risk-seeking preferences and optimize gains.

However, we are unaware of studies on probabilistic constraints that allow much lower values of ε; i.e., deliberately encourage extreme risk behavior. This observation is explainable by the fact that such probabilistic constraints are typically used to design highly-reliable systems (i.e., with a large probability, $1 - \varepsilon$, of satisfying the constraints) with catastrophic consequences of failures. Our setting is precisely the opposite — we explicitly consider this framework but with large ε values. As such, we cater to the risk-seeking decision-maker who is prone to make irrational choices that could even be ruinous. As we mention above, and we demonstrate later in this work, such behavior does find application in a variety of settings reflecting human tendency to follow suboptimal decisions that do not maximize expected returns. Before we provide concrete examples in support of such risk-seeking behavior, we define the problem setup.

6.1.2 Problem Setting

To formalize the framework of risk-seeking behavior that we study in this work, consider a discrete planning horizon of $|T|$ periods indexed by $t \in \{1, 2, \ldots, T\}$. For sake of explanation, consider a resource allocation problem that seeks to maximize returns (other problems extend from this setting). In each period t, the decision-maker selects an amount of resources $x_t \geq 0$ that provides a benefit $g(x_t) \geq 0$. However, the available resources, denoted by $w_t^\omega \geq 0$, are uncertain and depend on a random scenario $\omega \in \Omega$. After the realization of the scenario, the decision-maker may be forced to amend large out-of-reach choices of x with a correction, $y_t^\omega \geq 0$; specifically, to satisfy a linear restriction $x_t - y_t^\omega \leq w_t^\omega$. These resource actions incur a penalty $f(y_t^\omega) \geq 0$. Then, there are two grounds that encourage risk-seeking behavior.

G.1 Rational risk-seeking behavior is directly influenced by the relationship of g and f. Large incentives (given by g) and low penalties (given by a combination of a set or subset of scenarios of f) motivate larger profits. For instance, trivially, if $f(y^\omega) < g(x)$ for all ω then there is no motivation to take risk. This corresponds to a setting of the ruinous gambler being asked to participate in a lottery where

he could never win; we remove such settings from our consideration. This ground determines the *value* of the decision-maker's strategy.

G.2 Irrational risk-seeking behavior is motivated by an allowance of constraint violations. We consider the setting where the decision-maker is permitted to violate the restriction for up to ε fraction of scenarios; i.e., in these "lucky" scenarios, the profits of the decision-maker may be arbitrarily large. It is precisely these scenarios that attract our decision-maker to make risky irrational choices where the greed of profit in a few scenarios influences the entire decision-making framework. This ground determines the *level* of the decision-maker's risk.

The relationship of value and risk tolerance determines desirable and undesirable strategies. For example, consequences with low value and high risk are undesirable (such axioms have been postulated in earlier works, see, e.g., Ekenberg et al. (2001)). With this background, consider the following generic mathematical optimization — that we refine later — to determine a baseline setup of risk-seeking strategic behavior.

$$\max_{x,y} \quad \sum_{t=1}^{T} g(x_t) - \mathbb{E}\left[\sum_{t=1}^{T} f(y_t^\omega)\right] \tag{6.1a}$$

$$\text{s.t.} \quad \mathbb{P}\left(x_t - y_t^\omega \leq w_t^\omega,\ \forall t\right) \geq 1 - \varepsilon, \tag{6.1b}$$

$$x_t \geq 0, \quad \forall t \in T, \tag{6.1c}$$

$$y_t^\omega \geq 0, \quad \forall \omega \in \Omega,\ t \in T. \tag{6.1d}$$

In the prototype formulation given by model (6.1), the objective function (6.1a) considers an expectation over all scenarios of uncertainty; i.e., the value here in the sense of G.1 is that of a risk-neutral decision-maker. The probabilistic constraint (6.1b) caters to the level of risk in the sense of G.2. Unlike traditional probabilistic constraint optimization models (see, e.g., Prékopa (1995)), we allow ε to be large resulting in a framework where several constraints need not be satisfied. This framework applies to several practical real-world scenarios for risk-seeking behavior:

(i) R&D Investment: A firm decides on its research and development (R&D) budget, x_t, for each period t, while the availability of government grants or tax incentives, w_t^ω, fluctuates. If $x_t > w_t^\omega$, the firm must cover the shortfall with its own funds, denoted by y_t^ω. The firm receives a reward $g(x_t)$ from successful R&D projects but incurs a penalty $f(y_t^\omega)$ whenever internal funds are used to fill the gap. A fraction ε of all scenarios are considered *lucky* by the firm, where the government or investors provide the requested amount as "goodwill" effectively bailing out the firm. For example, the European Union allows authorization of such rescue aids for up to six months (Chindooroy et al., 2007), and the US spent $1.6 trillion to bail out firms in the financial crisis of 2008 (Jiang et al., 2014).

(ii) Money Laundering: An actor chooses the amount to launder, x_t, during each period t, but the capacity of the laundering channels, w_t^ω, is uncertain (e.g., due to scrutiny limits or operational constraints). Small amounts of laundering go unnoticed, e.g., in the UK shoplifting for goods less than £200 carried a perceived immunity until recently (Gov.uk, 2025). However, if $x_t > w_t^\omega$, the actor is forced to pay additional fines or graft, represented by y_t^ω. The actor may receive a benefit $g(x_t)$ from successful laundering, but must incur a penalty $f(y_t^\omega)$ if the channel is exceeded. Such over-limit laundering is allowed in at most an ε fraction of scenarios, considered "lucky" scenarios in which the actor avoids detection.

(iii) Tax Fraud: A taxpayer decides how much income to under-report, x_t, while the authority's enforcement threshold, w_t^ω, remains uncertain (e.g., based on audit rates or detection capabilities). For instance, the US Internal Revenue Service (IRS) maintains a secrecy around selection policies for auditing undeclared income (Snow and Warren, 2005). The taxpayer gains a reward $g(x_t)$ from reducing taxes owed, but incurs a penalty $f(y_t^\omega)$ upon detection if $x_t > w_t^\omega$. Fraudulent reporting beyond the threshold goes unnoticed in at most an ε fraction of instances. Past work shows that a larger degree of uncertainty about fraud detection increases compliance (Snow and Warren, 2005); for foundational work on tax evasion, see Allingham and Sandmo (1972).

(iv) Insurance Pricing: An insurer sets premiums x_t against uncertain claim payouts w_t^ω. If claims exceed premiums, the insurer faces losses of value $f(y_t^\omega)$, where $y_t^\omega = w_t^\omega - x_t$. In the classical actuarial setting, the insurer expects such losses restricted to a small fraction of ε scenarios. However, the insurer is often willing to trade strict solvency for broader objectives in various settings, such as micro-insurance in rural markets where affordability is paramount (Roth et al., 2007), social inclusion and market-entry in emerging economies (Cole et al., 2010), or coverage of rare high-cost catastrophic events (Kunreuther, 1996).

(v) Short-Selling with Uncertain Collateral: An investor determines the number of shares, x_t, to short in each period t. The required margin, w_t^ω, varies across scenarios ω. If the short position exceeds the margin requirement (i.e., $x_t > w_t^\omega$), the investor must provide additional collateral given by $y_t^\omega = \max\{0, x_t - w_t^\omega\}$. The investor earns a reward $g(x_t)$ from the short position, but incurs a penalty $f(y_t^\omega)$ when required to post additional collateral. A risky investor (e.g., in hedge funds or intra-day trading) accepts margin calls in a sufficiently large ε fraction of scenarios.

(vi) Random Forgiveness Mechanism: A student requests a monthly allowance, x_t, from parents whose available budget, w_t^ω, fluctuates. If $x_t > w_t^\omega$, the student must cover the shortfall (given by $y_t^\omega = \max\{0, x_t - w_t^\omega\}$) with personal funds.

The student gains a satisfaction utility $g(x_t)$ from the allowance but incurs an inconvenience penalty $f(y_t^\omega)$ when contributing personally. Occasional generosity or emotional leniency from the parents results in a random forgiveness mechanism in ε fraction of scenarios where the parents cover the full request regardless of their budget.

(vii) A venture capitalist invests an amount x_t in a startup at time t. Due to uncertainty in startup performance, the returns, w_t^ω, vary across scenarios ω. If the investment is not fully recovered (i.e., if $x_t > w_t^\omega$), the capital loss is given by $y_t^\omega = \max\{0, x_t - w_t^\omega\}$. The investor receives a reward $g(x_t)$ for deploying capital (e.g., expected upside), but suffers a penalty $f(y_t^\omega)$ from unrecovered funds. A willingness to accept under-performance in a large fraction of scenarios while aiming to capture high returns in a few scenarios reflects the skewed-return nature of early-stage investing, see, e.g., Kerr et al. (2014).

A risk-seeker usually finds satisfaction if their rewards are higher than the average, or a baseline, for example, a stock market index return used as a performance indicator for a mutual fund or an individual stock investor. This aligns with the prospect theory, which posits that outcomes are evaluated as gains or losses relative to a reference point (Kahneman and Tversky, 1988).

To capture preference for high-payoff outcomes, we introduce mean-preserving reward multipliers $r^\omega > 0$ satisfying

$$\mathbb{E}_\omega[r^\omega] = \frac{1}{|\Omega|} \sum_{\omega \in \Omega} r^\omega = 1. \tag{6.2}$$

The per-scenario payoff is defined as

$$U^\omega(x, y) = r^\omega \sum_{t \in T} g(x_t) - \sum_{t \in T} f(y_t^\omega)\,, \tag{6.3}$$

and its expectation

$$m(x, y) \;=\; \mathbb{E}_\omega[U^\omega(x, y)] = \sum_{t \in T} g(x_t) - \sum_{t \in T} \mathbb{E}_\omega[f(y_t^\omega)]\,, \tag{6.4}$$

which coincides with the objective of the risk-neutral model (6.1). If $r^\omega = 1$ for all $\omega \in \Omega$, the payoff for all scenarios for which $y_t^\omega = 0$ will always be the same and may not reflect the greediness for results from different scenarios.

For each scenario ω, let

$$s^\omega \;=\; \max\{0,\; U^\omega(x, y) - m(x, y)\}, \tag{6.5}$$

which rewards above-average scenario outcomes. We denote the upside mean-deviation risk-seeking model as:

$$\max_{x,y,s} \quad m(x,y) + \lambda\, \mathbb{E}_\omega[s^\omega] \tag{6.6a}$$

$$\text{s.t.} \quad \mathbb{P}(x_t - y_t^\omega \le w_t^\omega,\ \forall t \in T) \ge 1-\varepsilon, \tag{6.6b}$$

$$U^\omega(x,y) = r^\omega \sum_{t\in T} g(x_t) - \sum_{t\in T} f(y_t^\omega)\,, \quad \forall \omega \in \Omega, \tag{6.6c}$$

$$m(x,y) = \mathbb{E}_\omega[U^\omega(x,y)]\,, \tag{6.6d}$$

$$s^\omega = \max\{0,\ U^\omega(x,y) - m(x,y)\}, \quad \forall \omega \in \Omega, \tag{6.6e}$$

$$0 \le x_t \le \bar{x}, \quad \forall t \in T, \tag{6.6f}$$

$$y_t^\omega \ge 0, \quad \forall \omega \in \Omega,\ t \in T, \tag{6.6g}$$

$$s^\omega \ge 0, \quad \forall \omega \in \Omega. \tag{6.6h}$$

where $\lambda > 0$ is a risk appetite parameter. The objective function in (6.6a) encourages risk-seeking behavior for the decision maker in the sense of G.1 by seeking for the scenarios that have high payoff values.

The max-operator in (6.6e) can be represented with an SOS1 split by introducing auxiliary variables $s_-^\omega \ge 0$ and binary variables κ^ω,

$$s^\omega - s_-^\omega \;=\; U^\omega(x,y) - m(x,y), \qquad \forall \omega \in \Omega, \tag{6.7a}$$

$$(s^\omega,\ s_-^\omega) \;\in\; \text{SOS1}, \qquad \forall \omega \in \Omega, \tag{6.7b}$$

$$s^\omega \ge 0,\ s_-^\omega \ge 0 \qquad \forall \omega \in \Omega. \tag{6.7c}$$

$$\kappa^\omega = 0 \Rightarrow s^\omega = 0, \qquad \forall \omega \in \Omega, \tag{6.7d}$$

$$\kappa^\omega = 1 \Rightarrow s_-^\omega = 0, \qquad \forall \omega \in \Omega, \tag{6.7e}$$

$$\kappa^\omega \in \{0,1\}, \qquad \forall \omega \in \Omega. \tag{6.7f}$$

This construction guarantees $s^\omega = \max\{0,\ U^\omega(x,y) - m(x,y)\}$ by forcing at most one of s^ω and s_-^ω to be positive while preserving the identity $s^\omega - s_-^\omega = U^\omega(x,y) - m(x,y)$.

To obtain a mixed-integer reformulation of the chance constraint (6.6b), we introduce binary variables $z^\omega \in \{0,1\}$ indicating whether scenario ω is violated, i.e., $z^\omega = 1$ if $x_t - y_t^\omega \le w_t^\omega$ fails for some $t \in T$, and $z^\omega = 0$ otherwise. Using these indicators, the chance constraint is enforced by limiting the number of violating scenarios:

$$\sum_{\omega\in\Omega} z^\omega \;\le\; \lfloor \varepsilon|\Omega| \rfloor, \tag{6.8}$$

and by relaxing the scenario constraints only when $z^\omega = 1$. The penalty is waived by setting $y_t^\omega = 0$ for all $t \in T$ whenever a scenario is violated ($z^\omega = 1$), which corresponds to the modeling assumption that recourse costs are incurred only in non-violating scenarios.

We denote the set of scenarios where the per-scenario payoff is higher than the expected payoff as

$$\Omega_g := \{\omega \in \Omega : U^\omega(x,y) \geq m(x,y)\}, \tag{6.9}$$

which we call the "Greedy Set". We denote the set of violating scenarios as

$$\Omega_\ell := \{\omega \in \Omega : z^\omega = 1\}, \tag{6.10}$$

which we call the "Lucky Set".

The model (6.6) is reformulated as: The model (6.6) is reformulated as:

$$\max_{x,y,z,s,s_-,U,m} \quad m + \lambda \frac{1}{|\Omega|} \sum_{\omega\in\Omega} s^\omega \tag{6.11a}$$

$$\text{s.t.} \quad U^\omega = r^\omega \sum_{t\in T} g(x_t) - \sum_{t\in T} f(y_t^\omega), \qquad \forall \omega \in \Omega, \tag{6.11b}$$

$$m = \frac{1}{|\Omega|} \sum_{\omega\in\Omega} U^\omega, \tag{6.11c}$$

$$s^\omega - s_-^\omega = U^\omega - m, \qquad \forall \omega \in \Omega, \tag{6.11d}$$

$$(s^\omega,\ s_-^\omega) \in \text{SOS1}, \qquad \forall \omega \in \Omega, \tag{6.11e}$$

$$s^\omega \geq 0,\ s_-^\omega \geq 0, \qquad \forall \omega \in \Omega, \tag{6.11f}$$

$$\kappa^\omega = 0 \Rightarrow s^\omega = 0, \qquad \forall \omega \in \Omega, \tag{6.11g}$$

$$\kappa^\omega = 1 \Rightarrow s_-^\omega = 0, \qquad \forall \omega \in \Omega, \tag{6.11h}$$

$$\kappa^\omega \in \{0,1\}, \qquad \forall \omega \in \Omega, \tag{6.11i}$$

$$\sum_{\omega\in\Omega} z^\omega \leq \lfloor \varepsilon|\Omega| \rfloor, \tag{6.11j}$$

$$z^\omega = 0 \Rightarrow x_t - y_t^\omega \leq w_t^\omega, \qquad \forall \omega \in \Omega,\ t \in T, \tag{6.11k}$$

$$z^\omega = 1 \Rightarrow y_t^\omega = 0, \qquad \forall \omega \in \Omega,\ t \in T, \tag{6.11l}$$

$$0 \leq x_t \leq \bar{x}, \qquad \forall t \in T, \tag{6.11m}$$

$$y_t^\omega \geq 0, \qquad \forall \omega \in \Omega,\ t \in T, \tag{6.11n}$$

$$z^\omega \in \{0,1\}, \qquad \forall \omega \in \Omega. \tag{6.11o}$$

Model (6.11) can be interpreted in the context of research and development (R&D) portfolio allocation, where decision makers must commit resources to a project over a given period. Other sources of funding, such as government grants or investor contributions, are uncertain. The risk preference parameter λ captures the firm's appetite to pursue scenarios with substantial external funding support, reflecting its ambition to achieve faster growth relative to conservative decisions. The violation tolerance ε represents the proportion of time during which the combined external support and the firm's own funding are insufficient to cover the investment value, acknowledging that the firm may have contingency plans to address such shortfalls.

6.1.3 Example

Consider a two-period decision problem in which the decision variables $x_1 \geq 0$ and $x_2 \geq 0$ represent the amount of resource used in each period. There are three equally likely scenarios, and for each scenario ω, the maximum available resource in period t is denoted by w_t^ω:

$$w_1^1 = 100, \quad w_2^1 = 100, \quad w_1^2 = 2, \quad w_2^2 = 0, \quad w_1^3 = 0, \quad w_2^3 = 5. \tag{6.12}$$

The benefit function for each period is $g(x_t) = 8x_t$, and the penalty for exceeding the available resource is $f(y_t^\omega) = 25y_t^\omega$. The scenario reward multipliers are specified as $r^1 = 1.2$, $r^2 = 1.0$, and $r^3 = 0.8$.

A probabilistic constraint requires that the resource limits be satisfied in at least two of the three scenarios, corresponding to an allowable violation probability of $\varepsilon = \frac{1}{3}$. Although feasibility is possible for combinations with both $x_1 > 0$ and $x_2 > 0$ through positive slack variables, such solutions are not optimal under the risk-neutral objective. The expected penalty associated with the first unit of either x_1 or x_2 already exceeds its marginal benefit, and increasing either variable further only worsens the trade-off. Therefore, the optimal policy always allocates resources to a single period.

For the case $(x_1, 0)$, if $0 < x_1 \leq 2$, no slack is required, and the objective $m(x, y) = 8x_1$ increases up to $x_1 = 2$. For $x_1 > 2$, scenario 2 incurs slack $y_1^2 = x_1 - 2$, yielding

$$m(x, y) = \frac{50 - x_1}{3},$$

which is decreasing. Hence, the optimal solution in this case is $(2, 0)$ with objective value 16.

For the case $(0, x_2)$, if $0 < x_2 \leq 5$, the objective is $m(x, y) = 8x_2$, maximized at $x_2 = 5$. When $x_2 > 5$, scenario 3 incurs slack $y_2^3 = x_2 - 5$, giving

$$m(x, y) = \frac{125 - x_2}{3},$$

a decreasing function. Thus, the optimal solution is $(0, 5)$ with objective value 40. In summary, the risk-neutral optimal decision is $(0, 5)$ with an objective value of 40.

Next, we consider the upside risk-seeking objective in (6.6) with a parameter $\lambda > 0$. Following a similar approach to the risk-neutral analysis, we first examine the case $(x_1, 0)$. If $0 < x_1 \leq 2$, then the per-scenario payoffs are $U^1(x, y) = 9.6x_1$, $U^2(x, y) = 8x_1$, and $U^3(x, y) = 6.4x_1$. The objective value is therefore

$$m(x, y) + \lambda \mathbb{E}\left[(U^\omega(x, y) - m(x, y))^+\right] = 8x_1 + \frac{1.6\lambda x_1}{3} = \left(8 + \frac{8}{15}\lambda\right) x_1,$$

which is increasing in x_1. Hence, the optimal solution in this range is $(2, 0)$ with objective value $16 + \frac{16}{15}\lambda$.

For $x_1 > 2$, scenario 2 requires slack $y_1^2 = x_1 - 2$, and the per-scenario payoffs become

$$U^1(x, y) = 9.6x_1, \quad U^2(x, y) = 8x_1 - 25(x_1 - 2) = 50 - 17x_1, \quad U^3(x, y) = 6.4x_1.$$

From these, we obtain

$$U^1(x, y) - m(x, y) = \frac{29.8x_1 - 50}{3} > 0, \quad U^2(x, y) - m(x, y) = \frac{100 - 50x_1}{3} < 0,$$

and

$$U^3(x, y) - m(x, y) = \frac{20.2x_1 - 50}{3},$$

where the last term is nonnegative when $x_1 > \frac{250}{101}$. If $2 < x_1 \leq \frac{250}{101}$, the objective becomes

$$\frac{50 - x_1}{3} + \frac{\lambda}{3}\left(\frac{29.8x_1 - 50}{3}\right) = \frac{(29.8\lambda - 3)x_1 + 50(3 - \lambda)}{9},$$

which is increasing when $\lambda > \frac{15}{149}$. Thus, the optimal solution in this case is $\left(\frac{250}{101}, 0\right)$ with objective value

$$\frac{50(3 - \lambda)}{9} + \frac{250}{909}(29.8\lambda - 3).$$

If $\lambda < \frac{15}{149}$, the function is decreasing, and the optimal solution remains $(2, 0)$ with the objective value $16 + \frac{16}{15}\lambda$.

When $\frac{250}{101} < x_1 \leq 100$, scenarios 1 and 3 are greedy. The objective becomes

$$\frac{50 - x_1}{3} + \frac{\lambda}{3}\left(\frac{50x_1 - 100}{3}\right) = \frac{(50\lambda - 3)x_1 + 50(3 - 2\lambda)}{9},$$

which is increasing if $\lambda > 0.06$. In this region, the optimal solution is $(100, 0)$ with objective value $\frac{4900\lambda - 150}{9}$.

For the case $(0, x_2)$, if $0 < x_2 \leq 5$, the per-scenario payoffs are $U^1(x, y) = 9.6x_2$, $U^2(x, y) = 8x_2$, and $U^3(x, y) = 6.4x_2$. The objective value becomes

$$8x_2 + \frac{\lambda}{3}(1.6x_2) = \left(8 + \frac{8}{15}\lambda\right)x_2,$$

which is increasing, leading to the solution $(0, 5)$ with objective value $40 + \frac{8}{3}\lambda$.

If $x_2 > 5$, scenario 3 requires slack $y_2^3 = x_2 - 5$, giving

$$U^1(x, y) = 9.6x_2, \quad U^2(x, y) = 8x_2, \quad U^3(x, y) = 125 - 18.6x_2.$$

TABLE 6.1: Optimal decisions, objective values, and associated lucky and greedy scenario sets for selected values of the risk appetite parameter λ.

λ	Regime	(x_1^*, x_2^*)	Objective	Ω_ℓ	Ω_g
0	$\lambda \leq \frac{15}{274}$	$(0,5)$	40.00	$\{2\}$	$\{1\}$
0.04	$\lambda \leq \frac{15}{274}$	$(0,5)$	40.11	$\{2\}$	$\{1\}$
0.06	$\lambda > \frac{15}{274}$	$(0,100)$	43.20	$\{2\}$	$\{1,2\}$
0.08	$\lambda > \frac{15}{274}$	$(0,100)$	54.82	$\{2\}$	$\{1,2\}$

In this range, scenarios 1 and 2 exceed the expected value. The objective becomes

$$\frac{125 - x_2}{3} + \frac{\lambda}{3}\left(\frac{29.8x_2 - 125}{3} + \frac{25x_2 - 125}{3}\right) = \frac{(54.8\lambda - 3)x_2 + 125(3 - 2\lambda)}{9}.$$

This function is increasing if $\lambda > \frac{3}{54.8} = \frac{15}{274}$, leading to the optimal solution $(0,100)$; otherwise, the optimum is $(0,5)$.

We further observe that the extreme risk-seeking solution $(100,0)$ is strictly dominated by $(0,100)$ for all $\lambda \geq 0$. The objective function value for $(100,0)$ is given by $\frac{4900\lambda - 150}{9}$, whereas for $(0,100)$ it is $\frac{5230\lambda + 75}{9}$.

The optimal decision under the upside risk-seeking objective evolves systematically with the parameter λ. When $\lambda \leq \frac{15}{274}$, the decision-maker behaves essentially risk-neutral, and the optimal solution remains $(0,5)$ with objective value $40 + \frac{8}{3}\lambda$. Once $\lambda > \frac{15}{274}$, the model begins to favor higher-payoff but riskier scenarios. This threshold represents the *tipping point of rationality*. Below this value, the greed (upside reward) is insufficient to justify the structural risk of the $(0,100)$ solution. However, once λ crosses this threshold, the decision-maker effectively abandons the "average" scenario to chase the "jackpot" in scenarios 1 and 2.

6.1.4 Discussion on Risk-Seeking Model

We discuss the experimental results from solving the probabilistic-constrained model (6.6) under various parameter settings. In this case, we consider a linear reward, $g(x) = x$, and a penalty, $f(y) = By$. The focus is on how risk-seeking behavior emerges when the violation tolerance ε, the risk-seeking parameter λ, and the penalty B vary. The model is evaluated over twenty scenarios and four times of available resources w_t^ω.

We draw the reward multipliers r^ω from a normal distribution with parameters $(\mu, \sigma) = (1, 0.1)$ and then reorder them so that scenarios with higher mean budgets receive higher multipliers. This construction aligns favorable reward outcomes with more abundant resource availability, allowing the optimization model to express risk-seeking tendencies more visibly.

TABLE 6.2: Computation results of solving model (6.6) across tolerance ε, penalty ratio B, and risk preference parameter λ. Columns report the objective value, expected payoff $m(x, y)$, and the payoff ranges for lucky and greedy scenarios.

ε	B	λ	Objective	$m(x,y)$	Lucky Range	Greedy Range
0.1	10	0.0	168	168	[142.23, 186.08]	[172.27, 209.48]
		0.5	175.81	168	[142.78, 186.80]	[172.56, 210.29]
		1.0	183.67	167.75	[143.05, 187.15]	[170.80, 209.31]
		2.0	2593.97	-4619.85	[176.32, 187.24]	[173.35, 209.41]
	100	0.0	155.76	155.76	[121.78, 147.05]	[156.57, 179.36]
		0.5	158.46	155.76	[121.78, 147.05]	[156.57, 179.36]
		1.0	174.40	53.24	[140.48, 183.79]	[140.48, 206.90]
		2.0	24206.09	-47844.48	[175.97, 186.86]	[173.00, 210.36]
0.3	10	0.0	182.63	182.63	[143.05, 187.15]	[183.92, 209.31]
		0.5	185.78	182.63	[143.15, 187.29]	[184.05, 209.00]
		1.0	189.42	181.52	[144.61, 195.23]	[184.29, 212.99]
		2.0	2597.17	-4616.69	[179.68, 209.83]	[176.65, 212.97]
	100	0.0	182.29	182.29	[142.52, 186.46]	[183.24, 209.91]
		0.5	185.44	182.29	[142.52, 186.46]	[183.24, 209.91]
		1.0	188.56	182.29	[142.52, 186.46]	[183.24, 209.91]
		2.0	24209.06	-47841.66	[178.68, 208.67]	[175.67, 213.61]
0.5	10	0.0	185.25	185.25	[145.83, 209.82]	[185.85, 212.95]
		0.5	188.32	185.25	[145.98, 210.03]	[186.04, 213.17]
		1.0	191.54	184.75	[145.84, 209.83]	[185.87, 212.97]
		2.0	2598.92	-4615.26	[179.31, 211.17]	[179.31, 214.33]
	100	0.0	184.45	184.45	[144.21, 194.69]	[185.41, 212.39]
		0.5	187.60	184.45	[144.21, 194.69]	[185.41, 212.39]
		1.0	190.84	184.45	[144.21, 194.69]	[185.41, 212.39]
		2.0	24210.98	-47839.99	[178.79, 210.56]	[178.79, 215.54]

When $\lambda = 0$, the model behaves in a risk-neutral manner and maximizes the expected payoff $m(x, y)$ without emphasizing upside deviations. Table 6.2 shows that these risk-neutral solutions produce moderate decision values. As λ increases above zero, the model amplifies the importance of the positive deviation term $(U^{\omega}(x, y) - m(x, y))^{+}$, which rewards outcomes that exceed the mean payoff. Larger λ values drive higher total decision magnitudes in x and substantially increase the objective value. The trade-off becomes clear: the model achieves higher potential rewards but accepts lower or even negative mean payoffs. For $\lambda = 2$, the solutions demonstrate extreme risk-seeking behavior, with large objective values but strongly negative expected payoffs.

The violation tolerance ε determines how many scenarios can breach the resource constraints. Small values of ε, such as 0.1, restrict violations and lead the model to adopt conservative allocations. As ε grows to 0.3 and 0.5, the model uses the additional flexibility to allocate more resources toward favorable scenarios. The increase in ε expands the "Lucky" and "Greedy" payoff ranges and raises the objective, indicating that the optimizer converts probabilistic tolerance directly into more aggressive, reward-seeking allocations.

The parameter B controls how the model balances rewards against penalties. A lower penalty B reduces the cost of infeasibility and encourages larger decisions in x. Comparing $B = 10$ and $B = 100$ across all λ values shows that smaller penalties lead to higher objective values and broader "Lucky" and "Greedy" payoff ranges. Conversely, larger B values impose stronger penalties, reducing both the scale of x

and the variability in payoffs. For $\lambda = 0$, 0.5, and 1, the Lucky and Greedy ranges are the same for some tolerance levels $\varepsilon \in \{0.3, 0.5\}$. The same optimal decision vector x^* is found in these circumstances, which is why there are the same sets of lucky and greedy scenarios. The only difference is in the objective value, which varies with the risk choice parameter λ.

The Greedy Range is, by definition, composed of payoffs that exceed the expected value $m(x, y)$ and therefore always lies above the mean. In contrast, the Lucky Range arises from feasibility violations and depends on the cost structure of the resource constraints. Scenarios with smaller available budgets, w_t^ω, are cheaper to violate and are thus more likely to appear in the Lucky Set. As the risk-seeking parameter λ increases, the model prioritizes positive outcomes and allocates more resources to violated scenarios, resulting in significant payoffs. As a result, the Lucky Range shows a clear increasing trend with λ. The Greedy Range, on the other hand, deviates from this trend. Because the reward multipliers r^ω are fixed, the potential upside among high-payoff scenarios is limited, and the mean payoff $m(x, y)$ drops as the model grows more risk-seeking.

Overall, the results show a consistent trend: increasing λ or ε, or lowering B, shifts the model from conservative to risk-seeking behavior. The optimizer deliberately uses its violation allowance to pursue high-payoff scenarios and expand the potential reward range. This interaction among λ, ε, and B demonstrates how the model systematically converts higher risk tolerance into a stronger pursuit of upside performance.

6.1.5 Approximation for Joint Probabilistic-Constrained problem

Solving model (6.6) generally requires multi-dimensional integration and precise knowledge of the probability distribution of uncertainty. Since these computations are often intractable, especially when the number of scenarios and time periods are large, approximation methods are necessary to obtain feasible solutions or bounds.

In recent work, Singh and Watson (2019) approximate a two-stage joint probabilistic constrained optimization model by using set union approximations. Defining $A_t := \{\omega : x_t > y_t^\omega + w_t^\omega\}$ as the set of scenarios where failures occur at time t in (6.1), the joint probabilistic constraint can be reformulated as:

$$\mathbb{P}\left(\bigcup_{t \in T} A_t\right) \leq \varepsilon. \tag{6.13}$$

To approximate bounds for the union set, they consider only the terms $S_1 = \sum_{t \in T} \mathbb{P}(A_t)$ and $S_2 = \sum_{t,t' \in T, t' > t} \mathbb{P}(A_t \cap A_{t'})$. Replacing the probability of the union set with its lower bound provides an upper bound for the maximization model,

while replacing it with its upper bound yields a lower bound for the maximization model.

In previous studies, the variables were defined as follows: $u_t^\omega = 1$ indicates a failure at time t in scenario ω, and $v_{tt'}^\omega = 1$ indicates failures in scenario ω at both t and t'. This construction is beneficial because it allows us to consider the failures for each scenario in a specific time period, which is more informative. The terms S_1 and S_2 can then be expressed in terms of u and v with the number of binary variables $|\Omega||T|$ for u and $|\Omega||T|^2$ for v. Therefore, the number of variables associated with v scales by a factor of $|T|$ relative to u. A large number of binary variables might lead to models that are hard to solve.

6.1.6 Our Contributions

In this work, we present a risk-seeking behavior that is reflected in real-world decision-making, which might be suboptimal, and then we formulate it as a probabilistic-constrained model in two ways of risk-seeking. Solving such a model is intractable, so we propose alternative methods: to bound the model in the spirit of Singh and Watson (2019), to redefine the variables used in the approximation, and to investigate bounds obtained from solving linear programs. The key contributions of this article are as follows:

1. We introduce a risk-seeking model and formulate it as a probabilistic constraint.
2. We redefine the variables to approximate the probability of union sets where the discrete scenario sample has equal probabilities for each scenario.
3. We approximate the probabilistic constraint based on the probability of union sets using binomial moments.
4. We examine the relationship between the "Lucky" and "Greedy" sets in the computational results.

In this work, we discuss risk-seeking behavior and the approximation of probabilistic-constrained models in Section 6.1. In Section 6.2, we redefine the variables, introduce new optimization model formulations based on binomial moment bounds. Section 6.3 presents computational results, and Section 6.4 concludes the work with a summary and discussion.

6.2 Approximation Models

In this section, we first define binary and integer variables to capture the failure of each scenario and represent the sum of probability terms as the sum of these binary variables. We then present classical bounds used in bounding techniques from Singh et al. (2018); Singh (2020). After that, we introduce probabilistic bounds derived from the linear programs proposed in Prékopa (1988) and provide the corresponding approximation models based on these methods.

6.2.1 Variables' Definition

We let $B^\omega := \{t : x_t > y_t^\omega + w_t^\omega\}$ denote the set of time periods where failures occur in the scenario ω and $b^\omega := |B^\omega|$ denote the number of time periods that fail in scenario ω. Let $a_t := |A_t|$ where $A_t := \{\omega : x_t > y_t^\omega + w_t^\omega\}$, we have $\sum_{t\in T} a_t = \sum_{\omega\in\Omega} b^\omega$ the number of failure pairs (ω, t) be the same. As the probability $\mathbb{P}(A_t) = \frac{1}{N} a_t$, we have $S_1 = \sum_{t\in T} \mathbb{P}(A_t) = \frac{1}{N}\sum_{t\in T} a_t = \frac{1}{N}\sum_{\omega\in\Omega} b^\omega$ where $N = |\Omega|$. The number of two time periods (t, t') that both fail in the scenario ω is $\binom{b^\omega}{2}$. Thus, we can write $S_2 = \sum_{\omega\in\Omega} \frac{1}{N}\binom{b^\omega}{2}$ since the probability of each scenario is equal to $1/N$. Similarly, we can extend to $S_n = \sum_{\omega\in\Omega} \frac{1}{N}\binom{b^\omega}{n}$.

TABLE 6.3: Example of failures across four time periods and seven scenarios. Here, $\times$ represents a failure in scenario ω at time t.

	t_1	t_2	t_3	t_4
ω_1	$\times$	$\times$	$\times$	
ω_2		$\times$		$\times$
ω_3	$\times$			
ω_4			$\times$	
ω_5		$\times$	$\times$	
ω_6				$\times$
ω_7				

Table 6.3 illustrates an example of failures in four time periods and seven scenarios; the probabilities of failure at each time period are:

$$\mathbb{P}(A_{t_1}) = \tfrac{2}{7}, \quad \mathbb{P}(A_{t_2}) = \tfrac{3}{7}, \quad \mathbb{P}(A_{t_3}) = \tfrac{3}{7}, \quad \mathbb{P}(A_{t_4}) = \tfrac{2}{7}.$$

Thus, we calculate the following values:

$$S_1 = \tfrac{10}{7}, \quad S_2 = \tfrac{5}{7}, \quad S_3 = \tfrac{1}{7}.$$

The number of failures for each scenario is:

$$b^{\omega_1} = 3, \quad b^{\omega_2} = 2, \quad b^{\omega_3} = 1, \quad b^{\omega_4} = 1, \quad b^{\omega_5} = 2, \quad b^{\omega_6} = 1, \quad b^{\omega_7} = 0.$$

We observe that:

$$S_1 = \tfrac{10}{7} = \tfrac{1}{7}\sum_{\omega\in\Omega} b^\omega, \quad S_2 = \tfrac{5}{7} = \tfrac{1}{7}\sum_{\omega\in\Omega}\binom{b^\omega}{2}, \quad S_3 = \tfrac{1}{7} = \tfrac{1}{7}\sum_{\omega\in\Omega}\binom{b^\omega}{3}.$$

To capture the representation of failure in scenario ω at time t, we still use the binary variable u_t^ω as in Singh and Watson (2019). To avoid introducing a large number of variables and nonlinear constraints, we represent the count $b^\omega = \sum_{t\in T} u_t^\omega$ via a one-hot encoding. Let $\theta_\ell^\omega \in \{0,1\}$ be a binary variable for $\ell \in \{0,1,\ldots,|T|\}$ such that

$$\sum_{\ell=0}^{|T|} \theta_\ell^\omega = 1, \qquad \forall \omega \in \Omega, \tag{6.14a}$$

$$\sum_{t\in T} u_t^\omega = \sum_{\ell=0}^{|T|} \ell\, \theta_\ell^\omega, \qquad \forall \omega \in \Omega. \tag{6.14b}$$

The value of $\theta_\ell^\omega = 1$ if there are ℓ failures in scenario ω, i.e., $b^\omega = \ell$, and zero otherwise. Valid inequalities can be implemented to strengthen the formulation are

$$u_t^\omega \le 1 - \theta_0^\omega, \qquad \forall \omega \in \Omega,\ t \in T, \tag{6.15}$$

which enforces that if $\ell = 0$ is selected then no period can be violated.

With this variable definition, we can write

$$S_1 = \frac{1}{N}\sum_{\omega\in\Omega} b^\omega = \frac{1}{N}\sum_{\omega\in\Omega}\sum_{\ell=0}^{|T|} \ell\, \theta_\ell^\omega,$$

and

$$S_2 = \frac{1}{N}\sum_{\omega\in\Omega}\binom{b^\omega}{2} = \frac{1}{N}\sum_{\omega\in\Omega}\sum_{\ell=0}^{|T|}\binom{\ell}{2}\theta_\ell^\omega.$$

More generally,

$$S_n = \frac{1}{N}\sum_{\omega\in\Omega}\binom{b^\omega}{n} = \frac{1}{N}\sum_{\omega\in\Omega}\sum_{\ell=0}^{|T|}\binom{\ell}{n}\theta_\ell^\omega, \qquad n = 1,2,\ldots.$$

Although $S_2 = \frac{1}{2N}\sum_\omega b^\omega(b^\omega - 1) = \frac{1}{2N}\sum_\omega((b^\omega)^2 - b^\omega)$ can be written using a new variable, $c^\omega = (b^\omega)^2$, introduces a quadratic (and for $(b^\omega)^3, (b^\omega)^4, \ldots$ higher-degree) relation. This either yields a nonconvex polynomial model or requires additional linearizations that become weaker as the degree increases. In contrast, the one-hot representation (6.14) allows all binomial terms $\binom{b^\omega}{n}$ to be expressed *linearly* as $\sum_k \binom{k}{n}\theta_k^\omega$ with the number of variables $|\Omega|(|T|+1)$.

6.2.2 Classical Bounds

The classical bounds discussed in Singh and Watson (2019) and Singh (2020) are derived from well-known probability inequalities. Specifically, the bounds based on Bonferroni (1936) are expressed as:

$$\mathbb{P}\left(\bigcup_{t\in T} A_t\right) \leq S_1 \quad \text{and} \quad \mathbb{P}\left(\bigcup_{t\in T} A_t\right) \geq S_1 - S_2. \tag{6.16}$$

To achieve tighter bounds, Sathe et al. (1980) introduced refined inequalities, which are given by:

$$\mathbb{P}\left(\bigcup_{t\in T} A_t\right) \leq S_1 - \frac{2}{|T|}S_2 \quad \text{and} \quad \mathbb{P}\left(\bigcup_{t\in T} A_t\right) \geq \frac{S_1 + 2S_2}{|T|^2}. \tag{6.17}$$

The tightest available lower bound using only terms S_1 and S_2 is provided by Dawson and Sankoff (1967) and is formulated as:

$$\mathbb{P}\left(\bigcup_{t\in T} A_t\right) \geq 2\frac{(z-1)S_1 - S_2}{z(z-1)}, \tag{6.18}$$

where $z = 2 + \left\lfloor \frac{2S_2}{S_1} \right\rfloor$.

To approximate the probabilistic constraint (6.13) with these probabilistic bounds, the variables u_t^ω are linked to the period constraints via

$$x_t - y_t^\omega - w_t^\omega \leq M_t^\omega \, u_t^\omega, \qquad \forall t \in T,\ \omega \in \Omega, \tag{6.19}$$

so that when $u_t^\omega = 0$ the original constraint $x_t - y_t^\omega \leq w_t^\omega$ is enforced. Similarly, the penalty-waiving term can strengthen the formulation as

$$y_t^\omega \leq \bar{y}_t^\omega \, (1 - u_t^\omega), \qquad \forall t \in T,\ \omega \in \Omega, \tag{6.20}$$

where $\bar{y}_t^\omega$ is a valid upper bound on y_t^ω.

We approximate the joint probabilistic constraint by replacing the probability of the union set with an upper bound. This yields a lower bound on the original constraint (6.1), resulting in a conservative approximation as the feasible region shrinks. Conversely, replacing it with a lower bound term provides an upper bound on (6.1), which acts as a relaxation by enlarging the feasible region.

6.2.3 Binomial Moment Bounds

In this work, we consider probabilistic bounds on the union of events, ($\mathbb{P}(\bigcup_{t\in T} A_t)$), based on binomial moments. The tightest bounds are derived by solving a linear programming problem, as presented by Prékopa (1988). The optimization formulation for the lower bound of the union probability is obtained by solving:

$$\begin{aligned}
\min \quad & \sum_{t=1}^{T} q_t && && \text{(6.21a)}\\
\text{s.t.} \quad & \sum_{t=1}^{T} \binom{t}{i} q_t = S_i, && i = 1,\dots,k, && \text{(6.21b)}\\
& q_t \geq 0, && t = 1,\dots,T, && \text{(6.21c)}
\end{aligned}$$

and the upper bound is obtained by solving

$$\begin{aligned}
\max \quad & \sum_{t=1}^{T} q_t && && \text{(6.22a)}\\
\text{s.t.} \quad & \sum_{t=1}^{T} \binom{t}{i} q_t = S_i, && i = 1,\dots,k, && \text{(6.22b)}\\
& q_t \geq 0, && t = 1,\dots,T. && \text{(6.22c)}
\end{aligned}$$

We consider the case from the first binomial moment to the fourth binomial moment ($k \in \{1,2,3,4\}$), meaning we include the terms S_1, S_2, S_3, and S_4. Using these two bounds (6.21) and (6.22), the approximated constraint is given by

$$\sum_{t\in T} q_t^* \leq \varepsilon, \tag{6.23}$$

where q_t^* for $t = 1,\dots,T$ is an optimal solution to the linear program (6.21) and (6.22), respectively.

We first formulate an optimization model where we replace the probability of the union set with $\sum_{t\in T} q_t^*$, where q^* is an optimal solution to the linear program (6.21). This provides a valid upper bound on the optimal objective value of the model (6.6). The optimization model becomes a bilevel linear program for the k-th binomial

moment as

$$\max_{x,y,q^*,s} \quad m(x,y) + \lambda\, \mathbb{E}_\omega[s^\omega]\,, \tag{6.24a}$$

$$\text{s.t.} \quad \sum_{t\in T} q_t^* \le \varepsilon, \tag{6.24b}$$

$$\text{(6.6c), (6.6d), (6.6f), (6.6g),} \tag{6.24c}$$

$$\text{(6.7a)} - \text{(6.7f),} \tag{6.24d}$$

$$\text{(6.14)} - \text{(6.15),} \tag{6.24e}$$

$$\text{(6.19)} - \text{(6.20),} \tag{6.24f}$$

$$u_t^\omega \in \{0,1\}, \quad \forall \omega \in \Omega,\ t \in T, \tag{6.24g}$$

$$\theta_\ell^\omega \in \{0,1\}, \quad \forall \omega \in \Omega,\ \ell = 0,\dots,|T|, \tag{6.24h}$$

$$q^* \in \operatorname*{argmin}_q \left\{ \sum_{t\in T} q_t \;\middle|\; \begin{array}{l} \displaystyle\sum_{t\in T} \binom{t}{k} q_t = \frac{1}{|\Omega|} \sum_{\omega\in\Omega} \sum_{\ell=0}^{|T|} \binom{\ell}{k} \theta_\ell^\omega, \\ q_t \ge 0, \quad \forall t \in T \end{array} \right\}. \tag{6.24i}$$

To reformulate the bilevel model (6.24) into a single-level reformulation, we introduce $\nu_k \in \mathbb{R}$ for $k \in \{1,2,3,4\}$ as Lagrange multipliers for the equality constraints in the lower-level problem, and $\mu_t \ge 0$ for the inequality constraints $q_t \ge 0$ for $t \in T$.

The Lagrangian function is defined as

$$\mathcal{L}(q,\nu,\mu) = \sum_{t\in T} q_t + \sum_{i=1}^{k} \nu_i \left(\sum_{t\in T} \binom{t}{i} q_t \;-\; \frac{1}{|\Omega|} \sum_{\omega\in\Omega} \sum_{\ell=0}^{|T|} \binom{\ell}{i} \theta_\ell^\omega \right) - \sum_{t\in T} \mu_t\, q_t. \tag{6.25}$$

To derive the KKT conditions for the lower-level problem, we take the partial derivative of $\mathcal{L}$ with respect to each q_t and setting it to zero; we get

$$\frac{\partial \mathcal{L}}{\partial q_t} = 1 + \sum_{i=1}^{k} \binom{t}{i} \nu_i - \mu_t = 0 \tag{6.26}$$

This is equivalent to

$$\mu_t = 1 + \sum_{i=1}^{k} \binom{t}{i} \nu_i, \quad \forall\, t \in T. \tag{6.27}$$

The primal feasibility constraints are $\sum_{t\in T} \binom{t}{i} q_t = \frac{1}{|\Omega|} \sum_{\omega\in\Omega} \sum_{\ell=0}^{|T|} \binom{\ell}{i} \theta_\ell^\omega$, $i = 1,\dots,k$ and $q_t \ge 0$, $\forall t \in T$. The dual feasibility constraints are $\mu_t \ge 0$, $\forall\, t \in T$. Finally, the complementary slackness are $\mu_t\, q_t = 0$, $\forall\, t \in T$. We obtain the single-level approximation model, which provides an upper bound to (6.6) as the following theorem.

Theorem 6.1 (Binomial Upper Bound)**.** *The objective value of the following model provides an upper bound to* (6.6)*:*

$$\max_{x,y,u,\theta,s,s_-,q,\nu,\mu} \quad m(x,y) + \lambda\, \mathbb{E}_\omega[s^\omega] \tag{6.28a}$$

$$s.t. \quad \sum_{t=1}^{|T|} q_t \le \varepsilon, \tag{6.28b}$$

$$\sum_{t=1}^{|T|} \binom{t}{i} q_t = \frac{1}{|\Omega|} \sum_{\omega\in\Omega} \sum_{\ell=0}^{|T|} \binom{\ell}{i} \theta_\ell^\omega, \quad i = 1,\dots,k, \tag{6.28c}$$

$$\mu_t = 1 + \sum_{i=1}^{k} \binom{t}{i} \nu_i, \quad \forall\, t \in T \tag{6.28d}$$

$$(q_t, \mu_t) \in \text{SOS1}, \quad t = 1,\dots,|T|, \tag{6.28e}$$

$$q_t \ge 0,\ \mu_t \ge 0, \quad t = 1,\dots,|T|, \tag{6.28f}$$

$$\nu_i \in \mathbb{R}, \quad \forall i \in \{1,\dots,k\} \tag{6.28g}$$

$$\text{(6.6c), (6.6d), (6.6f), (6.6g),} \tag{6.28h}$$

$$\text{(6.7a)} - \text{(6.7f),} \tag{6.28i}$$

$$\text{(6.14)} - \text{(6.15),} \tag{6.28j}$$

$$\text{(6.19)} - \text{(6.20),} \tag{6.28k}$$

$$\text{(6.24g)} - \text{(6.24h).} \tag{6.28l}$$

We use SOS1 branching to enforce complementarity by requiring that at most one of q_t and μ_t is positive for each $t = 1,\dots,|T|$, which is equivalent to $q_t\mu_t = 0$.

From $\max \sum_{i=1}^{T} q_i = -\min \sum_{i=1}^{T}(-q_i)$, we can similarly derive the KKT conditions to obtain the following theorem.

Theorem 6.2 (Binomial Lower Bound)**.** *The objective value of the following model provides a lower bound to* (6.6)*:*

$$\max_{x,y,u,\theta,s,s_-,q,\nu,\mu} \quad m(x,y) + \lambda\, \mathbb{E}_\omega[s^\omega] \tag{6.29a}$$

$$s.t. \quad \sum_{t=1}^{|T|} q_t \le \varepsilon, \tag{6.29b}$$

$$\sum_{t=1}^{|T|} \binom{t}{i} q_t = \frac{1}{|\Omega|} \sum_{\omega\in\Omega} \sum_{\ell=0}^{|T|} \binom{\ell}{i} \theta_\ell^\omega, \quad i = 1,\dots,k, \tag{6.29c}$$

$$\mu_t = -1 + \sum_{i=1}^{k} \binom{t}{i} \nu_i, \quad \forall\, t \in T \tag{6.29d}$$

$$(q_t, \mu_t) \in \text{SOS1}, \quad t = 1,\dots,|T|, \tag{6.29e}$$

$$q_t \ge 0,\ \mu_t \ge 0, \quad t = 1,\dots,|T|, \tag{6.29f}$$

$$\nu_i \in \mathbb{R}, \quad \forall i \in \{1,\dots,k\}, \tag{6.29g}$$

$$\text{(6.6c), (6.6d), (6.6f), (6.6g),} \tag{6.29h}$$

$$\text{(6.7a)} - \text{(6.7f),} \tag{6.29i}$$

$$\text{(6.14)} - \text{(6.15),} \tag{6.29j}$$

$$\text{(6.19)} - \text{(6.20),} \tag{6.29k}$$

$$\text{(6.24g)} - \text{(6.24h).} \tag{6.29l}$$

The only difference between Theorems 6.1 and 6.2 is the value of μ_t in the constraints (6.28d) and (6.29d).

6.3 Computational Experiments

In this section, we present a comparison of computational results for objective bounds of the model (6.6) from the approximation bounds in Theorem 6.1 and 6.2 using the first to the fourth moment of binomial.

6.3.1 Data Generation

The experiments utilize two datasets to represent contrasting risk environments. The first dataset, denoting *Baseline Dataset*, generates a series of scenarios with values over a particular period of time that replicate standard stochastic volatility characterized by high temporal correlation. A seasonal trend is determined through a sinusoidal function, with observed values resulting from the relationship of this trend and a lognormal noise component. Furthermore, scenarios may experience global shocks, resulting in a subset of scenarios being entirely scaled down, indicative of a bad

period characterized by persistent risk across the time horizon. The second dataset, denoting *Sparse Dataset*, generates scenarios with values over a time period that simulate rare, discrete component failures. The scenarios are set to a high operating baseline of 85% capacity, but they experience random, abrupt declines to a low level of 20% capacity, as dictated by a Poisson distribution.

Each dataset includes the sample sizes of 100 scenarios with 12 time periods. We then generate the reward multiplier r^{ω} using a normal distribution $\mathcal{N}(1, 0.1)$, and reorder the scenarios based on average resources across all periods, arranging them from smallest to largest, while also sorting the reward multipliers to reflect risk-seeking behavior that prioritizes favorable scenario outcomes.

6.3.2 Computation Results

We solve the model (6.6) and its approximation models based on the binomial moment in Theorems 6.1 and 6.2 from the first to the fourth moment. The parameter values are $\varepsilon \in \{0.2, 0.3, 0.4\}$ and $\lambda \in \{0, 0.25, 0.5, 0.75, 1\}$. All computations are performed on a computing cluster equipped with an Intel Xeon Gold 6138 2.0 GHz processor, 192 GB of RAM, and Gurobi version 10.0.3, using Python 3.8.3 with a time limit of 7200 seconds.

TABLE 6.4: Summary of the binomial moment bounds obtained from solving approximation models in Theorems 6.1 and 6.2 with the two generated datasets. Here, columns "LB", "UB", "GAP" represent the average lower bound, upper bound, and MIPGAP over parameters $\varepsilon \in \{0.2, 0.3, 0.4\}$ and $\lambda \in \{0, 0.25, 0.5, 0.75, 1\}$. The bold values indicate the tightest bound or lowest MIPGAP achieved.

	Baseline Dataset			**Sparse Dataset**		
Model	**LB**	**UB**	**Gap (%)**	**LB**	**UB**	**Gap (%)**
True Model	584.7	812.3	38.0	552.4	681.6	22.6
Order-1 LB	514.9	580.9	**12.1**	531.5	563.9	**5.8**
Order-1 UB	1078.6	1224.0	**13.1**	943.2	1032.7	**9.2**
Order-2 LB	**584.4**	936.6	59.2	**551.8**	817.1	47.0
Order-2 UB	699.7	1130.0	60.7	629.8	962.8	52.2
Order-3 LB	583.8	1118.0	90.3	550.8	931.4	68.1
Order-3 UB	593.3	1093.2	83.4	576.4	929.3	60.8
Order-4 LB	583.2	1065.2	81.5	550.9	907.0	63.5
Order-4 UB	590.0	**1068.1**	79.8	568.0	**889.0**	55.8

Table 6.4 demonstrates the bounds obtained from solving the approximation models with binomial moment bounds using orders one to four. The models with named LB provide a valid lower bound to the true model, while the models with named UB give a valid upper bound. For both datasets, the binomial of order one models solve the

most efficiently with the smallest MIPGAP within the time limit, which is also faster than solving the true model directly. Solving these models yields intuitive bounds for the overall model.

For the lower bound approximations, as shown in Table 6.4, higher objective values correspond to tighter bounds. The lower bound obtained from the second-order model is larger than that of the first-order bound, with a 13% bound improvement on the baseline dataset and 4% on the sparse dataset. Although approximating with a binomial of the third and fourth order theoretically presents tighter bounds with only a few more variables needed, using the second-order provides the tightest lower bound, as it makes the model less complicated. Moreover, the lower bound obtained from the true model is almost the same as from the second-order approximation. In terms of the MIPGAP, it increases significantly from the second-order to the third-order while it slightly drops from the third to the fourth order. Therefore, the second-order binomial term is sufficient to provide a tight lower bound.

For upper bound approximations, lower objective values correspond to tighter bounds. We observe that the upper bounds decrease with each successive order, showing an improvement of approximately 7% from the first to the second order, followed by smaller marginal gains of roughly 3% between subsequent orders. Consistent with theoretical expectations, the fourth-order model yields the tightest approximation. Regarding the MIPGAP, the trend mirrors that of the lower bound models. Thus, incorporating higher-order binomial moments leads to tighter upper bounds.

TABLE 6.5: Performance comparison table obtained from solving approximation models in previous literature, (6.16)- (6.18), and Theorems 6.1 and 6.2 with the second order. Here, columns "LB", "UB", "GAP" represent the average lower bound, upper bound, and MIPGAP. The bold values indicate the valid bounds for the approximation models.

Model	LB	UB	Gap (%)
True Model	**199.5**	**288.9**	33.3
(6.16) LB	**186**	231	15.6
(6.16) UB	790.6	**790.6**	0
(6.17) LB	**186**	233.7	15.7
(6.17) UB	521.6	**581.3**	6.5
(6.18) UB	206.7	**428.6**	104.1
Order-2 Binomial LB	**185.9**	230.4	15.8
Order-2 Binomial UB	209.8	**422.1**	97.4

Table 6.5 compares the bounds obtained from the approximation models with previous probabilistic bounds and the second-order of the proposed method. For the lower bound approximations, all methods achieve similar bounds and MIPGAPs

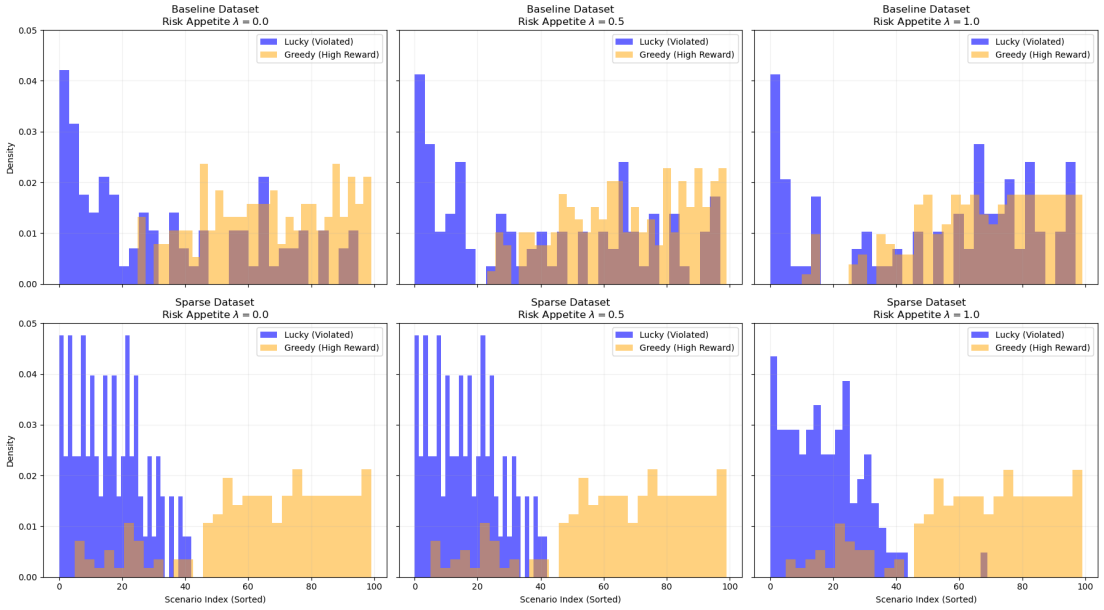


FIGURE 6.1: Distribution of scenario indices for the "Lucky" (violated, blue) and "Greedy" (high-reward, orange) sets, comparing the Baseline dataset (top row) and the Sparse dataset (bottom row) across varying risk appetites ($\lambda \in \{0.0, 0.5, 1.0\}$).

because these bounds are improved by finding feasible solutions. In contrast, the upper bounds represent dual bounds, which are improved through the pruning process of the branch-and-bound tree. While the upper bounds from (6.16) can be solved to optimality within seconds, the resulting bounds are loose. The upper bounds from (6.17) provide a tighter approximation with small MIPGAPs. While the bounds from (6.18) are the tightest among previous works (Singh, 2020), the upper bounds from our proposed model are the tightest overall.

Figure 6.1 illustrates the distribution of "Lucky" and "Greedy" scenarios, which highlights the fundamental difference between the two datasets. Intuitively, a chance-constrained solution should be lucky enough to ignore the scenarios with the fewest resources (tightest constraints) while being greedy enough to meet the scenarios with the highest benefits. This pattern is obvious in the Sparse dataset. The histogram indicates a clear separation: the Lucky set (violated scenarios, blue bars) is significantly concentrated at the lower indices, corresponding to low-resource situations, whereas the Greedy set (high-reward scenarios, orange bars) is grouped at the upper indices. The Baseline dataset, on the other hand, shows a notable structural overlap. Greedy scenarios are not limited to high indices but occur throughout the spectrum, most frequently in the low-resource zone. This forces the solver to make a difficult decision, which contributes to the combinatorial complexity reflected in the higher MIPGAPs, see Table 6.4.

The decision-maker's risk appetite (λ) changes the selection of these decisions. As shown in Figure 6.1, the Lucky set changes when λ goes up. In the risk-neutral setting

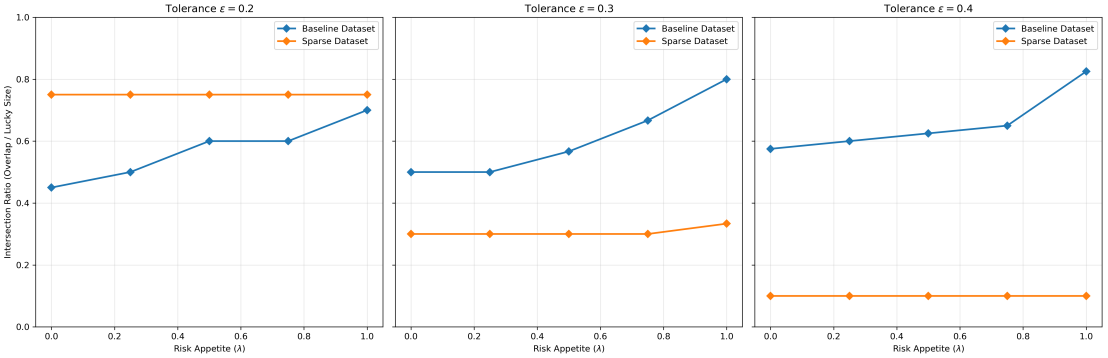
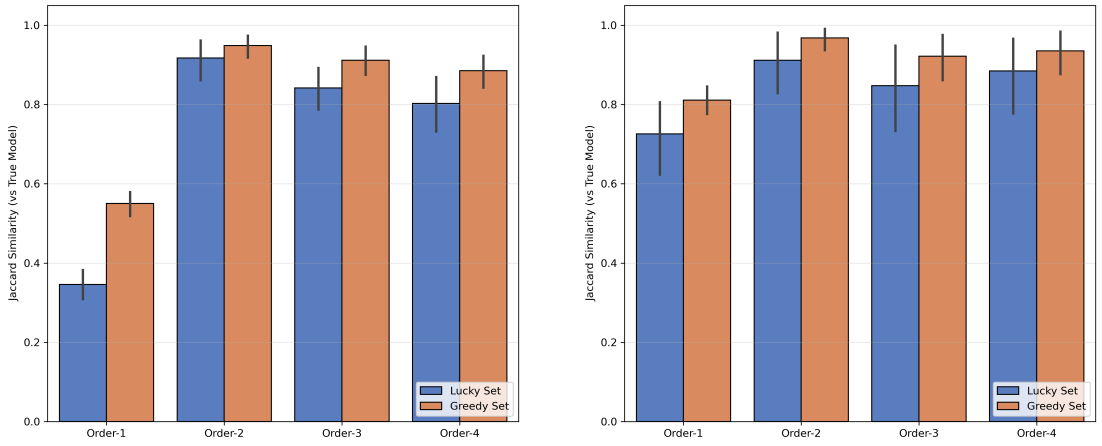


FIGURE 6.2: The intersection ratio between the optimal set of violated scenarios ("Lucky") and the set of high reward scenarios ("Greedy") is analyzed as a function of risk appetite (λ) across different tolerance levels ($\varepsilon \in \{0.2, 0.3, 0.4\}$) in two datasets.

($\lambda = 0$), the model only maximizes expected profits. This keeps the Lucky set closely grouped around the violations with the lowest possible cost. But as λ gets bigger, the actor cares more about the possible upside. The Lucky set is shifting to situations with bigger rewards, which is shown by the histogram as a movement to the right. This behavior happens because the actor is ready to violate for moderately good scenarios in a risk-seeking regime if it means they can get to the extreme "jackpot" scenarios that help them meet the risk-seeking objective. This change is more noticeable in the baseline dataset.

Figure 6.2 shows the intersection ratio, which is the percentage of lucky and greedy scenarios across different tolerances from the solution of the true model (6.6). There is a clear structural difference between the two sets of data. The Baseline dataset consistently exhibits a high intersection ratio ($\approx 0.4 - 0.6$) even in a risk-neutral case. This ratio is monotonically increasing over the level of risk appetite up to just above 80 percent. This graph shows that with the nature of scenarios like this, the risk-seeking model highly encourages following the value of the risk appetite parameter. On the other hand, the intersection ratio is relatively high when the allowable violation is small (approximately 0.7 at $\varepsilon = 0.2$) but decreases to around 0.1 when $\varepsilon = 0.4$ in the sparse dataset. This means that an increase in the violation tolerance level provides a clearer separation between lucky and greedy in this dataset. However, in this case, the sparse dataset does not show significant changes in these ratios across different risk appetites; there is only a slight upward change when λ is adjusted from 0.8 to 1.0 in the setting of $\varepsilon = 0.3$.

We examined the Jaccard similarity between the solutions of the approximation model and model (6.6), as illustrated in Figure 6.3. The Order-1 approximation has inconsistent fidelity; it accurately represents approximately 80% of the sets in the Sparse dataset, however its similarity scores are below 0.5 in the Baseline dataset. This illustrates that neglecting correlations may result in a fundamental misidentification of the constraints that are violated, especially in structurally complex datasets. On the



(A) Baseline Dataset (B) Sparse Dataset

FIGURE 6.3: Structural fidelity of the approximation models compared to the model (6.6). The Jaccard similarity is calculated for both the lucky and greedy sets.

other hand, the Order-2 approximation frequently produces similarity scores greater than 0.9, implying high structural similarity. This outcome demonstrates that using pairwise joint probabilities is adequate to achieve satisfactory results. Higher-order models provide negligible improvement, indicating that pairwise interactions are the primary source of complexity in these cases.

6.4 Conclusion

In this work, we investigated risk-seeking behavior in stochastic optimization, a different perspective of the decision-making framework, which mostly focuses on risk-averse or risk-neutral behavior. We formulated a probabilistic-constrained optimization model that allows for large violation tolerance without any penalty, which is considered a lucky scenario, while also encouraging risk-seeking decision-making with greedy scenarios where there is upside from the expected value.

To solve the proposed model, we work in the same spirit as in Singh and Watson (2019); Singh (2020) by considering the probabilistic constraint as a constraint on the probability of union sets. We redefine the variables' definition to make them effective to represent higher-order joint sets.

We provide alternative models to derive valid bounds over scenario realizations based on binomial moment bounds from the first to the fourth order. The approximation models become bilevel linear programs, which are later reformulated into single-level formulation models using KKT conditions.

Our computational experiments on two distinct datasets offer insights about the trade-off between effective bounds and complexity. While the Order-1 approximation

is computationally efficient, the obtained bounds are loose, which could be good to see the ranges of objectives. In contrast, the Order-2 approximation shows as the most effective moment of the approximation method. It provides a significant improvement in bound tightness over the first-order model and attains high structural similarity in lucky and greedy sets of the true models. In contrast, higher-order provides tighter upper bounds as expected in theory in exchange for being harder to solve optimally.

We show that the relationship between lucky and greedy scenarios depends on not only allowable violation tolerance ε and risk-appetite level λ, but also the distribution of the resource scenarios. The high-reward scenarios can be violated for the jackpot scenario if the risk appetite is high.

In summary, this study provides a mathematical framework for modeling and solving risk-seeking problems. By using monomial moments from solving linear programs together with novel variable definitions, we can include high-order bounds in the calculation without needing a large number of added variables. The computation result, using second-order approximations, effectively balances tightness and tractability compared to other orders.

Chapter 7

Measuring the Economic Value of Wind–Solar Complementarity in Europe Using Chance Constraints

In the previous chapter, we investigated risk-seeking behavior under probabilistic constraints and developed several approximation models. In this chapter, we extend these chance-constrained models to energy system applications. The inherent variability of wind and solar PV generation poses significant risks for producers in day-ahead electricity markets, where operational commitments must be made before the actual output is realized. A common mitigation strategy is to invest in storage, but an alternative is to exploit the natural complementarity between wind and solar resources. We evaluate this economic value of cooperation using two stochastic optimization models with chance-constrained reliability guarantees: a Naive model without storage and a Storage-Enhanced model with battery dynamics. We evaluate the models using the European Meteorological derived High Resolution RES (EMHIRES) dataset for hourly wind and PV generation in 28 European countries in 2015. Our results show that cooperation consistently improves outcomes. Without storage, 24 of 28 countries achieve at least a 20% increase in expected profit, with gains in Finland, Lithuania, and Poland exceeding a seven-fold improvement. With storage included, absolute profits rise, but the relative advantage of cooperation becomes smaller and more uniform: joint profits are typically within 10–50% of the best independent operation. Cooperation remains most beneficial in countries like Belgium, Bulgaria, and the Netherlands, where wind and solar resources are both substantial and weakly negatively correlated. Methodologically, our work provides a demonstration of how chance-constrained optimization can evaluate renewable cooperation. For policymakers, our findings suggest that encouraging cooperative bidding can yield substantial and stable benefits in regions with balanced resources,

whereas in resource-scarce settings, expanding capacity or scaling storage should take precedence.

Note: a preprint of this work is available at https://optimization-online.org/?p=31844.

7.1 Introduction

The inherently intermittent nature of solar photovoltaic (PV) and wind power poses significant challenges for the reliable operation of electricity markets (Aflaki and Netessine, 2017; Glenk and Reichelstein, 2020; Piel et al., 2017). Solar output depends on diurnal and seasonal cycles, while wind generation is governed by complex meteorological dynamics that vary across space and time (Hu et al., 2015; Wolff et al., 2023). These fluctuations introduce risk in day-ahead electricity markets, where producers must commit to hourly energy deliveries before actual generation is realized. Failing to meet committed bids can result in financial penalties, reputational damage, or even the loss of future contracts (Sunar and Birge, 2019).

In operations and finance, a common strategy to manage uncertainty is to combine multiple sources of risk in a way that reduces overall exposure. This is especially effective when the uncertainties are negatively correlated or offer substitutable value. For example, in assortment planning, Kaut et al. (2021) show that pairing products with negatively correlated demand profiles can reduce the monetary portfolio risk. In financial hedging, Gaur and Seshadri (2005) illustrate how stock index options can protect against variable downturns when product demand is positively correlated with market performance. In manufacturing, Jordan and Graves (1995) demonstrate that process flexibility enables firms to shift capacity toward high-demand items, thus buffering against demand volatility across products. In all of these examples, the ability to combine or substitute across uncertain components reduces variance of the outcomes towards greater reliability.

This risk mitigation principle finds relevance to renewable energy systems as well. Wind and solar power exhibit partially complementary generation patterns — solar production peaks during daylight hours and declines at night, while wind power often intensifies in the evening or during non-summer months (we quantify this effect in Section 7.3). When these sources are negatively correlated, aggregating them can smooth total output and reduce the likelihood of simultaneous low generation. This complementarity has motivated the design of hybrid systems that co-locate wind and PV technologies (Ferrer-Martí et al., 2013; Golari et al., 2017; Salman et al., 2025), improving dispatchability and reducing backup requirements.

Recent studies have demonstrated both environmental and economic benefits of hybridizing wind and solar resources. For instance, Ferrer-Martí et al. (2013) formulate a mixed-integer program to co-optimize wind and PV in off-grid microgrids and report a 30.7% reduction in total capital cost. Golari et al. (2017) integrate intermittent renewable energy sources into a multi-plant, multi-stage production-inventory model and find that operational costs decrease by over 50% in a 12-stage single-plant setting and nearly 10% in a six-stage multi-plant setting. Their results also show that achieving a high green energy coefficient (the share of

renewable energy in total energy consumption) is feasible and effective in reducing carbon footprint. Similarly, Atakan et al. (2022) use a stochastic hierarchical planning framework to manage the uncertainty introduced by wind and solar, showing that renewable integration can reduce generation costs by up to 10.4% and lower daily CO_2 emissions. Glenk and Reichelstein (2020) analyze the investment value of hybrid wind and power-to-gas (P2G) systems, providing necessary and sufficient conditions under which joint deployment yields a synergistic benefit in terms of life-cycle unit costs. Empirical case studies further support these findings. In Urumqi, China, Li et al. (2013) show that hybrid wind–PV–battery systems reduce net present cost (NPC) by 9–11% relative to PV-battery and wind-battery systems. In Bangladesh, Nandi and Ghosh (2010) report even greater gains, with NPC reductions of 27% and 29% compared to wind-battery and PV-battery systems, respectively. Collectively, this extensive literature confirms that well-designed hybrid systems can significantly enhance the technical, economic, and environmental performance of renewable energy deployment.

Our work is in the same spirit as (Glenk and Reichelstein, 2020; Li et al., 2013; Nandi and Ghosh, 2010), but differs in focus and scope. Rather than evaluating investment decisions or long-term planning models, we study the operational coordination of a network of *independent* wind and solar producers in the context of the day-ahead electricity market. To model real-time delivery risk under uncertainty, we formulate stochastic optimization models with chance constraints, allowing a controlled probability of delivery shortfalls—an inherent feature of markets based on variable generation. A stochastic optimization framework is particularly well-suited for such an analysis, as it allows decision-making under uncertainty using simulations of scenario data. Further, chance constraints offer a flexible tool to enforce delivery reliability: operators can specify a desired confidence level (e.g., 95%) and ensure that energy commitments are met with high probability. This avoids the overly conservative behavior often associated with worst-case or deterministic models, while still providing reliability guarantees in uncertain environments. Employing this framework, we then quantify the economic benefits of joint operation and compare them to the profits from operating wind and solar assets independently. Our goal is to assess whether decentralized, market-based cooperation leads to measurable synergy, and to identify the operational conditions under which such cooperation is most effective.

With this background, the following are the key contributions of this work.

(i) We show that complementary generation profiles of wind and PV can substitute for storage. In many European regions, joint operation provides substantial synergy even without batteries — particularly in countries such as the Netherlands, the United Kingdom, and Germany where wind and solar outputs

are negatively correlated — allowing natural complementarity to smooth delivery risk.

(ii) We find that although the value of cooperation is reduced when battery storage is introduced, it is not entirely eliminated. Storage tends to diminish the variability benefit of complementarity, shifting the advantage toward countries with more balanced or weakly correlated resources. In some cases, storage substitutes for cooperation, while in others it enhances synergy through temporal arbitrage.

(iii) We demonstrate that even without physical integration such as co-location or shared storage, simple coordination among independent operators generates significant economic gains. This suggests that market-based incentives for cooperative bidding can be as important as infrastructure investment.

(iv) Finally, at a methodological level, we present a case-study demonstrating that chance constraints provide a flexible tool for evaluating renewable energy cooperation. By varying the reliability threshold, we quantify how tolerant a system must be to delivery shortfalls in order for cooperation to remain profitable. This offers guidance for the calibration of imbalance penalties and reliability requirements in market design.

The rest of the article is structured as follows. Section 7.2 introduces the modeling framework and defines our surrogate measure for economic synergy. In Section 7.3 we summarize our dataset, provide an exploratory analysis of wind and solar generation across Europe, and outline the forecasting procedure that we use to generate our scenario sets. Section 7.4 reports the computational experiments and evaluates the performance of both models that we present, including sensitivity analyses with respect to battery capacity and reliability thresholds. Finally, Section 7.5 highlights the main insights and discusses their implications for renewable energy policy and market design.

7.2 Problem Setting

We study the economic value of cooperation between wind and solar producers participating in the day-ahead electricity market. The central question is whether these resources achieve higher profits when operating jointly rather than independently, and how this outcome depends on the presence of storage and on system reliability requirements. To capture the uncertainty of renewable generation, we employ chance-constrained optimization models that maximize expected profit while ensuring that committed hourly deliveries are met with high probability. The

input to these models is a set of day-ahead forecast scenarios, generated from historical data as described in Section 7.3.2.

Our first framework, referred to as the *Naive* model, excludes storage and assumes that producers must commit to a fixed hourly delivery schedule before actual generation is realized. The model builds on the chance-constrained formulation in Singh et al. (2018), which analyzed the role of a fast-ramping generator in hedging renewable shortfalls. Let T denote the set of hours in the planning horizon, and let Ω denote the set of scenarios representing possible renewable generation outcomes. For each hour $t \in T$, the decision variable y_t denotes the committed delivery, while R_t represents the profit per unit of delivered energy. The available renewable generation in scenario $\omega \in \Omega$ is given by s_t^ω. The Naive model is formulated as

$$\max \quad \sum_{t \in T} R_t \, y_t \tag{7.1a}$$

$$\text{s.t.} \quad \mathbb{P}\left(y_t \leq s_t^\omega, \ \forall \, t \in T\right) \geq 1 - \varepsilon \tag{7.1b}$$

$$y_t \geq 0, \quad \forall \, t \in T. \tag{7.1c}$$

The chance constraint in (7.1b) is a *joint* reliability requirement over the full day: the entire vector of commitments $[y_t]_{t \in T}$ is covered by available generation $[s_t^\omega]_{t \in T}$ with probability at least $1 - \varepsilon$. Equivalently, at most an ε fraction of possible day-ahead realizations may experience any shortfall at any hour. Smaller ε yields more conservative commitments and lower imbalance risk, while larger ε allows more aggressive bidding at the expense of occasional shortfalls. In our experiments we consider $\varepsilon \in \{0.01, 0.05\}$, corresponding to 99% and 95% daily reliability, respectively.

Building on this baseline model, the second framework extends the formulation by introducing a battery system that can charge and discharge to buffer renewable variability and improve delivery reliability. This *Storage-Enhanced* model is closely related to the chance-constrained approach in Singh and Knueven (2021), which studied the effect of battery integration for solar-based bidding. Let p_t^ω and q_t^ω denote the charge and discharge decisions, respectively, in hour t under scenario ω, and let x_t^ω denote the battery state of charge. The battery operates with round-trip efficiency η and is bounded between $\underline{X}$ and $\overline{X}$; charging and discharging incur per-unit costs C_c and C_d. The model is given by

$$\max \quad \sum_{t \in T} R_t \, y_t - \mathbb{E}\left[C_c \, p_t^\omega + C_d \, q_t^\omega\right] \tag{7.2a}$$

$$\text{s.t.} \quad \mathbb{P}\left(y_t \leq s_t^\omega + q_t^\omega - p_t^\omega, \; \forall \, t \in T\right) \; \geq \; 1 - \varepsilon, \tag{7.2b}$$

$$x_{t+1}^\omega = x_t^\omega + \eta \, p_t^\omega - \tfrac{1}{\eta} \, q_t^\omega, \quad \forall \, t = 1, \ldots, |T| - 1, \; \forall \, \omega \in \Omega, \tag{7.2c}$$

$$\underline{X} \leq x_t^\omega \leq \overline{X}, \quad \forall \, t \in T, \; \forall \, \omega \in \Omega, \tag{7.2d}$$

$$y_t, \; p_t^\omega, \; q_t^\omega \geq 0, \quad \forall \, t \in T, \; \forall \, \omega \in \Omega. \tag{7.2e}$$

In model (7.2), we apply the same joint reliability level $1 - \varepsilon$ after accounting for storage. Further, the expectation in (7.2a) is taken with respect to the scenario distribution (implemented as a sample average approximation over Ω in our experiments). In optimal solutions of model (7.2), charging and discharging do not occur simultaneously (Singh and Knueven, 2021).

To implement the chance constraints in both models, we use a standard sample average approximation reformulation. Specifically, we introduce a binary variable z^ω for each scenario $\omega \in \Omega$, where $z^\omega = 1$ if the reliability constraint is violated in scenario ω. Then, each original chance constraint is replaced by scenario-specific linear constraints, augmented with a big-M term involving z^ω. Additionally, we enforce that at most $\lfloor N\varepsilon \rfloor$ violations occur across the $|\Omega|$ scenarios. This yields a mixed-integer linear program that we solve for each country and each reliability level.

We apply both models to three configurations: PV-only, wind-only, and joint wind–solar operation. Let z^{PV}, z^{W}, and z^{C} denote the optimal profits obtained by solving either (7.1) or (7.2) for these cases. To evaluate the economic benefit of cooperation, we define the *Synergy Ratio* as

$$\mathrm{SR} \; = \; \frac{z^{\mathrm{C}}}{z^{\mathrm{W}} + z^{\mathrm{PV}}}. \tag{7.3}$$

A value of $\mathrm{SR} > 1$ indicates that joint operation of wind and solar achieves higher profit than the sum of independent operations, thereby quantifying the economic value of complementarity. This performance measure provides the link between the models and the computational analysis that we conduct in Section 7.3.

7.3 Data Description

7.3.1 Summary of Renewable Generation Data

We use the EMHIRES dataset (European Meteorologically derived High-resolution Renewable Energy Sources time series (Gonzalez Aparicio et al., 2016, 2017)) that

provides hourly time series of onshore wind and PV generation across 35 European countries from January 1986 to December 2015. Each time series is reported as a *capacity factor* — the ratio of actual output in a given hour to the installed capacity — that takes values between zero and one. For this work, we focus on the year 2015 and restrict attention to the 28 countries for which complete capacity data and normalization parameters are available. Since our aim is to compare profitability across countries, we normalize the hourly capacity-factor profiles using the annual generation reported by each country's transmission system operator (TSO) for 2015. Multiplying the normalized capacity factors by installed capacities yields hourly generation in megawatts (MW), replicating each country's official annual production statistics. This normalization procedure is detailed in the appendix of Gonzalez Aparicio et al. (2016, 2017), and the final dataset used in our numerical experiments of Section 7.4 is available at our GitHub repository: `https://github.com/montreeklim/synergy_wind_solar`. In all tables and figures of this work, we identify countries by their ISO 3166-1 alpha-2 codes (e.g., DE = Germany, FR = France, etc.)

Table 7.1 summarizes the key descriptive statistics of hourly PV and wind generation across the 28 countries. For each technology, we report the mean, standard deviation, range (minimum and maximum), and the proportion of hours with zero output. We also calculate the PV share, defined as the ratio of average PV generation to the sum of average PV and wind generation, in order to characterize the balance between the two resources. The numbers show clear differences across the European countries. PV output follows a strict diurnal cycle, with nearly half of all hours producing zero output in most countries. Mean PV generation ranges from virtually zero in Ireland to nearly 4,000 MW in Germany. Wind generation, in contrast, is more continuous: in most countries, fewer than 5% of hours have zero output. Mean annual wind generation spans from below 1 MW in Slovenia and Slovakia to nearly 10,000 MW in Germany. The PV share is below 1% in northern countries such as Estonia, Finland, Ireland, Norway, Poland, and Sweden, while exceeding 98% in Slovenia and Slovakia.

Figures 7.1a and 7.1b illustrate that in most countries average wind generation exceeds PV generation, with exceptions such as Italy, Greece, Bulgaria, the Czech Republic, and Switzerland. There are five countries (Ireland, Finland, Norway, Estonia, and Latvia) that have an average PV generation of less than 1 MW, and two countries (Slovenia and Slovakia) that have an average wind generation of less than 1 MW. Figure 7.1c and 7.1d further highlight the geographic distribution of resources: PV capacity is concentrated around the Mediterranean basin, where solar irradiance is high and sunshine is predictable, whereas wind capacity is concentrated in northwestern Europe, particularly Germany, the United Kingdom, and Spain, where strong and persistent wind regimes prevail (we revisit this effect below). Finally, Figure 7.2 displays normalized hourly generation profiles, confirming the

TABLE 7.1: Descriptive statistics of hourly PV and wind generation in 2015, derived from EMHIRES (Gonzalez Aparicio et al., 2016, 2017) and scaled to MW using country-level installed capacities reported by national transmission system operators (TSOs). The columns "Mean", "SD", "Range", and "%Z" denote the hourly mean, standard deviation, minimum–maximum range, and percentage of zero-output hours, respectively. The final column reports the PV share, defined as the ratio of average PV generation to the sum of average PV and wind generation. For details, see Section 7.3.1.

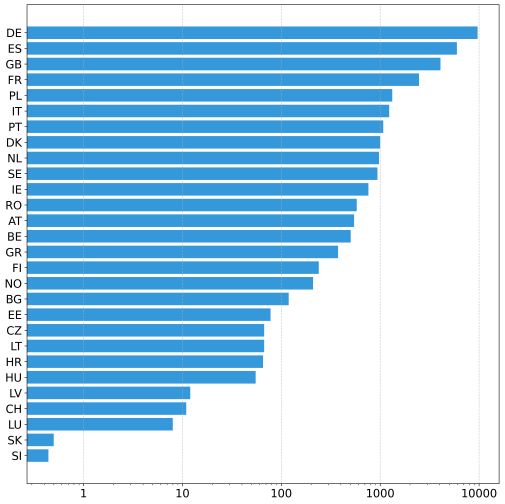
	PV (MW)				Wind (MW)				PV share (%)
Country	Mean	SD	Range	%Z	Mean	SD	Range	%Z	
AT	52.7	74.5	[0.0–287.6]	49.8	544.9	573.7	[0.0–1980.2]	0.0	8.8
BE	361.5	573.0	[0.0–2443.7]	49.9	503.2	504.7	[0.0–1726.0]	0.4	41.8
BG	152.1	226.4	[0.0–909.8]	49.4	118.8	126.5	[0.0–615.3]	0.1	56.1
CH	103.5	146.5	[0.0–551.1]	49.7	10.9	13.8	[0.0–60.0]	1.4	90.5
CZ	250.5	394.1	[0.0–1712.1]	49.4	67.1	74.7	[0.0–276.6]	0.6	78.9
DE	3922.2	5973.7	[0.0–25,535.6]	49.3	9670.2	10107.9	[10.4–42,757.6]	0.0	28.9
DK	83.0	135.7	[0.0–617.0]	49.6	999.7	844.9	[0.0–2837.1]	0.1	7.7
EE	0.5	1.0	[0.0–4.5]	50.6	77.8	65.2	[0.0–204.4]	0.4	0.6
ES	1220.1	1569.3	[0.0–5178.6]	49.5	5962.3	5054.0	[9.4–22,961.8]	0.0	17.0
FI	0.7	1.3	[0.0–6.3]	49.9	239.1	184.8	[0.7–679.8]	0.0	0.3
FR	828.1	1132.5	[0.0–4214.3]	49.9	2471.1	2031.9	[4.9–8683.0]	0.0	25.1
GB	853.2	1314.0	[0.0–6692.4]	49.3	4071.4	3031.0	[33.0–10,706.2]	0.0	17.3
GR	420.5	584.2	[0.0–2110.9]	49.8	375.1	349.0	[0.4–1517.5]	0.0	52.9
HR	5.8	8.9	[0.0–32.2]	49.4	65.4	83.2	[0.0–366.9]	4.4	8.1
HU	4.1	5.9	[0.0–21.1]	49.2	55.0	67.0	[0.0–263.4]	2.9	6.9
IE	0.1	0.1	[0.0–0.8]	49.7	759.9	583.3	[0.0–1759.9]	0.1	0.0
IT	2727.8	3819.8	[0.0–13,331.8]	49.6	1232.3	1396.9	[0.7–7444.8]	0.0	68.9
LT	6.7	11.5	[0.0–50.7]	50.3	67.1	66.9	[0.0–235.0]	0.4	9.1
LU	13.4	22.2	[0.0–95.6]	50.8	8.0	11.2	[0.0–48.4]	8.0	62.6
LV	0.2	0.3	[0.0–1.5]	50.2	12.0	11.9	[0.0–38.6]	2.7	1.6
NL	163.5	258.1	[0.0–1128.8]	49.9	972.4	850.1	[0.0–2594.1]	0.4	14.4
NO	0.7	1.3	[0.0–6.8]	49.2	209.5	124.5	[2.2–498.3]	0.0	0.3
PL	9.4	14.8	[0.0–66.2]	49.5	1324.5	1245.4	[1.2–4688.8]	0.0	0.7
PT	91.2	121.5	[0.0–388.4]	49.6	1073.9	968.5	[0.6–4506.3]	0.0	7.8
RO	178.0	250.9	[0.0–889.7]	49.6	579.7	568.3	[0.1–2859.6]	0.0	23.5
SE	7.8	13.1	[0.0–63.3]	49.5	937.4	752.5	[8.1–3018.3]	0.0	0.8
SI	32.2	51.6	[0.0–193.5]	50.1	0.4	0.7	[0.0–3.0]	17.5	98.8
SK	69.8	102.6	[0.0–383.8]	49.6	0.5	0.7	[0.0–2.6]	17.1	99.3

complementary temporal patterns of the two technologies: solar output peaks during daylight hours while wind generation tends to be stronger at night or during off-peak solar periods.

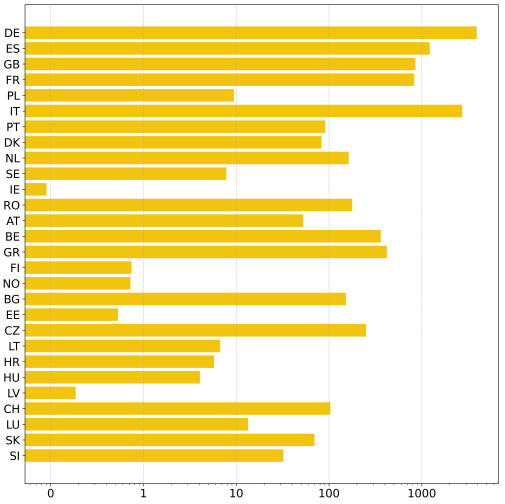
To examine complementarity more directly, we compute the Pearson correlation coefficient between hourly PV and wind generation during daytime hours (05:00–17:00). A negative correlation indicates that when one source produces strongly, the other tends to be weak, creating natural hedging opportunities; conversely, a positive correlation suggests that wind and solar outputs rise and fall together, offering less diversification benefit. As shown in Figure 7.3a, northern maritime countries such as Germany, the United Kingdom, and Norway exhibit strongly negative correlations in sunlit hours (as low as -0.55), reflecting a high degree of complementarity. In contrast, countries such as Hungary and Portugal display correlations closer to zero, implying weaker natural balancing.

To explore temporal patterns further, we group countries into four climatological regions: Atlantic Maritime, Continental, Nordic & Baltic, and Mediterranean.

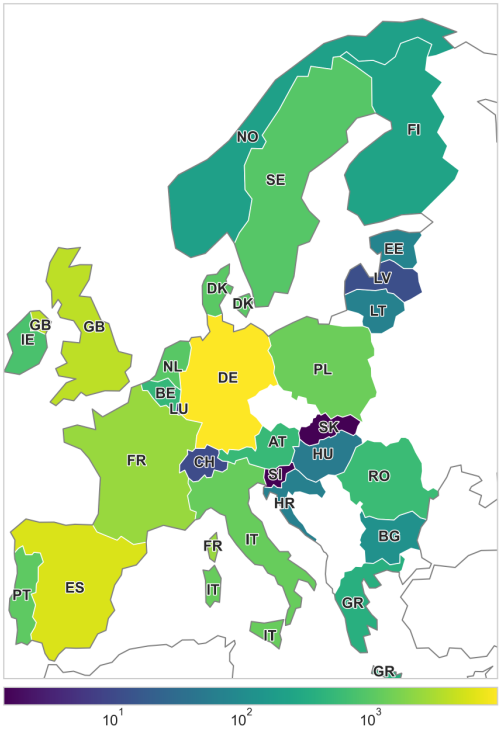
FIGURE 7.1: Average hourly wind and PV generation by country in 2015 (MW), shown on a logarithmic scale to accommodate cross-country magnitude differences, based on EMHIRES data (Gonzalez Aparicio et al., 2016, 2017). For details, see Section 7.3.



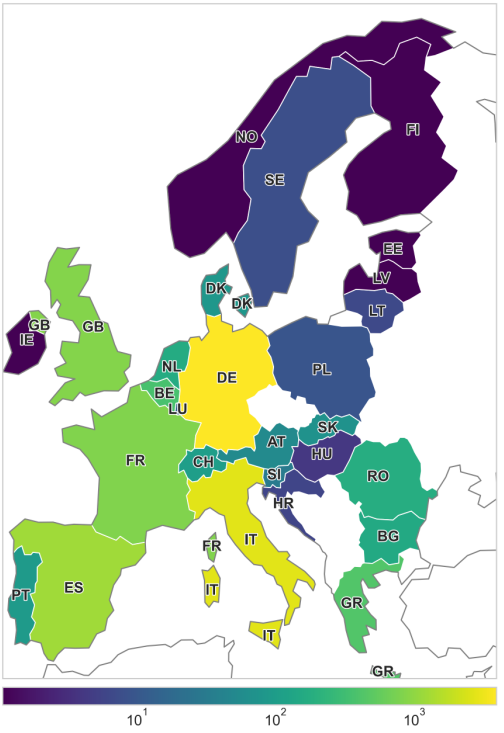
(A) Average hourly wind generation (bar chart).



(B) Average hourly PV generation (bar chart).



(C) Average hourly wind generation (heatmap).



(D) Average hourly PV generation (heatmap).

FIGURE 7.2: Intraday generation profiles of PV (orange), wind (blue), and their sum (green) for 28 European countries normalized by that country's peak combined hourly output. The x-axis denotes the hour of the day ranging from 0:00 to 23:00, while the y-axis provides the corresponding value. Data is based on the 2015 EMHIRES data (Gonzalez Aparicio et al., 2016, 2017). See Section 7.3 for details.

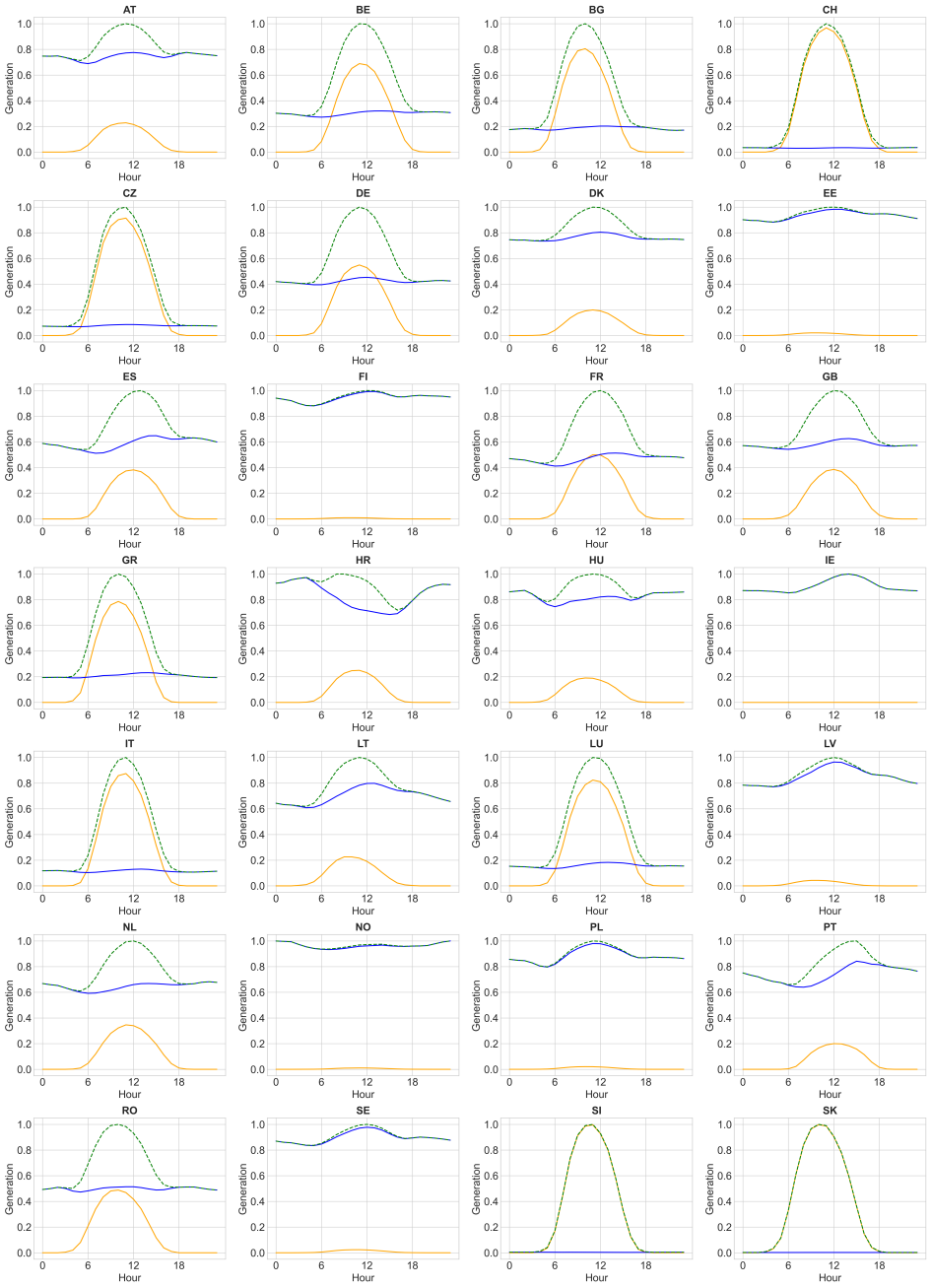
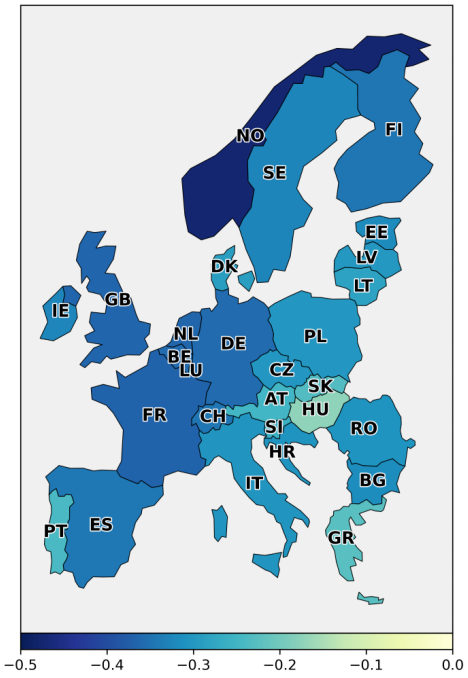
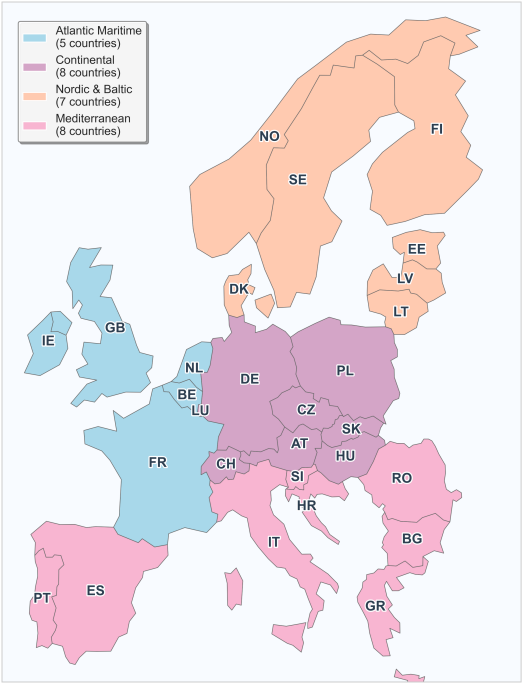


FIGURE 7.3: Spatial patterns of PV–wind correlation and climatological regionalization based on EMHIRES data (Gonzalez Aparicio et al., 2016, 2017). Daytime hours are defined as 05:00–17:00. For details, see Section 7.3.

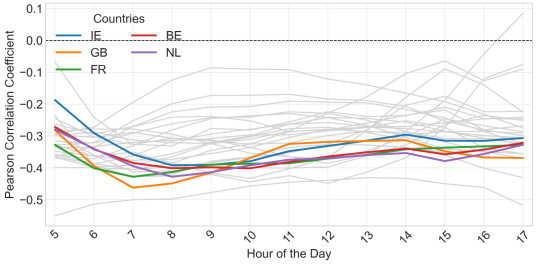


(A) Average hourly PV–wind correlation by country (daytime hours).

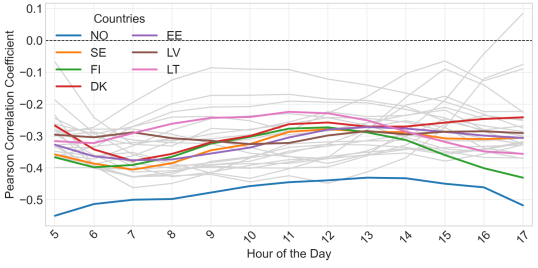


(B) European countries grouped by climate regions.

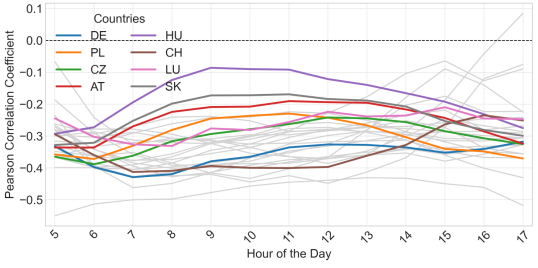
FIGURE 7.4: Hourly PV–wind correlation profiles for European countries, grouped by climatological region. The underlying data is from the 2015 EMHIRES dataset Gonzalez Aparicio et al. (2016, 2017). Colored lines represent countries within the focal region, while all other countries appear in gray. See Section 7.3 for details.



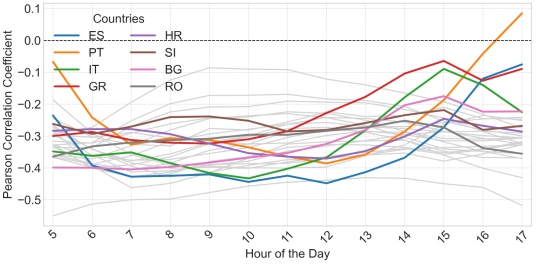
(A) Atlantic Maritime



(B) Nordic & Baltic



(C) Continental



(D) Mediterranean

Figure 7.4 provides further details by plotting hourly PV–wind correlations for each climatological region. The Atlantic Maritime group (Figure 7.4a) shows the weakest negative correlation in the early morning [05:00–06:00), with correlations strengthening around 07:00–08:00 and then remaining relatively stable throughout the rest of the day. In the Mediterranean group (Figure 7.4d), correlations are also weak in the early morning but decrease steadily toward noon, reaching their most negative values late morning before increasing again in the afternoon; in Portugal, correlations even turn positive by 17:00. The Nordic & Baltic and Continental groups (Figures 7.4b and 7.4c) exhibit similar patterns, with the strongest negative correlations occurring around [06:00–07:00), rising toward weaker values near midday, and then becoming more negative again in the evening hours. These regional patterns reinforce the view that the degree of complementarity between PV and wind varies systematically not only across geography but also across the intraday cycle.

7.3.2 Scenario Generation

To evaluate the models introduced in Section 7.2, we simulate scenarios of renewable generation uncertainty at the day-ahead horizon. Specifically, we use an autoregressive moving average (ARMA) model following the methodology given in Singh and Pozo (2019).

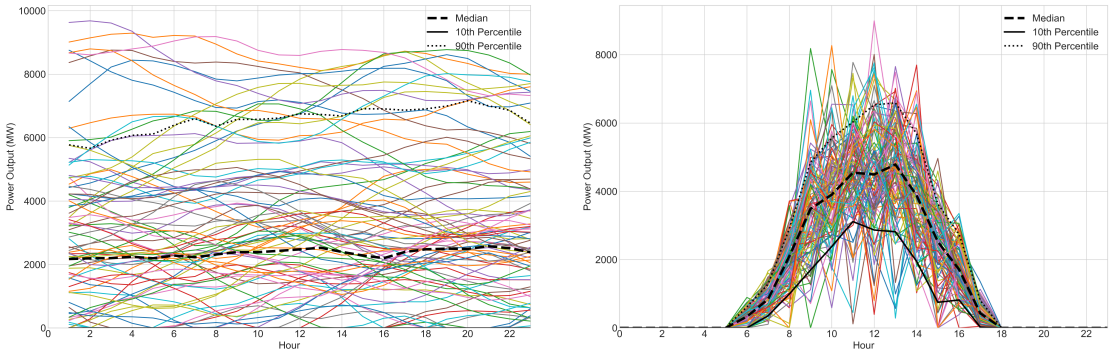
For wind generation, we fit an ARMA(p, q) model to the 60 days of data preceding the prediction date of September 28, 2015 (the 271st day of the year). We verify stationarity using the Augmented Dickey–Fuller (ADF) test, with the null hypothesis of a unit root rejected at the 5% level in all countries. We evaluate candidate lag orders $p, q \in \{0, \ldots, 5\}$ to select the model that minimizes the Bayesian Information Criterion (BIC), subject to passing Ljung–Box tests for residual autocorrelation at lags 5, 10, and 15. For four countries (Bulgaria, France, Greece, and Romania), no specification satisfies all Ljung–Box tests; in these cases, we select the model with minimum BIC, as our goal is scenario generation rather than best-in-class forecasting accuracy.

For PV generation, we fit separate ARMA models for each hour of the day, using the 270 days preceding September 28, 2015. For hours where output was zero in the previous seven days, we directly predict zero generation. We again verify stationarity with the ADF test, and difference the non-stationary series until stationarity is achieved. We then estimate ARMA(p, q) models with $p, q \in \{0, \ldots, 5\}$ and select lag orders using BIC, ensuring residuals do not exhibit significant autocorrelation. For differenced series, we transform back the forecast values by adding the last observed level of the original series.

Using these fitted models, we generate 10 independent batches of 100 Monte Carlo scenarios for each country and technology. Each scenario represents a 24-hour

trajectory of possible generation outcomes, truncated below at zero. For illustration, Figures 7.5a and 7.5b provide a representative set of 100 scenarios for the United Kingdom, with black lines indicating the 10th percentile, median, and 90th percentile. These scenarios capture the variability and uncertainty inherent in renewable generation and provide the basis for evaluating the performance of the Naive and Storage-Enhanced optimization models in Section 7.4.

FIGURE 7.5: Illustrative set of 100 hourly scenarios for renewable generation in the United Kingdom derived from ARMA models. The panels show (a) wind power and (b) solar PV output; dashed black lines indicate the median and solid black lines the 10th percentile across scenarios. For details, see Section 7.3.2.



(A) Wind generation

(B) PV generation

7.4 Computational Experiments

In this section, we present numerical experiments conducted using the models described in Section 7.2. The input consists of hourly wind and solar generation scenarios derived from the EMHIRES dataset (Section 7.3). For each of the 28 countries, we evaluate both the Naive and Storage-Enhanced models under reliability levels $1 - \varepsilon \in \{0.95, 0.99\}$, with 10 independent sets of 100 scenarios each. We run all computational experiments on a computing cluster with an Intel Xeon Gold 6138 2.0 GHz processor, 192 GB of RAM, and Gurobi 10.0.3 via Python 3.8.3. Each instance has a time limit of one hour.

We present our analysis in three parts. In Section 7.4.1, we first report results for the Naive model, highlighting the value of complementarity in the absence of storage. Section 7.4.2 examines how battery integration alters profits and synergy patterns. Section 7.4.3 presents sensitivity analyses with respect to storage capacity and reliability thresholds. Throughout, we interpret results in terms of the Synergy Ratio, defined in equation (7.3), which serves as our measure of the economic value of cooperation.

7.4.1 Analysis of the Naive Model: No Storage

Table 7.2 presents the 95% confidence intervals for the Synergy Ratio; our numbers exclude experiments where the synergy ratio is undefined. Under the Naive model (columns 2–3), requiring strict reliability ($\varepsilon = 0.01$) produces an undefined ratio in the Czech Republic, Estonia, Finland, Lithuania, Latvia, and Slovenia due to zero profits in both individual sources but a positive value for combined sources. Omitting these degenerate cases, Poland and Croatia have the highest Synergy Ratio with confidence intervals of $(14.27, 31.01)$ and $(1.77, 31.64)$, respectively. Slovakia has the smallest profit gains with a confidence interval of $(1.02, 1.07)$. Overall, the minimum ratio for all countries is greater than one, which indicates that combining wind and PV generation provides more profit than individual operation. However, we note that this ratio is highly dependent on the generated scenarios, e.g., in Ireland with $\varepsilon = 0.01$, the confidence interval range is between -8.20 to 139.29 with a negative lower bound value due to the large standard deviation. With the reliability threshold of $\varepsilon = 0.01$, the results are not strongly conclusive due to the relatively wide confidence intervals.

Relaxing the reliability threshold to $\varepsilon = 0.05$, narrows the confidence interval considerably suggesting more precise estimates. Specifically, the lower bounds fall into the tighter range of 0.99 to 8.85, while the upper bounds range from 1.02 to 30.54. This contraction occurs because allowing more scenario violations raises the minimum objective values, thereby reducing cases in which the profit from one resource is nearly zero. As a result, the relative gains from cooperation become more stable and easier to compare across countries, with less dependence on individual scenario variation. In this setting, the largest gains occur in Finland, Lithuania, and Poland, where joint operation yields at least 7.5 times the profit of independent wind or solar systems. By contrast, the lowest ratios remain in Latvia, Norway, Slovenia, and Slovakia, where values hover just above one indicating only marginal improvement over independent operation. Notably, Norway and Latvia have PV shares below two percent, while Slovenia and Slovakia have PV shares exceeding 98%. This pattern suggests that when one resource contributes only a negligible share of total generation, cooperation may not yield meaningful additional value.

TABLE 7.2: 95% confidence intervals defined over 10 samples for the Synergy Ratio, see equation (7.3), for 28 European countries for both the Naive and Storage-Enhanced models. Entries with "–" indicates that the synergy ratio is undefined for at least nine out of the 10 samples due to zero profits from both wind and PV generation sources but non-zero profits from the combination. For details, see Section 7.4.1.

Country	**Naive**		**Storage-Enhanced**	
	$\varepsilon = 0.01$	$\varepsilon = 0.05$	$\varepsilon = 0.01$	$\varepsilon = 0.05$
AT	(2.66, 8.02)	(3.05, 4.39)	(1.30, 1.39)	(1.18, 1.29)

continued on next page...

Table 7.2 (continued)

Country	Naive		Battery	
	$\varepsilon = 0.01$	$\varepsilon = 0.05$	$\varepsilon = 0.01$	$\varepsilon = 0.05$
BE	(3.76, 7.32)	(3.03, 4.04)	(1.30, 1.38)	(1.30, 1.38)
BG	(1.75, 3.49)	(1.58, 1.97)	(1.25, 1.36)	(1.29, 1.43)
CH	(1.43, 3.39)	(1.27, 1.39)	(1.06, 1.09)	(1.08, 1.10)
CZ	(–, –)	(1.40, 1.78)	(1.14, 1.21)	(1.15, 1.18)
DE	(1.51, 1.99)	(1.60, 1.96)	(1.19, 1.28)	(1.27, 1.36)
DK	(2.88, 3.88)	(2.70, 3.57)	(1.21, 1.28)	(1.24, 1.30)
EE	(–, –)	(2.77, 4.94)	(1.05, 1.08)	(1.04, 1.07)
ES	(1.26, 1.38)	(1.31, 1.42)	(1.14, 1.19)	(1.14, 1.20)
FI	(–, –)	(7.75, 16.38)	(1.02, 1.03)	(1.02, 1.03)
FR	(1.69, 2.09)	(1.72, 1.98)	(1.22, 1.27)	(1.20, 1.27)
GB	(2.16, 3.04)	(2.01, 2.40)	(1.25, 1.31)	(1.21, 1.28)
GR	(1.33, 1.45)	(1.31, 1.41)	(1.18, 1.23)	(1.20, 1.24)
HR	(1.77, 31.64)	(4.08, 6.34)	(1.17, 1.21)	(1.23, 1.27)
HU	(1.66, 9.76)	(2.57, 3.53)	(1.16, 1.20)	(1.18, 1.26)
IE	(-8.20, 139.29)	(1.78, 4.22)	(1.00, 1.00)	(1.00, 1.00)
IT	(1.25, 1.40)	(1.23, 1.33)	(1.12, 1.16)	(1.12, 1.16)
LT	(–, –)	(8.85, 30.54)	(1.36, 1.46)	(1.29, 1.42)
LU	(2.19, 8.09)	(2.22, 3.31)	(1.23, 1.35)	(1.27, 1.35)
LV	(–, –)	(0.99, 1.52)	(1.07, 1.10)	(1.07, 1.10)
NL	(3.10, 4.33)	(3.05, 3.94)	(1.39, 1.51)	(1.29, 1.39)
NO	(2.50, 8.12)	(1.05, 1.17)	(1.01, 1.04)	(1.01, 1.01)
PL	(14.27, 31.01)	(7.64, 10.62)	(1.05, 1.06)	(1.03, 1.05)
PT	(1.39, 1.60)	(1.40, 1.55)	(1.12, 1.20)	(1.05, 1.09)
RO	(-0.13, 11.03)	(2.20, 3.86)	(1.31, 1.42)	(1.31, 1.44)
SE	(2.49, 4.81)	(2.35, 4.03)	(1.07, 1.10)	(1.06, 1.09)
SI	(–, –)	(1.02, 1.06)	(1.01, 1.02)	(1.01, 1.02)
SK	(1.02, 1.07)	(1.01, 1.02)	(1.00, 1.01)	(1.01, 1.01)

7.4.2 Analysis of the Storage-Enhanced Model: With Storage

Table 7.2 (columns 4–5) reports the 95% confidence intervals for the Synergy Ratio under the Storage-Enhanced model with battery capacity set equal to the installed capacity for each resource (i.e., the maximum generation in an hour). In contrast to the Naive model, all countries yield defined ratios in this setting because the presence of a battery ensures strictly positive profits for both wind-only and PV-only operations, even under the stringent reliability requirement of $\varepsilon = 0.01$. As a result, the confidence intervals are markedly tighter and extreme values disappear. For $\varepsilon = 0.01$, the lower bounds range from 1.00 (Ireland) to 1.39 (the Netherlands), while the upper bounds range from 1.00 (Ireland) to 1.51 (the Netherlands). Relaxing the threshold to $\varepsilon = 0.05$ yields a very similar pattern, with lower bounds between 1.00 and 1.31 and upper

bounds between 1.00 and 1.44. This stability across reliability levels reflects how storage dampens the sensitivity of profits to forecast uncertainty.

However, the presence of storage reduces the Synergy Ratio in most countries. For example, Finland and Poland, which achieved Synergy Ratios well above seven in the Naive model, fall below 1.06 under the Storage-Enhanced model. The reason is that when one resource is scarce (e.g., PV in northern regions or wind in Slovenia and Slovakia), the battery substitutes for the missing complementarity, leaving little additional benefit from cooperation. In these cases, the ratio drops close to one, indicating that joint operation does not improve profit relative to independent participation significantly. Despite this, several countries retain Synergy Ratios above 1.25 even with storage. Belgium, Bulgaria, Lithuania, the Netherlands, and Romania all lie in this category, with ratios between 1.25 and 1.51. The presence of storage in these regions enhances the ability to exploit temporal arbitrage opportunities, allowing cooperation to remain economically beneficial. In countries with strongly complementary profiles (e.g., France, the Netherlands, the United Kingdom, Germany), Synergy Ratios are much lower compared to the Naive model and tend to decrease further as battery capacity increases, indicating that storage substitutes for natural balancing of renewable generation, see Section 7.4.3.

There are country-specific caveats in the above analysis. While Norway exhibits one of the most strongly negative PV–wind correlations, its Synergy Ratio remains low. In other words, even though PV and wind in Norway tend to alternate (i.e., high wind when solar is low, and vice versa), the economic gain from combining them is negligible because Norwegian solar output is extremely small compared to wind. This interpretation is reinforced by the cases of Slovenia and Slovakia (where the PV share is more than 98%), and by Finland, Ireland, and Norway (where the PV share is less than 0.5%). In each of these cases, the scarcity of one resource limits the potential for meaningful synergy. Similarly, the ratio for Ireland is not high, even though it shares correlation patterns with high-SR countries because its PV share is very low.

Summarizing, our results lead to two takeaways. First, storage systematically increases total profits, but it also reduces the incremental value of natural complementarity in highly correlated systems by providing an alternative buffer against variability. In these cases, storage acts as a substitute for the natural balancing between wind and solar. As capacity grows the incremental benefit of cooperation diminishes further. Second, the relative ordering of countries by Synergy Ratio remains stable across reliability thresholds, highlighting that the economic role of storage is not to eliminate complementarity, but to redistribute its value across regions with different resource profiles.

7.4.3 Sensitivity Analysis

Next, we examine how the value of cooperation depends on three factors: battery capacity, reliability thresholds, and the correlation between wind and solar generation. We also report average profit outcomes to properly place the Synergy Ratios in an economic context.

In the Storage-Enhanced model (7.2), the battery frequently serves as the primary instrument for arbitraging across hours with pronounced price differences. For instance, in several countries the optimal policy at hour 17:00 is to discharge the battery to its lower bound as the market return during that hour outweighs the cycling cost. This pattern suggests that benchmarking battery size to the installed renewable capacity of each setting may understate the potential benefits of cooperation. To investigate this effect, we re-ran the combined-system experiments on a dataset different from that used in Table 7.2, varying battery capacity to be either one-half or twice the installed capacity in each case.

Table 7.3 shows that reducing battery capacity markedly increases Synergy Ratios. With capacity set to one-half of installed capacity, confidence intervals widen for most countries, reflecting greater variability. Under $\varepsilon = 0.01$, Lithuania, the Netherlands, and Romania attain the highest Synergy Ratios, each reaching at least 1.47. Under $\varepsilon = 0.05$, Lithuania, Romania, and Belgium again lead with ratios of at least 1.40. By contrast, Finland, Ireland, Norway, Slovakia, and Slovenia record the lowest ratios converging to 1.00 across all battery sizes. This suggests that in these regions the Storage-Enhanced model offers little advantage over independent wind–battery and PV–battery systems. Doubling battery capacity produces the opposite effect: Synergy Ratios decline but become more stable, with narrower confidence intervals. Thus, smaller storage capacities raise the measured value of wind–solar complementarity, as cooperation compensates for the reduced buffering from storage, but this comes at the cost of wider uncertainty ranges. In extreme cases, further reducing capacity could replicate the volatility observed in the Naive model.

Sensitivity to the reliability parameter ε is much less pronounced in the Storage-Enhanced model than in the Naive model. Table 7.2 shows that relaxing the reliability requirement from $\varepsilon = 0.01$ to $\varepsilon = 0.05$ substantially narrows the confidence intervals in the Naive model (e.g., Poland from $(14.27, 31.01)$ to $(7.64, 10.62)$; Ireland from $(-8.20, 139.29)$ to $(1.78, 4.22)$). This contraction reflects the fact that higher tolerance reduces the frequency of near-zero denominators in the ratio. By contrast, confidence intervals under the Storage-Enhanced model change little between the two thresholds (e.g., Romania $(1.31, 1.42)$ to $(1.31, 1.44)$; Lithuania $(1.36, 1.46)$ to $(1.29, 1.42)$). These results confirm that storage buffers against variability not only in generation but also in the sensitivity of cooperative value to reliability requirements.

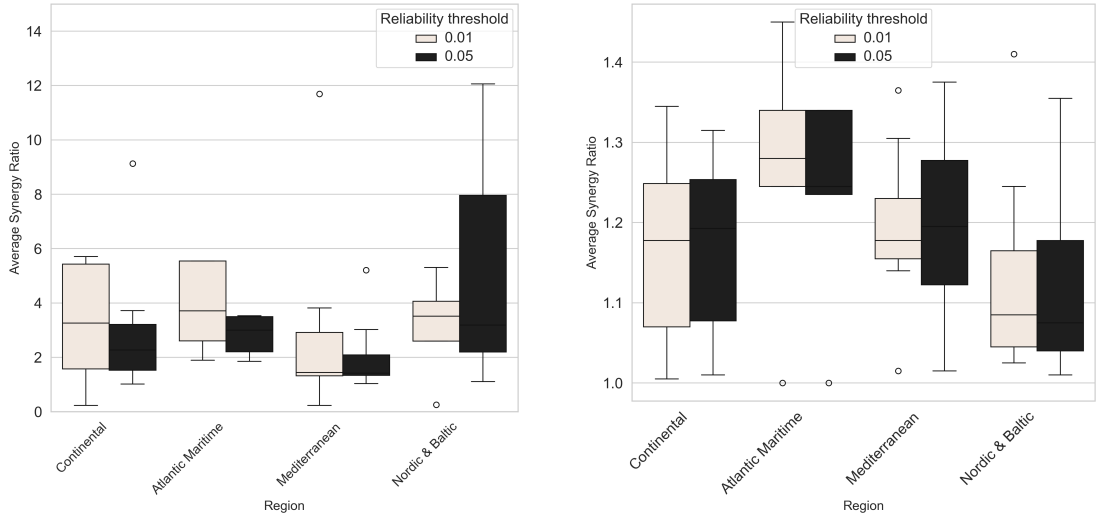
TABLE 7.3: Synergy Ratios, see equation (7.3), for 28 European countries under different battery capacities, for two reliability levels of 95% ($\varepsilon = 0.05$) and 99% ($\varepsilon = 0.01$). The Baseline column denotes the default battery size, while the Half Cap. and Double Cap. columns denote a battery size of half and double the installed capacity. For details, see Section 7.4.3.

Country	$\varepsilon = 0.01$			$\varepsilon = 0.05$		
	Half Cap.	Baseline	Double Cap.	Half Cap.	Baseline	Double Cap.
AT	(1.50, 1.62)	(1.30, 1.39)	(1.17, 1.22)	(1.21, 1.40)	(1.18, 1.29)	(1.12, 1.18)
BE	(1.44, 1.53)	(1.30, 1.38)	(1.21, 1.28)	(1.42, 1.51)	(1.30, 1.38)	(1.23, 1.29)
BG	(1.44, 1.55)	(1.25, 1.36)	(1.15, 1.24)	(1.45, 1.57)	(1.29, 1.43)	(1.19, 1.31)
CH	(1.10, 1.12)	(1.06, 1.09)	(1.05, 1.08)	(1.11, 1.13)	(1.08, 1.10)	(1.06, 1.08)
CZ	(1.22, 1.26)	(1.14, 1.21)	(1.11, 1.16)	(1.21, 1.27)	(1.15, 1.18)	(1.11, 1.14)
DE	(1.29, 1.41)	(1.19, 1.28)	(1.13, 1.20)	(1.36, 1.44)	(1.27, 1.36)	(1.19, 1.26)
DK	(1.39, 1.49)	(1.21, 1.28)	(1.12, 1.16)	(1.35, 1.47)	(1.24, 1.30)	(1.15, 1.18)
EE	(1.11, 1.16)	(1.05, 1.08)	(1.03, 1.04)	(1.06, 1.13)	(1.04, 1.07)	(1.02, 1.03)
ES	(1.20, 1.24)	(1.14, 1.19)	(1.09, 1.13)	(1.18, 1.24)	(1.14, 1.20)	(1.09, 1.14)
FI	(1.04, 1.06)	(1.02, 1.03)	(1.01, 1.02)	(1.03, 1.06)	(1.02, 1.03)	(1.01, 1.02)
FR	(1.30, 1.35)	(1.22, 1.27)	(1.15, 1.19)	(1.27, 1.33)	(1.20, 1.27)	(1.15, 1.20)
GC	(1.23, 1.28)	(1.18, 1.23)	(1.14, 1.18)	(1.24, 1.28)	(1.20, 1.24)	(1.16, 1.20)
HR	(1.36, 1.43)	(1.17, 1.21)	(1.09, 1.12)	(1.43, 1.50)	(1.23, 1.27)	(1.13, 1.15)
HU	(1.34, 1.42)	(1.16, 1.20)	(1.08, 1.11)	(1.30, 1.47)	(1.18, 1.26)	(1.10, 1.14)
IE	(1.00, 1.00)	(1.00, 1.00)	(1.00, 1.00)	(1.00, 1.00)	(1.00, 1.00)	(1.00, 1.00)
IT	(1.16, 1.20)	(1.12, 1.16)	(1.09, 1.13)	(1.14, 1.19)	(1.12, 1.16)	(1.10, 1.13)
LT	(1.60, 1.76)	(1.36, 1.46)	(1.21, 1.27)	(1.40, 1.67)	(1.29, 1.42)	(1.19, 1.25)
LU	(1.33, 1.47)	(1.23, 1.35)	(1.16, 1.26)	(1.35, 1.46)	(1.27, 1.35)	(1.19, 1.26)
LV	(1.14, 1.20)	(1.07, 1.10)	(1.03, 1.05)	(1.12, 1.18)	(1.07, 1.10)	(1.04, 1.05)
NL	(1.61, 1.76)	(1.39, 1.51)	(1.25, 1.33)	(1.38, 1.53)	(1.29, 1.39)	(1.21, 1.27)
NO	(1.02, 1.06)	(1.01, 1.04)	(1.01, 1.02)	(1.01, 1.02)	(1.01, 1.01)	(1.00, 1.01)
PL	(1.10, 1.13)	(1.05, 1.06)	(1.02, 1.03)	(1.04, 1.09)	(1.03, 1.05)	(1.02, 1.02)
PT	(1.20, 1.33)	(1.12, 1.20)	(1.07, 1.12)	(1.08, 1.14)	(1.05, 1.09)	(1.04, 1.06)
RO	(1.47, 1.67)	(1.31, 1.42)	(1.18, 1.24)	(1.46, 1.68)	(1.31, 1.44)	(1.19, 1.27)
SI	(1.02, 1.04)	(1.01, 1.02)	(1.01, 1.02)	(1.02, 1.03)	(1.01, 1.02)	(1.01, 1.02)
SK	(1.01, 1.02)	(1.00, 1.01)	(1.00, 1.01)	(1.01, 1.01)	(1.01, 1.01)	(1.00, 1.01)
SE	(1.15, 1.20)	(1.07, 1.10)	(1.04, 1.05)	(1.10, 1.17)	(1.06, 1.09)	(1.03, 1.05)
GB	(1.38, 1.45)	(1.25, 1.31)	(1.18, 1.22)	(1.29, 1.39)	(1.21, 1.28)	(1.17, 1.21)

Figure 7.6a illustrates that the box plot for the Synergy Ratio in the Naive model has a wider box for $\varepsilon = 0.01$ than for $\varepsilon = 0.05$ across all regions, except for the Nordic & Baltic region. In the latter, four of the six countries with an undefined ratio are excluded from the plot. The Nordic & Baltic region has the longest interquartile range and the highest median Synergy Ratio, whereas the Mediterranean region has the lowest values for both these measures. In the Storage-Enhanced model shown in Figure 7.6b, the difference in the box length between $\varepsilon = 0.01$ and $\varepsilon = 0.05$ is lesser compared to Figure 7.6a. The Atlantic Maritime region has the highest median for both of these reliability thresholds, whereas the Nordic & Baltic region has the lowest, despite having the highest median in the Naive model.

Figure 7.7 presents the relationship between PV–wind correlation and the average Synergy Ratio under the Storage-Enhanced model across three periods of the day. Correlation in the early morning is measured from 05:00 to 07:00, daytime from 07:00

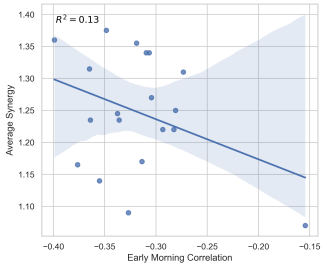
FIGURE 7.6: Synergy Ratio distributions by climatological region. The left panel shows box plots for the Naive model, and the right panel shows box plots for the Storage-Enhanced model. Both figures have different scales. For details, see Section 7.4.3.



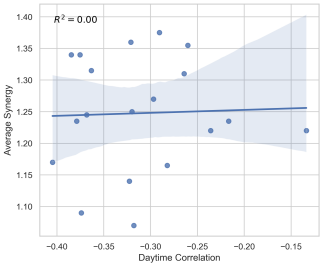
(A) Naive model.

(B) Storage-Enhanced model.

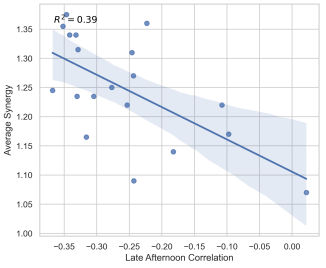
FIGURE 7.7: PV-wind correlation in each period versus average Synergy Ratio in the Storage-Enhanced model, excluding countries with a PV share below 0.5% or above 98%. Shaded bands denote 95% confidence intervals for the fitted regression lines.



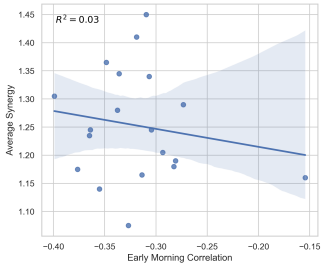
(A) Early Morning ($\epsilon = 0.01$)



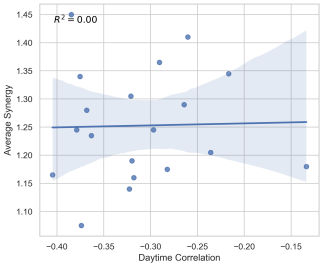
(B) Daytime ($\epsilon = 0.01$)



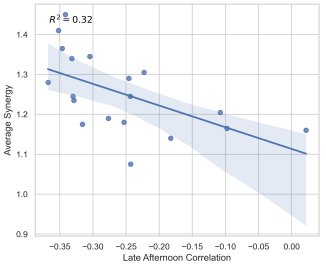
(C) Late Afternoon ($\epsilon = 0.01$)



(D) Early Morning ($\epsilon = 0.05$)



(E) Daytime ($\epsilon = 0.05$)



(F) Late Afternoon ($\epsilon = 0.05$)

to 16:00, and late afternoon from 16:00 to 18:00. We exclude countries with PV shares below 0.5% or above 95% since extreme scarcity of one resource renders cooperation uninformative. Fitted regressions yield $R^2 = 0.39$ for $\varepsilon = 0.01$ and $R^2 = 0.32$ for

$\varepsilon = 0.05$ in the late afternoon period, suggesting a weak but nontrivial association: more negative correlations in the late afternoon are coinciding with higher synergy. This pattern aligns with the intuition that complementarity is strongest when solar output declines sharply while wind availability offsets the shortfall. By contrast, no meaningful relationship between correlation and synergy is observed in the early morning or daytime periods.

Finally, Table 7.4 reports average profits for each country under both models. As expected, storage increases absolute profits in every case. The relative gains are very large in countries with near-zero baseline profits in the Naive model (e.g., Estonia, Finland, Ireland, Latvia, Slovenia,), but these numbers are inflated by the small denominators. By contrast, in larger systems with substantial renewable capacity (France, Greece, Italy, the United Kingdom), storage delivers more modest percentage gains (about 110–280%) yet translates into absolute increases of several hundred thousand euros. In other words, while small systems exhibit dramatic percentage improvements, the most significant economic benefits in absolute terms accrue to the larger countries.

Summarizing this sensitivity analysis, three patterns emerge. First, increasing battery capacity raises absolute profits but reduces the incremental value of cooperation, particularly in resource-scarce or strongly complementary systems where storage substitutes for natural balancing. Second, storage stabilizes outcomes by reducing the sensitivity of Synergy Ratios to reliability thresholds thereby providing more predictable cooperative benefits. Third, correlations matter, but only partially: while negative PV-wind correlation in the late afternoon tends to support higher synergy, other temporal and operational factors play an equally important role. Taken together, these findings highlight that the economic value of cooperation is not determined by complementarity alone, but also by its interaction with storage capacity and broader system design.

TABLE 7.4: Average profits (in thousands of US dollars) from the combined wind and PV system under the Naive and Storage-Enhanced models for two reliability levels, $\varepsilon \in \{0.01, 0.05\}$. The ratios in parentheses show the multiplicative increase in profit from the Naive to the Storage-Enhanced model. For details, see Section 7.4.3.

Country	**Naive**		**Storage-Enhanced**	
	$\varepsilon = 0.01$	$\varepsilon = 0.05$	$\varepsilon = 0.01$	$\varepsilon = 0.05$
AT	8.16	12.61	82.68 (10.13)	99.76 (7.91)
BE	30.61	53.07	284.19 (9.28)	316.05 (5.96)
BG	9.45	15.05	67.60 (7.15)	77.59 (5.16)
CH	2.72	5.33	44.86 (16.50)	48.66 (9.13)
CZ	4.09	9.34	109.32 (26.73)	121.19 (12.98)
DE	672.15	917.10	3512.92 (5.23)	3976.74 (4.34)

continued on next page...

Table 7.4 (continued)

Country	Naive		Storage-Enhanced	
	$\varepsilon = 0.01$	$\varepsilon = 0.05$	$\varepsilon = 0.01$	$\varepsilon = 0.05$
DK	23.81	33.15	170.56 (7.16)	198.04 (5.97)
EE	0.03	0.08	5.76 (192.00)	7.24 (90.50)
ES	550.36	624.77	1425.75 (2.59)	1549.28 (2.48)
FI	0.08	0.29	19.18 (239.75)	24.16 (83.31)
FR	313.82	381.75	895.10 (2.85)	978.09 (2.56)
GB	320.28	425.17	1231.19 (3.84)	1359.99 (3.20)
GR	121.57	136.47	274.29 (2.26)	291.17 (2.13)
HR	0.34	0.72	9.70 (28.53)	10.91 (15.15)
HU	0.57	0.90	8.66 (15.19)	9.93 (11.03)
IE	0.02	0.25	48.07 (2403.50)	72.65 (290.60)
IT	563.05	667.90	1533.34 (2.72)	1632.95 (2.44)
LT	0.38	0.83	10.09 (26.55)	11.55 (13.92)
LU	0.48	0.89	8.34 (17.38)	9.37 (10.53)
LV	0.01	0.03	1.24 (124.00)	1.38 (46.00)
NL	27.18	43.93	212.59 (7.82)	243.07 (5.53)
NO	1.03	6.33	24.39 (23.68)	38.55 (6.09)
PL	1.39	2.54	106.32 (76.49)	141.77 (55.81)
PT	44.06	49.35	183.59 (4.17)	228.16 (4.62)
RO	5.53	11.05	119.47 (21.60)	134.11 (12.14)
SE	1.53	2.55	68.93 (45.05)	85.21 (33.42)
SI	0.06	0.66	10.41 (173.50)	11.75 (17.80)
SK	1.98	4.06	27.85 (14.07)	29.99 (7.39)

7.5 Conclusion

Our work examines the economic value of cooperation between wind and solar producers in the European day-ahead electricity market across 28 countries, using chance-constrained optimization models applied to the EMHIRES generation dataset. Our analysis yields several key insights.

In the Naive setup without storage, cooperation almost uniformly outperforms separate operation. At a relaxed reliability level ($\varepsilon = 0.05$), all countries except Latvia, Norway, Slovenia, and Slovakia achieve at least a 20% profit gain from cooperation. These four countries have either a PV share below 0.5% or above 98%, which limits the potential for meaningful synergy. In Estonia, Finland, and Ireland, where PV share is less than one percent, Synergy Ratios under the stricter $\varepsilon = 0.01$ requirement are either undefined or highly variable. By contrast, countries such as Finland, Lithuania, and Poland achieve the largest Synergy Ratios — often above a factor of seven and occasionally reaching values near thirty — simply because their PV shares are very

small (below 10%), and, thus, provide strong complementarity with wind. These findings confirm that natural negative correlation between wind and solar generation can act as a powerful hedge against delivery risk, though the resulting profits remain volatile when reliability constraints are tight.

In the Storage-Enhanced model (where the battery size is either the installed capacity for each technology, or the sum of these in the joint operation case), 18 out of 28 countries show a clear cooperative benefit, with profit increases of at least 10% and up to 50%. Since storage guarantees strictly positive profits in individual systems, Synergy Ratios are always defined; further, confidence intervals are narrow: lower bounds range from 1.00 to 1.39, and upper bounds from 1.00 to 1.51. This illustrates how storage stabilizes profits and makes outcomes far less sensitive to the reliability threshold. At the same time, the battery storage substitutes for natural balancing in countries where one resource is negligible. Finland, Ireland, Norway, Slovakia, and Slovenia all record ratios close to one, showing that cooperation adds little once storage is available. In contrast, balanced systems such as Belgium, Lithuania, the Netherlands, and Romania sustain Synergy Ratios between 1.3 and 1.5, confirming that cooperation continues to pay off in regions with moderate negative correlations and sufficient PV scale.

The sensitivity analysis with smaller and larger storage capacities reinforces these patterns. When the battery capacity is reduced by 50%, 21 of 28 countries achieve at least a 10% profit gain, with some reaching up to 70%. With a 100% increase, 16 countries show such gains. Here, we see that when the battery size is smaller, the Synergy Ratio is larger but wider in its confidence interval, while with a larger battery size, cooperation profit is smaller but more robust for this metric. The highest values in this setting occur in Belgium, Bulgaria, Lithuania, Romania, and the Netherlands (nearly all are all clustered in the Atlantic and Mediterranean regions), with PV proportions between 9.1% and 56.1%. By contrast, Finland, Ireland, Norway, Slovakia, and Slovenia have Synergy Ratios just above one, reflecting the lack of meaningful contribution from either of the two resources. Together, these results show that storage raises overall profits but redistributes the value of complementarity: it amplifies cooperation in balanced portfolios, while crowding it out in resource-scarce systems.

Overall, our findings suggest a clear policy direction for renewable integration. In regions where wind and solar resources are both significant and moderately negatively correlated—especially in the late afternoon—cooperation among independent producers can deliver stable and economically meaningful gains, particularly when supported by appropriately sized batteries (not larger than installed capacity). In contrast, where one resource is minimal, policymakers should prioritize expanding renewable capacity before expecting cooperation to yield benefits. More broadly, our results highlight that promoting coordinated bidding of complementary renewables may be a cost-effective alternative to large-scale storage investments.

Finally, beyond our empirical insights, our work demonstrates the versatility of chance-constrained optimization as a framework for evaluating renewable cooperation under uncertainty. By varying the reliability threshold ε, this approach quantifies the trade-off between risk and profit, offering a practical tool for both market design and long-term planning of renewable portfolios in Europe.

Chapter 8

Conclusions

8.1 Research Overview

This thesis addresses integer programming (IP) in two fundamental aspects of mathematical optimization: the structure of problems and problems under uncertainty. In the preceding chapters, we have demonstrated ways of solving algebraic structures arising from number theory, clustering models where nonconvex unit-norm constraints are applied, and approximation models for chance-constrained programming in risk-seeking behavior models. The research presented in this thesis introduces novel branch-and-bound frameworks tailored to algebraic structure constraints (prime programs) and nonconvex geometry constraints (iterative polyhedron approximation) for a deterministic setting. In the stochastic setting, we proposed models opposing traditional risk-averse paradigms by formulating risk-seeking models within a chance-constrained modeling framework together with approximation methods. Not only are theoretical solving and modeling techniques studied in this thesis, but applications of quadratic optimization models are also applied to simultaneously balance fairness and accessibility in public infrastructure planning. Additionally, the cooperation between two different renewable energy owners is examined under chance-constrained models. These contributions deal with data-driven decision-making that is explainable and transparent for public policy implications.

This final chapter synthesizes these contributions, summarizing the specific methodological advances and empirical findings of each core chapter. It further discusses the broader implications of this work for the optimization community and outlines a roadmap for future research.

8.2 Summary of Research Findings

The structure of the thesis contains two primary themes within the integer programming framework: Exact Deterministic Methods (Chapters 3–5) and Chance-constrained Programming (Chapters 6 and 7). The key findings from each chapter are detailed below.

8.2.1 Deterministic Optimization under Structure Models

8.2.1.1 Prime Programming: Bridging Number Theory and Optimization

In Chapter 3, we formalized the *Prime Programming (PP)* problem, a subclass of IP where decision variables are restricted to the set of prime numbers $\mathbb{P}$. The formulation represents an intersection between number theory and mathematical programming; however, traditional optimization solvers cannot directly address this model formulation. We developed a specialized B&B algorithm that replaces standard integer branching with domain-specific rules. Within this special solution structure, we introduce the "Modulo" branching strategy, which partitions the search space based on the arithmetic progressions of primes, and variable-fixing strategies as heuristic methods. These heuristics allowed us to solve larger instances of the Pomerance conjecture and reduce the size of the B&B tree by an order of magnitude compared to standard prime programming branching. Furthermore, we extend the concept of the inference dual to PPs, which enables sensitivity analysis for prime-constrained problems, allowing us to bound the change in the optimal objective value under perturbations without re-solving the entire tree.

8.2.1.2 Iterative Polyhedron Approximation for Nonconvex Geometry

Chapter 4 addressed machine learning models, specifically the *k-Hyperplane Clustering (k-HC)* problem, whose model structure contains nonconvex unit ℓ_2-norm constraints. Standard convex relaxations fail for this problem because the convex hull of the unit sphere is the unit ball, which often renders the relaxation too loose to be useful. We propose the *Iterative Polyhedron Approximation with Spatial Branch-and-Bound (IPA-SBB)* framework to avoid zero-objective solutions. The core idea of the method is to replace the ball geometry constraint with a polyhedron, then iteratively improve the polyhedron to be closer to the geometry of the ball. Since the method requires branching on each facet of the polyhedron, and the constraints in each branch are derived from the number of vertices, we first demonstrate that the n-simplex is a perfect initial polyhedron as it has $n+1$ vertices and $n+1$ facets. Next, we derive approximation factors based on the simplex gauge, proving that the optimal value of

the relaxed problem is bounded within a factor of $1/n^2$ of the true optimal value. Thirdly, we introduce the concepts of "down-extension" and "up-extension" of facets to maintain valid linear relaxations throughout the B&B tree and a projection-based method to iteratively improve the polyhedron, which completes the algorithm. In terms of computation, we apply linear algebra methods to deal with numerical instabilities along with branching settings to improve the algorithm. On both synthetic datasets (*LowDim* and *HighDim*), IPA-SBB outperformed the default Xpress global solver, achieving speedups and fewer nodes explored by up to two orders of magnitude. This demonstrates the impact of the IPA-SBB algorithm in solving the hyperplane clustering problem to global optimality. The method is also applicable to linear classification problems, especially in the case of a low number of features (low dimensions).

8.2.1.3 Optimizing Public Infrastructure in Hampshire

In Chapter 5, we applied structured mixed-integer quadratic programming (MIQP) to address challenges arising from the financial management of the Household Waste Recycling Centre (HWRC) network in Hampshire. We utilized an existing model from Schmitt and Singh (2024) which balances accessibility and fairness in a selective facility location-allocation model; an application involving the closure of recycling centers in Bavaria, Germany, is examined in Schmidt and Singh (2024). We apply this model to the context of recycling centers in Hampshire, where Hampshire has fewer recycling centers but a higher population density compared to the Germany case. With the available data, we compared the council's closure plan with our model's optimal results. We demonstrate that our model does not increase the travel distance significantly and ensures the utilization of facilities is within their capacities, preventing unrealistic overcrowding at remaining facilities. Furthermore, the model's optimal solution for each number of closures is nested; the set of facilities closed in the optimal solution for k closures is a strict subset of the optimal solution for $k+1$ closures. This monotonicity provides robust policy guidance, allowing the council to implement closures sequentially without fear of future regret. The analysis shows that the Hampshire network is resilient up to five closures, with minimal impact on accessibility. However, beyond 12 closures, the system becomes structurally fragile, disproportionately burdening two groups. This quantitative evidence supports a moderate rationalization strategy over aggressive cost-cutting measures.

8.2.2 Chance-constrained Programming

8.2.2.1 Risk-Seeking Formulations and Union Bounds

Chapter 6 shifts the focus to chance-constrained programming, where the constraint usually enforces satisfying certain conditions with high probability while avoiding overly conservative decisions which might be impossible in some settings. We formulated a concept of *risk-seeking* modeling that reflects human decision-making, extending to a higher level of violation tolerance. Specifically, we consider an objective function where the risk-seeker looks for scenarios where returns are higher than expected results by incorporating an upside potential term into the risk-seeking model. Therefore, we introduce a model formulation within a two-stage probabilistic constraint framework in the context of a resource allocation setting. To solve these computationally difficult problems, we developed approximations based on the probability of the union of events. We consider the tightest bounds, which are in the form of the sum of individual probability terms (S_1) and the sum of joint probability terms for two, three, and four events (S_2, S_3, S_4). These bounds are obtained by solving a linear program as in Prékopa (1988); replacing the probabilistic term with these bounds results in bilevel linear models. We reformulate these into single-level formulations and discretize the uncertainty into a set of possible scenarios. Then, we apply an integer programming solver to solve the approximation models to obtain the objective bounds.

8.2.2.2 Economic Value of Wind-Solar Complementarity

Finally, Chapter 7 applies chance-constrained programming to the European day-ahead bidding energy market, quantifying the economic value of coordinating wind and solar generation. Using historical data from 28 European countries, we analyzed the correlation between wind and solar outputs. We defined a "Synergy Ratio" to measure the profit increase gained by operating wind and solar assets as a combined hybrid system rather than independently. We generated discrete scenarios using ARMA forecasting techniques, then solved optimization models in two settings: without storage and with storage. The results show that complementarity acts as a natural substitute for battery storage, resulting in high synergy ratios. The advantages of cooperation are reduced in the model with batteries; however, this provides higher profits and a more predictable level of cooperation benefits. The study provides policy recommendations to balance the levels of wind and solar generation as a key step in obtaining such benefits.

8.3 Synthesis of Contributions

This thesis offers advances to algorithm design, theoretical approximation, and practical application. This research extends the B&B algorithm by creating specialized solution approaches for integer programs, particularly via a branching rule for prime numbers and the IPA-SBB framework for unit-norm constraints. These algorithmic advancements tackle the intrinsic intractability of prime-number structures—in which conventional optimization approaches frequently fail and the non-convex nature of norm constraints, for which standard relaxation methods are typically inefficient. By leveraging these mathematical characteristics, the presented frameworks offer effective solutions and global optimality guarantees for these intractable problems.

This study broadens the conventional settings for chance-constrained models within stochastic optimization. While such models are often limited to risk-averse or risk-neutral settings, this thesis introduces risk-seeking models, which give a rigorous mathematical framework for capturing “upside potential” and “lucky” or “greedy” scenarios. An alternative methodology, derived from the probabilistic bounds of union sets, was developed to solve these models; this approach establishes both valid lower and upper bounds.

The practical applications are illustrated via case studies on recycling infrastructure and the integration of renewable energy. This research employs optimization models to assess the closure of recycling centers in Hampshire and the integration of wind and solar energy throughout Europe. The identification of ”fragility points” in the Hampshire network and “synergy ratios” in the European grid provides policymakers with actionable intelligence, demonstrating that advanced optimization could serve as an effective tool for managing complicated public policies.

8.4 Limitations and Future Research Directions

Despite these advances, several limitations remain that define the boundaries of the current work:

- **Scalability in Prime Programming:** While the variable-fixing heuristics allow for the solution of larger instances, the Prime Programming algorithm in Chapter 3 still faces exponential complexity. Future work aims to integrate cutting plane methods for prime programs to speed up the search within the B&B algorithm.
- **Dimensionality in IPA-SBB:** In Chapter 4, the IPA-SBB algorithm depends on the approximation of the unit ball using polyhedra. As the dimension n grows,

the number of nodes in the B&B tree expands exponentially. The method is efficient for low-to-medium dimensional clustering but may incur significant computational expenses for high-dimensional data without further dimensionality reduction. Therefore, using dimensionality reduction methods to find representative points, as suggested by the coreset study, is important for achieving the global optimality of the hyperplane clustering problem in more points and dimensions.

- **Model Assumptions in Public Transport:** The willingness to travel mentioned in Chapter 5 relies on historical UK commuting data gathered for general reasons, which may not correctly represent the specific willingness to travel to a specified recycling center. Moreover, the financial benefits from facility closures are derived from economic model projections. In the future, developing a generic, user-friendly repository would enable the application of these models to other regions of the UK or foreign contexts facing similar facility consolidation challenges. Additionally, a stochastic framework may be employed to enhance the model formulation, beyond average waste quantities or utilization rates to include more detailed information if it becomes available.

- **Static Assumptions in Energy Models:** The wind-solar analysis in Chapter 7 is based on historical data and day-ahead market frameworks. It fails to comprehensively consider real-time market dynamics, intraday trading, or physical transmission limitations (e.g., AC power flow), which may affect the actual value of complementarity.

Appendix A

Appendix for Prime Programs

A.1 Formulations for Three Toy-problems

The following three problems, which can be formulated as PPs, are inspired by questions found on the websites cited below. This section provides prime programming formulations of such problems.

Definition A.1. A twin prime is a prime number that is either two more or two less than another prime number.

Problem 8. Find all consecutive twin prime pairs $(x_1, x_1 + 2)$ and $(x_2, x_2 + 2)$ such that x_2 is smaller than the sum of the first pair.

Solution. The corresponding optimization model is:

$$\min_x 0 \text{ s.t.} \{x_2 \geq x_1 + 4, x_2 \leq 2x_1 + 1, x_3 = x_1 + 2, x_4 = x_2 + 2 \text{ with } x_1, x_2, x_3, x_4 \in \mathbb{P}\}. \tag{A.1}$$

To ensure the smallest solution pair, a variation of the problem is the following model:

$$\min_x x_1 \text{ s.t.} \{x_2 \geq x_1 + 4, x_2 \leq 2x_1 + 1 \text{ with } x_1, x_1 + 2, x_2, x_2 + 2 \in \mathbb{P}\}. \tag{A.2}$$

The solution to this model provides one feasible pair, (5, 7) and (11, 13); i.e., $(x_1^*, x_2^*, x_3^*, x_4^*) = (5, 11, 7, 13)$. For the next solution, we add a cutting plane $x_1 \geq x_1^* + 1$ and continue this process. Repeating this process provides us several such solutions but not necessarily *all* the solutions. In the first five iterations of the naive branch-and-bound procedure we describe in Section 3.3.1 of the main text, we obtain solutions $(5, 11, 7, 13), (11, 17, 13, 19), (17, 29, 19, 31), (29, 41, 31, 43), (41, 59, 43, 61)$. However, the solutions obtained from this model are not necessarily consecutive twin prime pairs. We can verify the consecutive nature of the obtained twin primes, $(x_1^*, x_1^* + 2)$ and $(x_2^*, x_2^* + 2)$, if there exists $(x_0, x_0 + 2) \in \mathbb{P}$ for some $x_0 \leq x_2^*$. □

For a similar problem to Problem 8,
see https://math.stackexchange.com/q/4014554.

Definition A.2. A super prime square is a prime number, x, such that $(a_1 + a_2 + a_3 + a_4)/6$ is prime for some $a_1, \ldots, a_4 \in \mathbb{Z}^+$ and $x + a_i$ is also a prime number for all $i = 1, \ldots, 4$.

Problem 9. Find the largest super prime square.

Solution. The corresponding optimization model is:

$$\max_{x,a} x_0 \text{ s.t.}\{6x_0 = a_1 + a_2 + a_3 + a_4, x_1 = x_0 + a_1, x_2 = x_0 + a_2, x_3 = x_0 + a_3, x_4 = x_0 + a_4, \\ x_2 \geq x_1 + 1, x_3 \geq x_2 + 1, x_4 \geq x_3 + 1 \text{ with } x_0, x_1, \ldots, x_4 \in \mathbb{P} \text{ and } a_1, \ldots, a_4 \in \mathbb{Z}^+\}. \tag{A.3}$$

To ensure the model is bounded, we need an upper bound for the variables. When the upper bound is set to 10^5, the solution to this model (using the branch-and-bound method we present in Section 3.3.1) provides one feasible tuple, $(x_0^*, x_1^*, x_2^*, x_3^*, x_4^*, a_1^*, a_2^*, a_3^*, a_4^*) = (3947, 9973, 9967, 9781, 9749, 6026, 6020, 5834, 5802)$ with the solution to the problem as $x_0^* = 3947$. Similarly, we obtain $x_0^* = 39727$ and $x_0^* = 399853$ when the upper bounds are set to 10^6 and 10^7, respectively. It is possible that there are infinitely many solutions. If that is the case, there is no largest super prime square. □

For a similar problem to Problem 9,
see https://math.stackexchange.com/q/2515549.

Definition A.3. A sexy prime pair is a pair of prime numbers differing by 6.

Problem 10. Find the largest sexy prime pair $(x, x + 6)$, such that their sum is divisible by 10.

Solution. The corresponding optimization model is:

$$\max_{x,z} x_1 \text{ s.t.}\{2x_1 + 6 = 10z, x_2 = x_1 + 6 \text{ with } x_1, x_2 \in \mathbb{P} \text{ and } z \in \mathbb{Z}^+\}. \tag{A.4}$$

Again, an upper bound for variables is needed to ensure the model is bounded. To simplify the model, we observe that if $2x + 6 \equiv 0 \mod 10$, then $x \equiv 2, 7 \mod 10$. Since x and $x + 6$ are primes, x must be of the form $x = 10a + 7$ for some $a \in \mathbb{Z}^+$. The simplified model is: $\max_{x,a} x_1$ s.t.$\{x_1 = 10a + 7, x_2 = x_1 + 6$ with $x_1, x_2 \in \mathbb{P}$ and $a \in \mathbb{Z}^+\}$. When the upper bound is set to 10^5, the solution to this model provides one feasible tuple, $(x_1^*, x_2^*, a^*) = (99817, 99823, 9981)$ with the sexy prime pair $(99817, 99823)$. Similarly, we obtain a sexy prime pair $(998737, 998743)$ and $(9999937, 9999943)$ when the upper bound is set to 10^6 and 10^7, respectively. It is

possible that there are infinitely many solutions. If that is the case, there is no largest such sexy prime pair. □

For a similar problem to Problem 10,
see https://math.stackexchange.com/q/3230707.

A.2 Proof of Sensitivity Analysis for PP

In this section we provide a proof of Theorem 3.1 of the main text. The proof mirrors the proof given in Dawande and Hooker (2000) with the key difference that the original work is for mixed-integer programs while ours considers prime programs. The central idea is to derive sufficient conditions that ensure $z^* - \Delta \leq z^*_\delta$. These conditions are informed by determining a set of infeasible (or, violated) inequalities, that the authors term as "surrogate inequalities", at each of the leaf nodes. To this end, we first drop both the prime number as well as the integer restrictions on the x variables of model (3.14). Then, the following relaxed model is solved at each node of the branch-and-bound tree.

$$\min_x \quad cx \tag{A.5a}$$
$$\text{s.t.} \quad Ax \geq b, \tag{A.5b}$$
$$Cx \geq d, \tag{A.5c}$$
$$0 \leq x \leq M. \tag{A.5d}$$

Here, $Cx \geq d$ denotes the branching cuts at each node where a variable x_j has a non-prime value, v. The left and right child nodes have branching cuts of the form $x_j \leq \underline{p}_v$ and $x_j \geq \bar{p}_v$, respectively. In constraint (A.5d), the bound of x variable is relaxed to between 0 and M; where we can write the model with an additional constraint $x_i \geq 2$ for all $i = 1, \ldots, n$ combined within the constraints (A.5c). Then, model (A.5) has the same structure as the relaxed model (17) employed in Dawande and Hooker (2000); hence, except for the branching rules of prime programs and mixed-integer programs, the surrogate inequalities at leaf nodes follow directly. Let (λ, μ, ν) be the non-negative dual multipliers of constraints (A.5b)-(A.5d), respectively. The following surrogate inequalities for model (A.5) directly result from Dawande and Hooker (2000).

(i) For infeasible leaf nodes, the surrogate inequality is $(\lambda A)x \geq \lambda b$.

(ii) For leaf nodes pruned by optimality, the surrogate inequality is $(\lambda A - c)x \geq \lambda b - \hat{z} + \Delta + \epsilon$, where $\hat{z}$ is the node's objective function value.

(iii) For leaf nodes pruned by bound, the surrogate inequality is $(\lambda A - c)x \geq \lambda b - \bar{z} + \Delta + \epsilon$, where $\bar{z}$ is the incumbent value of the objective function so far.

At node p, consider

$$x_j \in [\underline{v}_j, \bar{v}_j] \cap \mathbb{P},\ j = 1, \ldots, n, \tag{A.6}$$

where $\underline{v}, \bar{v}$ are prime numbers between 2 and M. We now define the following values as in Section 3.3.3,

$$z_p = \begin{cases} \epsilon, \\ \bar{z}_p - \Delta, \\ z_p^{UB} - \Delta, \end{cases} \quad q^p = \begin{cases} \lambda^p A, \\ \lambda^p A - c, \\ \lambda^p A - c, \end{cases} \quad q_\delta^p = \begin{cases} \lambda^p A_\delta, & \text{: if the node is pruned by infeasibility,} \\ \lambda^p A_\delta - c_\delta, & \text{: if the node is pruned by optimality,} \\ \lambda^p A_\delta - c_\delta, & \text{: if the node is pruned by bound.} \end{cases}$$

Then, the surrogate inequality at node p is $q^p x \geq \lambda^p b - z_p + \epsilon$. The corresponding perturbed surrogate inequality is then of the form $(q^p + q_\delta^p)x \geq \lambda^p(b + b_\delta) - z_p + \epsilon$. We now derive sufficient conditions such that the perturbed inequality remains infeasible for each leaf node.

Lemma A.4. *Let $x_j \in [\underline{v}_j, \bar{v}_j] \cap \mathbb{P}$, $j = 1, \ldots, n$, and consider a correspondingly infeasible surrogate inequality $q^p x \geq \lambda^p b - z_p + \epsilon$. Then, the perturbed inequality $(q^p + q_\delta^p)x \geq \lambda^p(b + b_\delta) - z_p + \epsilon$ remains infeasible if and only if there exist $\bar{q}_1^p, \ldots, \bar{q}_n^p$ such that*

$$\sum_{j=1}^{n} \left((q_j^p + q_{\delta j}^p)\underline{v}_j + \bar{q}_j^p(\bar{v}_j - \underline{v}_j) \right) < \lambda^p(b + b_\delta) - z_p + \epsilon, \tag{A.7a}$$

$$\bar{q}_j^p \geq q_j^p + q_{\delta j}^p,\ \bar{q}_j \geq 0,\ j = 1, \ldots, n. \tag{A.7b}$$

Proof. This result is based on Lemma 2 of Dawande and Hooker (2000). The largest possible value of $(q^p + q_\delta^p)x$ is as given by the following quantity:

$$\sum_{j: q_j^p + q_{\delta j}^p < 0} (q_j^p + q_{\delta j}^p)\underline{v}_j + \sum_{j: q_j^p + q_{\delta j}^p > 0} (q_j^p + q_{\delta j}^p)\bar{v}_j. \tag{A.8}$$

$\Longrightarrow$ Assume that the inequality $(q^p + q_\delta^p)x \geq \lambda^p(b + b_\delta) - z_p + \epsilon$ is infeasible. Let $\bar{q}_j^p = \max\{q_j^p + q_{\delta j}^p, 0\}$ for $j = 1, \ldots, n$ which satisfies equation (A.7b). We have:

$$\begin{aligned}\sum_{j=1}^{n}\left((q_j^p+q_{\delta j}^p)\underline{v}_j+\bar{q}_j^p(\bar{v}_j-\underline{v}_j)\right) &= \sum_{j:q_j^p+q_{\delta j}^p<0}(q_j^p+q_{\delta j}^p)\underline{v}_j+\sum_{j:q_j^p+q_{\delta j}^p>0}(q_j^p+q_{\delta j}^p)\underline{v}_j\\ &\quad+\sum_{j:q_j^p+q_{\delta j}^p>0}(q_j^p+q_{\delta j}^p)(\bar{v}_j-\underline{v}_j)\\ &= \sum_{j:q_j^p+q_{\delta j}^p<0}(q_j^p+q_{\delta j}^p)\underline{v}_j+\sum_{j:q_j^p+q_{\delta j}^p>0}(q_j^p+q_{\delta j}^p)\bar{v}_j\\ &< \lambda^p(b+b_\delta)-z_p+\epsilon,\end{aligned}$$

where the last inequality holds since the inequality assumed is infeasible, and the preceding expression is its largest term. Hence, equation (A.7) is satisfied.

$\Longleftarrow$ Now, assume that there exist $\bar{q}_1^p,\ldots,\bar{q}_n^p$ satisfying (A.7). Then,

$$\begin{aligned}\lambda^p(b+b_\delta)-z_p+\epsilon &> \sum_{j=1}^{n}((q_j^p+q_{\delta j}^p)\underline{v}_j+\bar{q}_j^p(\bar{v}_j-\underline{v}_j))\\ &\geq \sum_{j=1}^{n}(q_j^p+q_{\delta j}^p)\underline{v}_j+\sum_{j:q_j^p+q_{\delta j}^p>0}(q_j^p+q_{\delta j}^p)(\bar{v}_j-\underline{v}_j)\\ &= \sum_{j:q_j^p+q_{\delta j}^p<0}(q_j^p+q_{\delta j}^p)\underline{v}_j+\sum_{j:q_j^p+q_{\delta j}^p>0}(q_j^p+q_{\delta j}^p)\bar{v}_j\\ &> (q^p+q_\delta^p)x.\end{aligned}$$

Hence, the inequality $(q^p+q_\delta^p)x\geq\lambda^p(b+b_\delta)-z_p+\epsilon$ is infeasible. □

Lemma A.4 completes the proof of Theorem 3.1.

A.3 Example of Sensitivity Analysis for Prime Programming

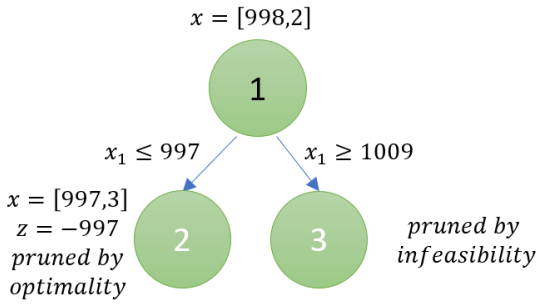
In this section, we present an additional example of sensitivity analysis for the prime program, specifically in relation to the Goldbach conjecture.

Problem 11. Find the largest prime number, x_1, satisfying the condition $2k+2=x_1+x_2$ for a given $k\in\mathbb{Z}^+$ where $x_2\in\mathbb{P}$.

The following PP formulates this problem with $k=499$.

$$\begin{aligned} z^* = \min \quad & -x_1 && \text{(A.9a)}\\ \text{s.t} \quad & x_1+x_2\geq 1000, && \text{(A.9b)}\\ & -x_1-x_2\geq -1000, && \text{(A.9c)}\\ & x_1,x_2\in[2,M]\cap\mathbb{P}. && \text{(A.9d)}\end{aligned}$$

FIGURE A.1: Enumeration tree for the prime branch-and-bound procedure applied to model (A.9). The two nodes marked in white are leaf nodes that are pruned below for reasons indicated, while the other node is further branched on. For details, see Section 3.3.1.



We consider a perturbation of the right hand side with $b_\delta = [k_\delta, -k_\delta]$ for even k_δ. In this case, the enumeration tree of the branch-and-bound procedure described in Section 3.3.1 of the main text contains three nodes as in Fig. A.1. For additional details, see Section 3.3.3 of the main text.

(i) Consider node 2. We have $\underline{x}^2 = [2,2], \bar{x}^2 = [997, M], z^2 = -997 - \Delta, \lambda^2 = [0,0]$. Then, $q^2 = [1,0], q^2_\delta = [0,0]$. In order to satisfy the sufficient conditions of Theorem 3.1 at this node, we require a solution $\bar{q}^2_1, \bar{q}^2_2$ such that $2 + 995\bar{q}^2_1 + (M-2)\bar{q}^2_2 \leq 997 + \Delta$ and $\bar{q}^2_1 \geq 1, \bar{q}^2_2 \geq 0$. By setting $\bar{q}^2_1, \bar{q}^2_2 = [1,0]$ we obtain $\Delta \geq 0$.

(ii) Consider node 3. We have $\underline{x}^3 = [1009,2], \bar{x}^3 = [M, M], z^3 = \varepsilon, \lambda^3 = [0,1]$. Then, $q^3 = [-1,-1], q^3_\delta = [0,0]$. In order to satisfy the sufficient conditions of Theorem 3.1 at this node, we require a solution $\bar{q}^3_1, \bar{q}^3_2$ such that $-1011 + (M-1009)\bar{q}^3_1 + (M-2)\bar{q}^3_2 \leq -1000 - k_\delta - \varepsilon$ and $\bar{q}^3_1, \bar{q}^3_2 \geq 0$. By setting $\bar{q}^3_1, \bar{q}^3_2 = [0,0]$ we obtain $k_\delta \leq 11 - \varepsilon$.

The optimal solution of model (A.9) is $x^* = [997, 3]$. Thus, we conclude that $z^*_\delta \geq -997 - \Delta$ holds for all $\Delta \geq 0$ if $k_\delta = \{2, 4, 6, 8, 10\}$; i.e., the optimal value of x^*_1 for the perturbation of model (A.9) does not exceed 997.

Unlike Problem 7 of the main text, enumerating all the possibilities requires checking all prime numbers below 997 (there are 167 such numbers). However, consider $k_\delta = 2$ and the corresponding optimization model: $\max_x x_1$ s.t.$\{x_1 + x_2 = 1002$ with $x_1, x_2 \in \mathbb{P}\}$. From our results of the sensitivity analysis above, we begin directly with $x_1 = 997$. Then, $[x_1, x_2] = [997, 5]$ is feasible, and, hence optimal. Similarly, for $k_\delta = 4, 8, 10$ we obtain the optimal solution in the first computation itself. For $k_\delta = 6$ the corresponding optimization model, $\max_x x_1$ s.t.$\{x_1 + x_2 = 1006$ with $x_1, x_2 \in \mathbb{P}\}$, is infeasible for $x_1 = 997$. In the second computation, we test $x_1 = 991$; i.e., the prime

immediately smaller than 997. The problem is still infeasible. However, in the third iteration, we have $x_1 = 983$ and obtain an solution $[x_1, x_2] = [983, 23]$. Thus, we require at most three additional computations and there is no need to solve another optimization model. In contrast, the branch-and-bound tree for this instance of model (A.9) has 11 nodes.

A.4 Numerical Results for Linear Equations in Primes

Example A.1. *An illustration of the benefit of increasing the allowed time limit for solving model* (3.28)*.*

Consider an instance of model (3.28) solved with Routine 2 with $n = 8$, $M = 3000$ and a time limit of 300 seconds. With both the `Naive` strategy and the `Modulo` strategy with arg $= 11$, the instance neither obtains a feasible solution nor proves infeasibility within the considered time limit. However, increasing the time limit to 7200 and 600 seconds, respectively, provides feasible solutions. Feasible solutions are obtained relatively quickly for the `Modulo` strategy with arg $= 1, 5, 7$.

□

Example A.2. *An illustration of the computational benefit from using the three enhanced variable fixing strategies of Routine 1 over the* `Naive` *fixing strategy.*

Consider an instance of model (3.28) with $n = 8$ and $M = 3000$ solved using the `Naive branching strategy` . The `Naive` fixing strategy takes 821 seconds to solve this instance (see Table 3.1). Now, consider the known solution, $\bar{x}$, for $n = 7$ with $\bar{x} = [5, 17, 41, 101, 257, 461, 521]$ as an input for Routine 2.

(i) In the `SelectAll` strategy, the first seven components are fixed; i.e., $x = \bar{x}$. Here, the instance's infeasibility is determined practically instantly. However, increasing M to 10000 provides a feasible solution in practically no extra time: $[5, 17, 41, 101, 257, 461, 521, 4157]$. This reduces the computational time from 821 seconds to only 82 seconds; the latter is the computational time to obtain the $n = 7$ solution (see Table 3.1).

(ii) In the `ExcludeOne` strategy with $l = 6$, we have six elements fixed. Then, we have $x = [5, 17, 41, 101, 257, 521, \phi, \phi]$. A feasible solution is determined in no time: $[5, 17, 41, 101, 257, 521, 761, 881]$. This again reduces the computational time to 82 seconds, which is the time required for the $n = 7$ solution.

(iii) In the `ExcludeTwo` strategy with $(l_1, l_2) = (6, 7)$, we have 5 elements fixed. Then, we have $\bar{x} = [5, 17, 41, 101, 257]$. A feasible solution is obtained in only an additional second: $x = [5, 17, 41, 101, 257, 521, 761, 881]$. The computational

time is again reduced from 821 seconds of a `Naive` strategy (see Table 3.1) to 83 seconds; this includes 82 seconds to obtain the $n = 7$ solution.

□

Example A.3. *An illustration of the computational benefit from using the* `Modulo` *branching strategy of Routine 2 over the* `Naive` *branching strategy.*

Consider an instance of model (3.28) with $n = 8$ and $M = 3000$ solved using the `Naive fixing strategy` . The `Naive` branching strategy takes 821 seconds to solve this instance (see Table 3.1).

(i) With the `Modulo` strategy and arg $= 1$, the instance is proven infeasible in 88 seconds. However, increasing M to 10000 provides a different feasible solution, $[13, 73, 181, 241, 373, 1693, 1801, 1861]$, in only two seconds.

(ii) With the `Modulo` strategy and arg $= 5$, the same feasible solution as the `Naive` strategy is found in 37 seconds.

(iii) With the `Modulo` strategy and arg $= 7$, the instance is proven infeasible in 246 seconds. However, increasing M to 10000 provides another feasible solution, $[7, 19, 67, 139, 607, 859, 907, 1699]$, in only two seconds.

(iv) With the `Modulo` strategy and arg $= 11$, the instance is proven infeasible in 195 seconds. However, increasing M to 10000 provides another feasible solution, $[11, 23, 71, 191, 1103, 1871, 2543, 7043]$, in 144 seconds.

□

The following tables present the numerical results obtained from Algorithm 3 using different fixing and branching strategies, each with an appropriate time limit for different values of n. All models yield solutions practically instantaneously for $n = 5$. The tables depict the computational time for each strategy for $n = 6$ to 10; the row with the lowest computation time for each branching strategy is highlighted in bold. For instances that cannot obtain a solution or report infeasibility within a given time limit, we report as "Intractable". The results of the algorithm with the inputs $n = 11$; $M_i = 1000$ for $i = 1, \ldots, n$; $T = 6000$; $t = 3000$; $\alpha = 0.2$ are not shown here since all except two cases are intractable. These two cases are the `SelectAll` fixing strategy with the `Modulo` branching strategy arguments of 5 and 7 respectively; they yield solutions $[5, 17, 881, 1181, 1637, 14957, 24197, 131297, 184721, 6353021, 69739337]$ and $[7, 19, 907, 1699, 1747, 3967, 4759, 16519, 23167, 4891507, 13542967]$ with a computation time of 3772 and 664 seconds, respectively. Our algorithm is unable to find a solution or report infeasibility in 6000 seconds for $n = 12$. For details, see Section 3.5.3 of the main text.

TABLE A.1: Computational results for solving model (3.28) using the fixing strategies in Routine 1 and branching strategies in Routine 2. Inputs in Algorithm 3 are $n = 5$; $M_i = 1000$ for $i = 1, \ldots, n$; $T = 600$; $t = 300$; $\alpha = 0.2$. All instances obtain a solution immediately.

Strategy				**Solution**
Branching		**Fixing**		
name	**arg**	**name**	**arg**	
`Naive`	ϕ	`Naive`	ϕ	[3, 7, 19, 139, 859]
		`SelectAll`	ϕ	[3, 7, 919, 2179, 4159]
		`ExcludeOne`	[1]	[67, 127, 607, 631, 1867]
			[2]	[3, 1123, 1483, 2383, 4219]
			[3]	[3, 7, 859, 1279, 4219]
			[4]	[3, 7, 919, 2179, 4159]
		`ExcludeTwo`	[1, 2]	[1579, 1987, 1999, 2287, 4639]
			[1, 3]	[43, 823, 883, 1051, 1423]
			[1, 4]	[67, 127, 151, 331, 547]
			[2, 3]	[3, 139, 223, 523, 619]
			[2, 4]	[3, 1123, 1279, 1303, 2203]
			[3, 4]	[3, 7, 19, 139, 859]
`Modulo`	1	`Naive`	ϕ	[13, 61, 73, 241, 853]
		`SelectAll`	ϕ	[13, 853, 1129, 1213, 2113]
		`ExcludeOne`	[1]	[1321, 1621, 1873, 3001, 4813]
			[2]	[13, 1129, 1213, 2113, 2389]
			[3]	[13, 853, 1213, 1753, 2113]
			[4]	[13, 853, 1129, 1213, 2389]
		`ExcludeTwo`	[1, 2]	[241, 373, 601, 853, 1381]
			[1, 3]	[1237, 1621, 1861, 3457, 4597]
			[1, 4]	[1321, 1621, 1777, 3001, 3457]
			[2, 3]	[13, 241, 373, 601, 853]
			[2, 4]	[13, 1129, 1213, 2113, 2389]
			[3, 4]	[13, 853, 1453, 1489, 1609]
`Modulo`	5	`Naive`	ϕ	[5, 17, 29, 89, 449]
		`SelectAll`	ϕ	[5, 17, 881, 1181, 1637]
		`ExcludeOne`	[1]	[1229, 1637, 2777, 2897, 2909]
			[2]	[5, 53, 89, 113, 929]
			[3]	[5, 17, 509, 617, 1109]
			[4]	[5, 17, 881, 1181, 1637]
		`ExcludeTwo`	[1, 2]	[89, 113, 389, 449, 1433]
			[1, 3]	[1361, 1637, 2417, 3257, 3461]
			[1, 4]	[1229, 1637, 2417, 2897, 2909]
			[2, 3]	[5, 89, 113, 269, 929]
			[2, 4]	[5, 53, 89, 929, 953]
			[3, 4]	[5, 17, 29, 89, 449]
`Modulo`	7	`Naive`	ϕ	[7, 19, 67, 127, 607]
		`SelectAll`	ϕ	[7, 19, 907, 1699, 1747]
		`ExcludeOne`	[1]	[1087, 2011, 2467, 3691, 4987]
			[2]	[7, 967, 1279, 3307, 3727]
			[3]	[7, 19, 907, 1699, 1747]
			[4]	[7, 19, 907, 1699, 1747]
		`ExcludeTwo`	[1, 2]	[1231, 1327, 1531, 1567, 1987]

Table A.1 – *Continued from previous page*

Strategy				Solution
Branching		Fixing		
name	**arg**	**name**	**arg**	
			[1, 3]	[1447, 1867, 1879, 3499, 3559]
			[1, 4]	[1087, 2011, 2251, 2311, 3691]
			[2, 3]	[7, 127, 151, 271, 547]
			[2, 4]	[7, 967, 1531, 2131, 3571]
			[3, 4]	[7, 19, 67, 859, 907]
`Modulo`	11	`Naive`	ϕ	[11, 23, 71, 191, 443]
		`SelectAll`	ϕ	[11, 23, 911, 1283, 2711]
		`ExcludeOne`	[1]	[1223, 1979, 2243, 2939, 3083]
			[2]	[11, 1031, 1187, 1931, 2207]
			[3]	[11, 23, 911, 1283, 2843]
			[4]	[11, 23, 911, 1283, 2843]
		`ExcludeTwo`	[1, 2]	[191, 827, 947, 2027, 2447]
			[1, 3]	[1223, 2243, 2423, 2999, 3083]
			[1, 4]	[1223, 1979, 2243, 2939, 3083]
			[2, 3]	[11, 911, 947, 1607, 1667]
			[2, 4]	[11, 1031, 1163, 1583, 2111]
			[3, 4]	[11, 23, 71, 191, 443]

TABLE A.2: Computational results for solving model (3.28) using the fixing strategies in Routine 1 and the branching strategies in Routine 2. Inputs in Algorithm 3 are $n = 6$; $M_i = 1000$ for $i = 1, \ldots, n$; $T = 3000$; $t = 600$; $\alpha = 0.2$.

Strategy				Solution	Time (s)
Branching		Fixing			
name	**arg**	**name**	**arg**		
`Naive`	ϕ	`Naive`	ϕ	[3, 11, 23, 71, 191, 443]	120
		`SelectAll`	ϕ	[3, 7, 919, 2179, 4159, 13399]	**1**
		`ExcludeOne`	[1]	[127, 607, 631, 1867, 8971, 10267]	21
			[2]	[3, 1483, 2383, 4219, 6823, 13723]	18
			[3]	[3, 7, 1279, 4219, 7159, 8179]	13
			[4]	[3, 7, 919, 4159, 13399, 37039]	255
			[5]	[3, 7, 919, 2179, 4159, 13399]	36
		`ExcludeTwo`	[1, 2]	[1999, 2287, 4639, 6199, 9007, 10459]	74
			[1, 3]	[823, 1051, 1423, 2083, 2143, 4951]	22
			[1, 4]	[127, 151, 547, 607, 1951, 2791]	22
			[1, 5]	[127, 607, 631, 3931, 6007, 8971]	443
			[2, 3]	[3, 523, 619, 859, 1723, 2719]	16
			[2, 4]	[3, 1279, 2203, 3163, 3463, 4423]	15
			[2, 5]	[3, 1483, 2383, 2551, 4051, 6163]	13
			[3, 4]	[3, 7, 859, 1279, 4219, 8179]	149
			[3, 5]	[3, 7, 1279, 4759, 5179, 8179]	228
			[4, 5]	[3, 7, 919, 2179, 4159, 13399]	716
`Modulo`	1	`Naive`	ϕ	[13, 61, 73, 241, 601, 853]	2
		`SelectAll`	ϕ	[13, 853, 1129, 1213, 2113, 2389]	**0**
		`ExcludeOne`	[1]	[1621, 1873, 3001, 4813, 5521, 21313]	8
			[2]	[13, 1213, 2113, 2389, 9013, 15493]	9
			[3]	[13, 853, 1753, 2113, 3121, 4021]	1
			[4]	[13, 853, 1129, 2389, 3049, 3313]	1
			[5]	[13, 853, 1129, 1213, 2113, 2389]	**0**
		`ExcludeTwo`	[1, 2]	[601, 853, 1381, 2341, 2593, 3121]	2
			[1, 3]	[1621, 3457, 4597, 5641, 5857, 7417]	7
			[1, 4]	[1621, 1777, 3457, 4597, 5641, 7417]	5
			[1, 5]	[1621, 1873, 3001, 9433, 10141, 11593]	39
			[2, 3]	[13, 601, 853, 1213, 3373, 4021]	5
			[2, 4]	[13, 1213, 2389, 2953, 3049, 3733]	1
			[2, 5]	[13, 1213, 2113, 2293, 2389, 7333]	10
			[3, 4]	[13, 853, 1609, 2053, 2113, 2389]	1
			[3, 5]	[13, 853, 1753, 2113, 3121, 4021]	2
			[4, 5]	[13, 853, 1129, 1213, 2113, 2389]	**0**
`Modulo`	5	`Naive`	ϕ	[5, 17, 41, 101, 257, 461]	**0**
		`SelectAll`	ϕ	[5, 17, 881, 1181, 1637, 14957]	1
		`ExcludeOne`	[1]	[1637, 2777, 2897, 2909, 3917, 6389]	2
			[2]	[5, 89, 113, 929, 1193, 1949]	**0**
			[3]	[5, 17, 617, 1109, 1277, 1709]	**0**
			[4]	[5, 17, 881, 1637, 2357, 4877]	1
			[5]	[5, 17, 881, 1181, 9281, 11717]	7
		`ExcludeTwo`	[1, 2]	[389, 449, 1433, 3413, 3833, 5009]	4
			[1, 3]	[1637, 3257, 3461, 4637, 5441, 10457]	12
			[1, 4]	[1637, 2417, 2909, 3917, 6257, 6869]	7

Table A.2 – *Continued from previous page*

Strategy				Solution	Time (s)
Branching		Fixing			
name	arg	name	arg		
			[1, 5]	[1637, 2777, 2897, 3821, 4001, 4157]	1
			[2, 3]	[5, 269, 929, 1277, 1949, 2897]	1
			[2, 4]	[5, 89, 953, 1109, 4133, 6653]	12
			[2, 5]	[5, 89, 113, 389, 449, 1433]	1
			[3, 4]	[5, 17, 449, 509, 1877, 4817]	6
			[3, 5]	[5, 17, 617, 1109, 1277, 1709]	1
			[4, 5]	[5, 17, 881, 1301, 2861, 3797]	4
Modulo	7	Naive	ϕ	[7, 19, 67, 127, 547, 607]	**0**
		SelectAll	ϕ	[7, 19, 907, 1699, 1747, 3967]	**0**
		ExcludeOne	[1]	[2011, 2467, 3691, 4987, 13567, 28771]	21
			[2]	[7, 1279, 3307, 3727, 4159, 5407]	1
			[3]	[7, 19, 1699, 1747, 3967, 4759]	1
			[4]	[7, 19, 907, 1747, 3967, 4759]	1
			[5]	[7, 19, 907, 1699, 1747, 3967]	1
		ExcludeTwo	[1, 2]	[1531, 1567, 1987, 2791, 3547, 3847]	2
			[1, 3]	[1867, 3499, 3559, 3727, 4759, 8179]	7
			[1, 4]	[2011, 2251, 3691, 4363, 5443, 7963]	6
			[1, 5]	[2011, 2467, 3691, 6211, 6247, 11887]	25
			[2, 3]	[7, 271, 547, 607, 967, 2791]	2
			[2, 4]	[7, 1531, 3571, 3967, 6991, 8191]	7
			[2, 5]	[7, 1279, 3307, 3727, 4159, 5407]	2
			[3, 4]	[7, 19, 907, 1699, 1747, 3967]	6
			[3, 5]	[7, 19, 1699, 2887, 3727, 3739]	3
			[4, 5]	[7, 19, 907, 1699, 1747, 3967]	6
Modulo	11	Naive	ϕ	[11, 23, 83, 251, 443, 683]	**0**
		SelectAll	ϕ	[11, 23, 911, 1283, 2711, 2843]	**0**
		ExcludeOne	[1]	[1979, 2243, 2939, 3083, 4919, 5099]	1
			[2]	[11, 1187, 1931, 2207, 5147, 5231]	1
			[3]	[11, 23, 1283, 2843, 3203, 12611]	4
			[4]	[11, 23, 911, 2843, 4691, 5903]	1
			[5]	[11, 23, 911, 1283, 2711, 2843]	**0**
		ExcludeTwo	[1, 2]	[947, 2027, 2447, 4211, 5867, 5987]	5
			[1, 3]	[2243, 2999, 3083, 5903, 6299, 7703]	6
			[1, 4]	[1979, 2243, 3083, 4679, 4919, 5099]	2
			[1, 5]	[1979, 2243, 2939, 3083, 4919, 5099]	2
			[2, 3]	[11, 1607, 1667, 2087, 2591, 3011]	1
			[2, 4]	[11, 1163, 2111, 2843, 5903, 7823]	10
			[2, 5]	[11, 1187, 1931, 2207, 5147, 5231]	3
			[3, 4]	[11, 23, 443, 491, 683, 911]	**0**
			[3, 5]	[11, 23, 1283, 2711, 2843, 3203]	2
			[4, 5]	[11, 23, 911, 1283, 2711, 2843]	1

TABLE A.3: Computational results for solving model (3.28) using the fixing strategies in Routine 1 and the branching strategies in Routine 2. Inputs in Algorithm 3 are $n = 7$; $M_i = 1000$ for $i = 1, \ldots, n$; $T = 3000$; $t = 600$; $\alpha = 0.2$.

Strategy				Solution	Time (s)
Branching		Fixing			
name	**arg**	**name**	**arg**		
`Naive`	ϕ	`Naive`	ϕ	[5, 17, 41, 101, 257, 461, 521]	243
		`SelectAll`	ϕ	[3, 7, 919, 2179, 4159, 13399, 37039]	**4**
		`ExcludeOne`	[1]	[607, 631, 1867, 8971, 10267, 10687, 24151]	42
			[2]	[3, 2383, 4219, 6823, 13723, 14083, 25639]	29
			[3]	[3, 7, 4219, 7159, 8179, 10039, 20599]	36
			[4]	[3, 7, 919, 13399, 37039, 44179, 191599]	2940
			[5]	Intractable	3000
			[6]	[3, 7, 919, 2179, 4159, 13399, 37039]	167
		`ExcludeTwo`	[1, 2]	[4639, 6199, 9007, 10459, 18127, 21379, 37987]	942
			[1, 3]	[1051, 2083, 2143, 4951, 13063, 33331, 39043]	2077
			[1, 4]	[151, 547, 1951, 2791, 6091, 6451, 8167]	60
			[1, 5]	[607, 631, 3931, 8971, 10687, 10771, 11047]	455
			[1, 6]	[607, 631, 1867, 8971, 10267, 10687, 24151]	600
			[2, 3]	[3, 859, 1723, 2719, 4219, 5503, 8179]	69
			[2, 4]	[3, 2203, 3463, 4423, 14419, 21943, 25219]	1424
			[2, 5]	[3, 2383, 2551, 6163, 8263, 8311, 15091]	253
			[2, 6]	[3, 2383, 4219, 6823, 10039, 20599, 25903]	318
			[3, 4]	[3, 7, 4219, 8179, 10039, 20599, 29179]	1352
			[3, 5]	Intractable	3000
			[3, 6]	[3, 7, 4219, 7159, 8179, 10039, 20599]	422
			[4, 5]	Intractable	3000
			[4, 6]	Intractable	3000
			[5, 6]	[3, 7, 919, 2179, 4159, 37039, 42379]	3372
`Modulo`	1	`Naive`	ϕ	[73, 241, 601, 853, 1021, 1213, 1381]	367
		`SelectAll`	ϕ	[13, 853, 1129, 1213, 2113, 2389, 7333]	**1**
		`ExcludeOne`	[1]	[1873, 3001, 4813, 5521, 21313, 42433, 68053]	79
			[2]	[13, 2113, 2389, 9013, 15493, 18913, 72613]	87
			[3]	[13, 853, 2113, 3121, 4021, 9973, 21661]	11
			[4]	[13, 853, 1129, 3049, 3313, 7993, 10009]	3
			[5]	[13, 853, 1129, 1213, 2389, 3049, 20593]	10
			[6]	[13, 853, 1129, 1213, 2113, 2389, 7333]	2
		`ExcludeTwo`	[1, 2]	[1381, 2341, 2593, 3121, 8521, 9733, 12613]	22
			[1, 3]	[3457, 5641, 5857, 7417, 11821, 36637, 37321]	207
			[1, 4]	[1777, 3457, 5641, 7417, 9337, 23197, 24061]	102
			[1, 5]	[1873, 3001, 9433, 11593, 21313, 23293, 23833]	94
			[1, 6]	[1873, 3001, 4813, 5521, 5881, 7333, 21313]	12
			[2, 3]	[13, 1213, 3373, 4021, 6673, 9901, 14653]	38
			[2, 4]	[13, 2389, 3049, 3733, 9109, 18253, 20593]	86
			[2, 5]	[13, 2113, 2293, 7333, 11353, 16573, 17449]	53
			[2, 6]	[13, 2113, 2389, 9013, 19429, 21529, 36973]	143
			[3, 4]	[13, 853, 2113, 2389, 3313, 6733, 9013]	18
			[3, 5]	[13, 853, 2113, 4021, 8713, 9973, 21661]	97
			[3, 6]	[13, 853, 2113, 3121, 3313, 9013, 11701]	23
			[4, 5]	[13, 853, 1129, 2389, 3049, 3313, 10009]	19

Table A.3 – *Continued from previous page*

Strategy				Solution	Time (s)
Branching		Fixing			
name	arg	name	arg		
			[4, 6]	[13, 853, 1129, 3049, 3313, 7993, 10009]	16
			[5, 6]	[13, 853, 1129, 1213, 2113, 2389, 7333]	12
Modulo	5	Naive	ϕ	[5, 17, 41, 101, 257, 461, 521]	**2**
		SelectAll	ϕ	[5, 17, 881, 1181, 1637, 14957, 24197]	**2**
		ExcludeOne	[1]	[2777, 2897, 2909, 3917, 6389, 44357, 49937]	87
			[2]	[5, 113, 929, 1193, 1949, 8669, 29333]	28
			[3]	[5, 17, 1109, 1277, 1709, 16217, 16229]	12
			[4]	[5, 17, 881, 2357, 4877, 28097, 48821]	92
			[5]	[5, 17, 881, 1181, 11717, 14957, 24197]	12
			[6]	[5, 17, 881, 1181, 1637, 14957, 24197]	18
		ExcludeTwo	[1, 2]	[1433, 3413, 3833, 5009, 6329, 14249, 15149]	34
			[1, 3]	[3257, 4637, 5441, 10457, 18521, 19841, 27281]	173
			[1, 4]	[2417, 2909, 6257, 6869, 9377, 16529, 20897]	57
			[1, 5]	[2777, 2897, 3821, 4157, 19001, 24821, 54101]	1028
			[1, 6]	[2777, 2897, 2909, 3917, 6689, 12269, 23189]	77
			[2, 3]	[5, 1277, 1949, 2897, 6977, 8669, 14489]	52
			[2, 4]	[5, 953, 4133, 6653, 9833, 19433, 23549]	89
			[2, 5]	[5, 113, 389, 1433, 1709, 6269, 6329]	7
			[2, 6]	[5, 113, 929, 1193, 5309, 7589, 14969]	73
			[3, 4]	[5, 17, 1877, 4817, 13121, 14897, 20261]	93
			[3, 5]	[5, 17, 1109, 1709, 2969, 10949, 11597]	16
			[3, 6]	[5, 17, 1109, 1277, 1889, 10457, 15749]	70
			[4, 5]	[5, 17, 881, 3797, 7817, 10457, 11717]	33
			[4, 6]	[5, 17, 881, 2357, 7817, 19697, 44537]	625
			[5, 6]	[5, 17, 881, 1181, 9281, 11717, 14957]	53
Modulo	7	Naive	ϕ	[7, 19, 67, 139, 607, 859, 907]	4
		SelectAll	ϕ	[7, 19, 907, 1699, 1747, 3967, 4759]	**0**
		ExcludeOne	[1]	[2467, 3691, 4987, 13567, 28771, 45307, 75211]	69
			[2]	[7, 3307, 3727, 4159, 5407, 27127, 39727]	51
			[3]	[7, 19, 1747, 3967, 4759, 8707, 16519]	7
			[4]	[7, 19, 907, 3967, 4759, 15427, 16519]	9
			[5]	[7, 19, 907, 1699, 3967, 4759, 15427]	8
			[6]	[7, 19, 907, 1699, 1747, 3967, 4759]	1
		ExcludeTwo	[1, 2]	[1987, 2791, 3547, 3847, 8167, 8887, 9511]	15
			[1, 3]	[3499, 3727, 4759, 8179, 25087, 30259, 38839]	238
			[1, 4]	[2251, 3691, 5443, 7963, 10303, 25171, 31723]	87
			[1, 5]	[2467, 3691, 6211, 11887, 25147, 27091, 33391]	218
			[1, 6]	[2467, 3691, 4987, 13567, 20407, 21211, 56731]	246
			[2, 3]	[7, 607, 967, 2791, 6451, 14407, 30871]	119
			[2, 4]	[7, 3571, 6991, 8191, 10687, 12547, 14071]	22
			[2, 5]	[7, 3307, 3727, 5407, 8179, 10039, 13399]	18
			[2, 6]	[7, 3307, 3727, 4159, 14479, 20599, 21019]	79
			[3, 4]	[7, 19, 1747, 3967, 5347, 8707, 12739]	36
			[3, 5]	[7, 19, 2887, 3739, 11467, 27739, 74719]	2214
			[3, 6]	[7, 19, 1747, 3967, 5347, 8707, 12739]	30
			[4, 5]	[7, 19, 907, 3967, 4759, 15427, 16519]	79

Table A.3 – *Continued from previous page*

Strategy				Solution	Time (s)
Branching		**Fixing**			
name	**arg**	**name**	**arg**		
			[4, 6]	[7, 19, 907, 3967, 4759, 15427, 16519]	73
			[5, 6]	[7, 19, 907, 1699, 1747, 3967, 4759]	2
`Modulo`	11	`Naive`	ϕ	[23, 179, 359, 443, 599, 683, 839]	68
		`SelectAll`	ϕ	[11, 23, 911, 1283, 2711, 2843, 3803]	**0**
		`ExcludeOne`	[1]	[2243, 2939, 3083, 4919, 5099, 9323, 21563]	9
			[2]	[11, 1931, 2207, 5147, 5231, 8171, 9767]	3
			[3]	[11, 23, 2843, 3203, 12611, 83003, 91631]	208
			[4]	[11, 23, 911, 4691, 5903, 6491, 27743]	17
			[5]	[11, 23, 911, 1283, 2843, 5903, 12611]	6
			[6]	[11, 23, 911, 1283, 2711, 2843, 3803]	1
		`ExcludeTwo`	[1, 2]	[2447, 4211, 5867, 5987, 9311, 13007, 47387]	909
			[1, 3]	[2999, 5903, 6299, 7703, 20939, 24419, 25643]	182
			[1, 4]	[2243, 3083, 4919, 5099, 8219, 10739, 24419]	96
			[1, 5]	[2243, 2939, 3083, 5099, 5903, 11483, 44879]	305
			[1, 6]	[2243, 2939, 3083, 4919, 5099, 9323, 21563]	89
			[2, 3]	[11, 2087, 2591, 3011, 15107, 20147, 22787]	108
			[2, 4]	[11, 2111, 5903, 7823, 8663, 9851, 15791]	16
			[2, 5]	[11, 1931, 2207, 5231, 5387, 6491, 8171]	4
			[2, 6]	[11, 1931, 2207, 5147, 5231, 8171, 9767]	9
			[3, 4]	[11, 23, 683, 911, 1283, 2711, 3803]	3
			[3, 5]	[11, 23, 2711, 3203, 4691, 9923, 12503]	17
			[3, 6]	[11, 23, 2843, 3203, 4691, 17483, 24971]	159
			[4, 5]	[11, 23, 911, 2843, 3803, 24971, 28631]	199
			[4, 6]	[11, 23, 911, 4691, 6491, 6911, 26891]	174
			[5, 6]	[11, 23, 911, 1283, 2711, 2843, 3803]	1

TABLE A.4: Computational results for solving model (3.28) using either fixing strategy in Routine 1 or branching strategy in Routine 2. Inputs in Algorithm 3 are $n = 8$; $M_i = 1000$ for $i = 1, \ldots, n$; $T = 3000$; $t = 600$; $\alpha = 0.2$.

Strategy				Solution	Time (s)
Branching		Fixing			
name	**arg**	**name**	**arg**		
`Naive`	ϕ	`Naive`	ϕ	Intractable	3000
		`SelectAll`	ϕ	[3, 7, 919, 2179, 4159, 13399, 37039, 301759]	**25**
		`ExcludeOne`	[1]	Intractable	3000
			[2]	[3, 4219, 6823, 13723, 14083, 25639, 25903, 40039]	44
			[3]	[3, 7, 7159, 8179, 10039, 20599, 22699, 29179]	42
			[4]	Intractable	3000
			[5]	Intractable	3000
			[6]	Intractable	3000
			[7]	Intractable	3000
		`ExcludeTwo`	[1, 2]	Intractable	3000
			[1, 3]	Intractable	3000
			[1, 4]	Intractable	3000
			[1, 5]	[631, 3931, 8971, 10771, 11047, 11827, 18691, 27367]	996
			[1, 6]	Intractable	3000
			[1, 7]	Intractable	3000
			[2, 3]	[3, 2719, 4219, 5503, 8179, 16843, 23623, 27763]	777
			[2, 4]	Intractable	3000
			[2, 5]	Intractable	3000
			[2, 6]	Intractable	3000
			[2, 7]	[3, 4219, 6823, 13723, 14083, 22543, 25639, 25903]	321
			[3, 4]	Intractable	3000
			[3, 5]	Intractable	3000
			[3, 6]	Intractable	3000
			[3, 7]	[3, 7, 7159, 8179, 10039, 20599, 22699, 29179]	286
			[4, 5]	Intractable	3000
			[4, 6]	Intractable	3000
			[4, 7]	Intractable	3000
			[5, 6]	Intractable	3000
			[5, 7]	Intractable	3000
			[6, 7]	Intractable	3000
`Modulo`	1	`Naive`	ϕ	[13, 73, 181, 241, 373, 1693, 1801, 1861]	1579
		`SelectAll`	ϕ	[13, 853, 1129, 1213, 2113, 2389, 7333, 20593]	**1**
		`ExcludeOne`	[1]	[3001, 4813, 5521, 21313, 42433, 68053, 70573, 178693]	254
			[2]	[13, 2389, 9013, 15493, 18913, 72613, 176509, 283009]	1268
			[3]	[13, 853, 3121, 4021, 9973, 21661, 46273, 118093]	259
			[4]	[13, 853, 1129, 3313, 7993, 10009, 20173, 30169]	16
			[5]	[13, 853, 1129, 1213, 3049, 20593, 53353, 128053]	237
			[6]	[13, 853, 1129, 1213, 2113, 7333, 20593, 41113]	42
			[7]	[13, 853, 1129, 1213, 2113, 2389, 7333, 20593]	11
		`ExcludeTwo`	[1, 2]	[2593, 3121, 8521, 9733, 12613, 12781, 46441, 72673]	1026
			[1, 3]	Intractable	3000
			[1, 4]	[3457, 5641, 9337, 23197, 24061, 32797, 34381, 112921]	1757
			[1, 5]	Intractable	3000
			[1, 6]	Intractable	3000

Table A.4 – *Continued from previous page*

Strategy				Solution	Time (s)
Branching		**Fixing**			
name	**arg**	**name**	**arg**		
			[1, 7]	Intractable	3000
			[2, 3]	[13, 4021, 6673, 9901, 14653, 22861, 74713, 78301]	1472
			[2, 4]	[13, 3049, 9109, 18253, 20593, 21433, 21529, 72109]	883
			[2, 5]	[13, 2293, 7333, 16573, 17449, 19009, 24469, 26053]	58
			[2, 6]	[13, 2389, 9013, 19429, 36973, 38149, 45553, 75109]	221
			[2, 7]	[13, 2389, 9013, 15493, 18913, 42373, 54949, 72613]	256
			[3, 4]	[13, 853, 3313, 6733, 9013, 18013, 37573, 38149]	376
			[3, 5]	[13, 853, 4021, 9973, 21661, 46273, 81373, 118093]	1335
			[3, 6]	[13, 853, 3121, 3313, 11701, 24061, 36973, 53281]	283
			[3, 7]	[13, 853, 3121, 4021, 9973, 46273, 57301, 79861]	1040
			[4, 5]	[13, 853, 1129, 3313, 10009, 20173, 30169, 64969]	770
			[4, 6]	[13, 853, 1129, 3313, 10009, 20173, 30169, 64969]	772
			[4, 7]	[13, 853, 1129, 3313, 7993, 8389, 10009, 20173]	30
			[5, 6]	[13, 853, 1129, 1213, 7333, 19009, 20593, 51109]	662
			[5, 7]	[13, 853, 1129, 1213, 3049, 7993, 20593, 45289]	178
			[6, 7]	[13, 853, 1129, 1213, 2113, 2389, 7333, 20593]	69
`Modulo`	5	`Naive`	ϕ	[5, 17, 41, 101, 257, 521, 761, 881]	56
		`SelectAll`	ϕ	[5, 17, 881, 1181, 1637, 14957, 24197, 131297]	**6**
		`ExcludeOne`	[1]	[2897, 2909, 3917, 6389, 44357, 49937, 63809, 65609]	90
			[2]	[5, 929, 1193, 1949, 8669, 29333, 42209, 222533]	861
			[3]	Intractable	3000
			[4]	Intractable	3000
			[5]	[5, 17, 881, 1181, 14957, 24197, 113357, 131297]	340
			[6]	[5, 17, 881, 1181, 1637, 24197, 44537, 108821]	147
			[7]	[5, 17, 881, 1181, 1637, 14957, 51341, 92357]	177
		`ExcludeTwo`	[1, 2]	[3833, 5009, 6329, 14249, 15149, 25253, 33353, 33569]	110
			[1, 3]	Intractable	3000
			[1, 4]	[2909, 6257, 9377, 16529, 20897, 22229, 23057, 57329]	487
			[1, 5]	Intractable	3000
			[1, 6]	Intractable	3000
			[1, 7]	[2897, 2909, 3917, 6389, 44357, 49937, 63809, 65609]	324
			[2, 3]	[5, 2897, 6977, 8669, 14489, 22937, 30449, 37337]	201
			[2, 4]	[5, 4133, 9833, 19433, 23549, 35969, 127709, 217733]	2771
			[2, 5]	[5, 389, 1433, 6269, 6329, 12653, 17609, 65993]	1053
			[2, 6]	[5, 929, 1193, 5309, 14969, 31769, 33773, 77489]	1046
			[2, 7]	[5, 929, 1193, 1949, 8669, 56873, 68633, 84713]	1534
			[3, 4]	[5, 17, 13121, 14897, 20261, 23957, 50321, 98057]	2828
			[3, 5]	[5, 17, 1709, 10949, 11597, 14369, 52769, 64109]	465
			[3, 6]	Intractable	3000
			[3, 7]	Intractable	3000
			[4, 5]	Intractable	3000
			[4, 6]	[5, 17, 881, 7817, 44537, 48821, 92357, 131297]	2164
			[4, 7]	Intractable	3000
			[5, 6]	[5, 17, 881, 1181, 14957, 24197, 113357, 131297]	2870
			[5, 7]	[5, 17, 881, 1181, 14957, 24197, 113357, 131297]	2812
			[6, 7]	[5, 17, 881, 1181, 1637, 28097, 51341, 82361]	2375

Table A.4 – *Continued from previous page*

Strategy				Solution	Time (s)
Branching		Fixing			
name	arg	name	arg		
Modulo	7	Naive	ϕ	[7, 19, 67, 379, 859, 907, 1699, 1747]	2052
		SelectAll	ϕ	[7, 19, 907, 1699, 1747, 3967, 4759, 16519]	**1**
		ExcludeOne	[1]	[3691, 4987, 13567, 28771, 45307, 75211, 111427, 397687]	1747
			[2]	Intractable	3000
			[3]	[7, 19, 3967, 4759, 8707, 16519, 23167, 43987]	25
			[4]	[7, 19, 907, 4759, 15427, 16519, 34747, 56767]	68
			[5]	[7, 19, 907, 1699, 4759, 15427, 16519, 34747]	17
			[6]	[7, 19, 907, 1699, 1747, 4759, 15727, 23167]	14
			[7]	[7, 19, 907, 1699, 1747, 3967, 4759, 16519]	8
		ExcludeTwo	[1, 2]	[3547, 3847, 8167, 8887, 9511, 34171, 34591, 105907]	2408
			[1, 3]	[3727, 8179, 25087, 30259, 38839, 39619, 52267, 64879]	328
			[1, 4]	Intractable	3000
			[1, 5]	[3691, 6211, 11887, 27091, 33391, 66271, 81727, 130687]	1651
			[1, 6]	[3691, 4987, 13567, 20407, 56731, 80347, 103567, 184711]	2043
			[1, 7]	Intractable	3000
			[2, 3]	[7, 2791, 6451, 14407, 30871, 76231, 79231, 95191]	843
			[2, 4]	[7, 6991, 10687, 12547, 14071, 16567, 98731, 155851]	2582
			[2, 5]	[7, 3727, 5407, 10039, 13399, 29587, 38839, 43627]	207
			[2, 6]	Intractable	3000
			[2, 7]	[7, 3727, 4159, 5407, 27127, 38167, 47119, 67867]	422
			[3, 4]	Intractable	3000
			[3, 5]	Intractable	3000
			[3, 6]	[7, 19, 3967, 5347, 12739, 17839, 102679, 125299]	2550
			[3, 7]	[7, 19, 3967, 4759, 8707, 16519, 23167, 43987]	187
			[4, 5]	[7, 19, 907, 15427, 16519, 32359, 56767, 101527]	2404
			[4, 6]	[7, 19, 907, 4759, 16519, 23167, 27739, 37567]	160
			[4, 7]	[7, 19, 907, 4759, 15427, 16519, 34747, 56767]	246
			[5, 6]	[7, 19, 907, 1699, 4759, 15427, 16519, 34747]	283
			[5, 7]	[7, 19, 907, 1699, 4759, 15427, 16519, 34747]	254
			[6, 7]	[7, 19, 907, 1699, 1747, 3967, 4759, 16519]	64
Modulo	11	Naive	ϕ	[11, 83, 251, 263, 443, 1103, 1511, 2111]	2605
		SelectAll	ϕ	[11, 23, 911, 1283, 2711, 2843, 3803, 24971]	**2**
		ExcludeOne	[1]	[2939, 3083, 4919, 5099, 9323, 21563, 34403, 87443]	88
			[2]	[11, 2207, 5147, 5231, 8171, 9767, 21767, 87767]	160
			[3]	Intractable	3000
			[4]	[11, 23, 911, 5903, 6491, 27743, 52511, 61331]	47
			[5]	[11, 23, 911, 1283, 5903, 12611, 49331, 91631]	179
			[6]	[11, 23, 911, 1283, 2711, 3803, 24971, 28631]	27
			[7]	[11, 23, 911, 1283, 2711, 2843, 3803, 24971]	16
		ExcludeTwo	[1, 2]	Intractable	3000
			[1, 3]	[5903, 7703, 20939, 24419, 25643, 33623, 69383, 79379]	542
			[1, 4]	[3083, 4919, 8219, 10739, 24419, 30539, 36599, 42719]	234
			[1, 5]	Intractable	3000
			[1, 6]	[2939, 3083, 4919, 5099, 21563, 26183, 34403, 87443]	593
			[1, 7]	[2939, 3083, 4919, 5099, 9323, 24419, 34403, 52223]	791
			[2, 3]	Intractable	3000

Table A.4 – *Continued from previous page*

Strategy				Solution	Time (s)
Branching		Fixing			
name	**arg**	**name**	**arg**		
			[2, 4]	[11, 5903, 8663, 9851, 15791, 24203, 31583, 108863]	1306
			[2, 5]	[11, 2207, 5231, 6491, 8171, 17471, 35027, 61007]	704
			[2, 6]	[11, 2207, 5147, 5231, 9767, 20771, 21767, 87767]	2140
			[2, 7]	[11, 2207, 5147, 5231, 8171, 9767, 21767, 87767]	2281
			[3, 4]	[11, 23, 1283, 2711, 3803, 21683, 24971, 29063]	307
			[3, 5]	[11, 23, 3203, 9923, 12503, 29243, 32363, 67523]	1242
			[3, 6]	Intractable	3000
			[3, 7]	Intractable	3000
			[4, 5]	Intractable	3000
			[4, 6]	Intractable	3000
			[4, 7]	[11, 23, 911, 5903, 6491, 9851, 27743, 34763]	202
			[5, 6]	[11, 23, 911, 1283, 3803, 5783, 24971, 28631]	196
			[5, 7]	[11, 23, 911, 1283, 5903, 29243, 39971, 57383]	598
			[6, 7]	[11, 23, 911, 1283, 2711, 2843, 3803, 24971]	166

TABLE A.5: Computational results for solving model (3.28) using the fixing strategies in Routine 1 and the branching strategies in Routine 2. Inputs in Algorithm 3 are $n = 9$; $M_i = 1000$ for $i = 1, \ldots, n$; $T = 3000$; $t = 600$; $\alpha = 0.2$.

Strategy				**Solution**	**Time (s)**
Branching		**Fixing**			
name	**arg**	**name**	**arg**		
`Naive`	ϕ	`Naive`	ϕ	Intractable	3000
		`SelectAll`	ϕ	[3, 7, 919, 2179, 4159, 13399, 37039, 301759, 1291159]	**113**
		`ExcludeOne`	[1]	Intractable	3000
			[2]	Intractable	3000
			[3]	Intractable	3000
			[4]	Intractable	3000
			[5]	Intractable	3000
			[6]	Intractable	3000
			[7]	Intractable	3000
			[8]	Intractable	3000
		`ExcludeTwo`	[1, 2]	Intractable	3000
			[1, 3]	Intractable	3000
			[1, 4]	Intractable	3000
			[1, 5]	Intractable	3000
			[1, 6]	Intractable	3000
			[1, 7]	Intractable	3000
			[1, 8]	Intractable	3000
			[2, 3]	Intractable	3000
			[2, 4]	Intractable	3000
			[2, 5]	Intractable	3000
			[2, 6]	Intractable	3000
			[2, 7]	Intractable	3000
			[2, 8]	Intractable	3000
			[3, 4]	Intractable	3000
			[3, 5]	Intractable	3000
			[3, 6]	Intractable	3000
			[3, 7]	Intractable	3000
			[3, 8]	Intractable	3000
			[4, 5]	Intractable	3000
			[4, 6]	Intractable	3000
			[4, 7]	Intractable	3000
			[4, 8]	Intractable	3000
			[5, 6]	Intractable	3000
			[5, 7]	Intractable	3000
			[5, 8]	Intractable	3000
			[6, 7]	Intractable	3000
			[6, 8]	Intractable	3000
			[7, 8]	Intractable	3000
`Modulo`	1	`Naive`	ϕ	Intractable	3000
		`SelectAll`	ϕ	[13, 853, 1129, 1213, 2113, 2389, 7333, 20593, 41113]	**2**
		`ExcludeOne`	[1]	Intractable	3000
			[2]	Intractable	3000
			[3]	Intractable	3000
			[4]	[13, 853, 1129, 7993, 10009, 20173, 30169, 59509, 106453]	123

Table A.5 – *Continued from previous page*

Strategy				Solution	Time (s)
Branching		**Fixing**			
name	**arg**	**name**	**arg**		
			[5]	Intractable	3000
			[6]	[13, 853, 1129, 1213, 2113, 20593, 41113, 72733, 335173]	1852
			[7]	[13, 853, 1129, 1213, 2113, 2389, 20593, 41113, 72733]	67
			[8]	[13, 853, 1129, 1213, 2113, 2389, 7333, 20593, 41113]	37
		`ExcludeTwo`	[1, 2]	Intractable	3000
			[1, 3]	Intractable	3000
			[1, 4]	Intractable	3000
			[1, 5]	Intractable	3000
			[1, 6]	Intractable	3000
			[1, 7]	Intractable	3000
			[1, 8]	Intractable	3000
			[2, 3]	Intractable	3000
			[2, 4]	Intractable	3000
			[2, 5]	[13, 7333, 16573, 19009, 24469, 26053, 28429, 75109, 113329]	682
			[2, 6]	Intractable	3000
			[2, 7]	Intractable	3000
			[2, 8]	Intractable	3000
			[3, 4]	Intractable	3000
			[3, 5]	Intractable	3000
			[3, 6]	[13, 853, 3313, 11701, 36973, 53281, 60601, 135601, 159721]	1619
			[3, 7]	Intractable	3000
			[3, 8]	Intractable	3000
			[4, 5]	Intractable	3000
			[4, 6]	Intractable	3000
			[4, 7]	Intractable	3000
			[4, 8]	[13, 853, 1129, 7993, 10009, 20173, 30169, 59509, 106453]	1518
			[5, 6]	Intractable	3000
			[5, 7]	Intractable	3000
			[5, 8]	Intractable	3000
			[6, 7]	Intractable	3000
			[6, 8]	Intractable	3000
			[7, 8]	[13, 853, 1129, 1213, 2113, 2389, 7333, 20593, 41113]	483
`Modulo`	5	`Naive`	ϕ	Intractable	3000
		`SelectAll`	ϕ	[5, 17, 881, 1181, 1637, 14957, 24197, 131297, 184721]	**9**
		`ExcludeOne`	[1]	[2909, 3917, 6389, 44357, 49937, 63809, 65609, 134369, 414077]	2595
			[2]	Intractable	3000
			[3]	Intractable	3000
			[4]	Intractable	3000
			[5]	Intractable	3000
			[6]	Intractable	3000
			[7]	Intractable	3000
			[8]	[5, 17, 881, 1181, 1637, 14957, 24197, 131297, 184721]	731
		`ExcludeTwo`	[1, 2]	Intractable	3000
			[1, 3]	Intractable	3000
			[1, 4]	Intractable	3000
			[1, 5]	Intractable	3000

Table A.5 – *Continued from previous page*

Strategy				Solution	Time (s)
Branching		Fixing			
name	arg	name	arg		
			[1, 6]	Intractable	3000
			[1, 7]	Intractable	3000
			[1, 8]	Intractable	3000
			[2, 3]	Intractable	3000
			[2, 4]	Intractable	3000
			[2, 5]	Intractable	3000
			[2, 6]	Intractable	3000
			[2, 7]	Intractable	3000
			[2, 8]	Intractable	3000
			[3, 4]	Intractable	3000
			[3, 5]	Intractable	3000
			[3, 6]	Intractable	3000
			[3, 7]	Intractable	3000
			[3, 8]	Intractable	3000
			[4, 5]	Intractable	3000
			[4, 6]	Intractable	3000
			[4, 7]	Intractable	3000
			[4, 8]	Intractable	3000
			[5, 6]	Intractable	3000
			[5, 7]	Intractable	3000
			[5, 8]	Intractable	3000
			[6, 7]	Intractable	3000
			[6, 8]	[5, 17, 881, 1181, 1637, 44537, 51341, 92357, 107621]	505
			[7, 8]	[5, 17, 881, 1181, 1637, 14957, 24197, 131297, 184721]	3089
`Modulo`	7	`Naive`	ϕ	Intractable	3000
		`SelectAll`	ϕ	[7, 19, 907, 1699, 1747, 3967, 4759, 16519, 23167]	**2**
		`ExcludeOne`	[1]	[4987, 13567, 28771, 45307, 75211, 111427, 397687, 405967, 677587]	2088
			[2]	Intractable	3000
			[3]	Intractable	3000
			[4]	Intractable	3000
			[5]	Intractable	3000
			[6]	[7, 19, 907, 1699, 1747, 15727, 23167, 32359, 41479]	27
			[7]	[7, 19, 907, 1699, 1747, 3967, 16519, 23167, 101527]	186
			[8]	[7, 19, 907, 1699, 1747, 3967, 4759, 16519, 23167]	11
		`ExcludeTwo`	[1, 2]	Intractable	3000
			[1, 3]	Intractable	3000
			[1, 4]	Intractable	3000
			[1, 5]	Intractable	3000
			[1, 6]	Intractable	3000
			[1, 7]	Intractable	3000
			[1, 8]	[4987, 13567, 28771, 45307, 75211, 111427, 116191, 280411, 397687]	2181
			[2, 3]	Intractable	3000
			[2, 4]	Intractable	3000
			[2, 5]	Intractable	3000
			[2, 6]	Intractable	3000
			[2, 7]	Intractable	3000

Table A.5 – *Continued from previous page*

Strategy				Solution	Time (s)
Branching		**Fixing**			
name	**arg**	**name**	**arg**		
			[2, 8]	Intractable	3000
			[3, 4]	Intractable	3000
			[3, 5]	Intractable	3000
			[3, 6]	Intractable	3000
			[3, 7]	Intractable	3000
			[3, 8]	Intractable	3000
			[4, 5]	Intractable	3000
			[4, 6]	Intractable	3000
			[4, 7]	Intractable	3000
			[4, 8]	Intractable	3000
			[5, 6]	Intractable	3000
			[5, 7]	Intractable	3000
			[5, 8]	[7, 19, 907, 1699, 15427, 16519, 32359, 81619, 101527]	1841
			[6, 7]	[7, 19, 907, 1699, 1747, 16519, 23167, 32359, 101527]	1731
			[6, 8]	[7, 19, 907, 1699, 1747, 15727, 23167, 32359, 41479]	85
			[7, 8]	[7, 19, 907, 1699, 1747, 3967, 4759, 16519, 23167]	121
`Modulo`	11	`Naive`	ϕ	Intractable	3000
		`SelectAll`	ϕ	[11, 23, 911, 1283, 2711, 2843, 3803, 24971, 28631]	**2**
		`ExcludeOne`	[1]	[3083, 4919, 5099, 9323, 21563, 34403, 87443, 97943, 584183]	3087
			[2]	Intractable	3000
			[3]	Intractable	3000
			[4]	Intractable	3000
			[5]	Intractable	3000
			[6]	Intractable	3000
			[7]	[11, 23, 911, 1283, 2711, 2843, 24971, 28631, 406403]	2639
			[8]	[11, 23, 911, 1283, 2711, 2843, 3803, 24971, 28631]	17
		`ExcludeTwo`	[1, 2]	Intractable	3000
			[1, 3]	Intractable	3000
			[1, 4]	[4919, 8219, 24419, 30539, 36599, 42719, 72287, 89759, 129419]	2060
			[1, 5]	Intractable	3000
			[1, 6]	Intractable	3000
			[1, 7]	Intractable	3000
			[1, 8]	Intractable	3000
			[2, 3]	Intractable	3000
			[2, 4]	Intractable	3000
			[2, 5]	Intractable	3000
			[2, 6]	Intractable	3000
			[2, 7]	Intractable	3000
			[2, 8]	Intractable	3000
			[3, 4]	Intractable	3000
			[3, 5]	Intractable	3000
			[3, 6]	Intractable	3000
			[3, 7]	Intractable	3000
			[3, 8]	Intractable	3000
			[4, 5]	Intractable	3000
			[4, 6]	Intractable	3000

Table A.5 – *Continued from previous page*

Strategy				Solution	Time (s)
Branching		Fixing			
name	arg	name	arg		
			[4, 7]	Intractable	3000
			[4, 8]	Intractable	3000
			[5, 6]	Intractable	3000
			[5, 7]	Intractable	3000
			[5, 8]	Intractable	3000
			[6, 7]	[11, 23, 911, 1283, 2711, 24971, 34031, 57383, 117503]	2525
			[6, 8]	[11, 23, 911, 1283, 2711, 24971, 34031, 57383, 117503]	2388
			[7, 8]	[11, 23, 911, 1283, 2711, 2843, 3803, 24971, 28631]	54

TABLE A.6: Computational results for solving model (3.28) using the fixing strategies in Routine 1 and the branching strategies in Routine 2. Inputs in Algorithm 3 are $n = 10$; $M_i = 1000$ for $i = 1, \ldots, n$; $T = 6000$; $t = 3000$; $\alpha = 0.2$.

Strategy				**Solution**	**Time (s)**
Branching		**Fixing**			
name	**arg**	**name**	**arg**		
`Naive`	ϕ	`Naive`	ϕ	Intractable	6000
		`SelectAll`	ϕ	[3, 7, 919, 2179, 4159, 13399, 37039, 301759, 1291159, 1765759]	**156**
		`ExcludeOne`	[1]	Intractable	6000
			[2]	Intractable	6000
			[3]	Intractable	6000
			[4]	Intractable	6000
			[5]	Intractable	6000
			[6]	Intractable	6000
			[7]	Intractable	6000
			[8]	Intractable	6000
			[9]	Intractable	6000
		`ExcludeTwo`	[1, 2]	Intractable	6000
			[1, 3]	Intractable	6000
			[1, 4]	Intractable	6000
			[1, 5]	Intractable	6000
			[1, 6]	Intractable	6000
			[1, 7]	Intractable	6000
			[1, 8]	Intractable	6000
			[1, 9]	Intractable	6000
			[2, 3]	Intractable	6000
			[2, 4]	Intractable	6000
			[2, 5]	Intractable	6000
			[2, 6]	Intractable	6000
			[2, 7]	Intractable	6000
			[2, 8]	Intractable	6000
			[2, 9]	Intractable	6000
			[3, 4]	Intractable	6000
			[3, 5]	Intractable	6000
			[3, 6]	Intractable	6000
			[3, 7]	Intractable	6000
			[3, 8]	Intractable	6000
			[3, 9]	Intractable	6000
			[4, 5]	Intractable	6000
			[4, 6]	Intractable	6000
			[4, 7]	Intractable	6000
			[4, 8]	Intractable	6000
			[4, 9]	Intractable	6000
			[5, 6]	Intractable	6000
			[5, 7]	Intractable	6000
			[5, 8]	Intractable	6000
			[5, 9]	Intractable	6000
			[6, 7]	Intractable	6000
			[6, 8]	Intractable	6000
			[6, 9]	Intractable	6000

Table A.6 – *Continued from previous page*

Strategy				Solution	Time (s)
Branching		Fixing			
name	arg	name	arg		
			[7, 8]	Intractable	6000
			[7, 9]	Intractable	6000
			[8, 9]	Intractable	6000
Modulo	1	Naive	ϕ	Intractable	6000
		SelectAll	ϕ	[13, 853, 1129, 1213, 2113, 2389, 7333, 20593, 41113, 335173]	**18**
		ExcludeOne	[1]	Intractable	6000
			[2]	Intractable	6000
			[3]	Intractable	6000
			[4]	Intractable	6000
			[5]	Intractable	6000
			[6]	Intractable	6000
			[7]	[13, 853, 1129, 1213, 2113, 2389, 41113, 72733, 85669, 335173]	1147
			[8]	Intractable	6000
			[9]	[13, 853, 1129, 1213, 2113, 2389, 7333, 20593, 41113, 335173]	2394
		ExcludeTwo	[1, 2]	Intractable	6000
			[1, 3]	Intractable	6000
			[1, 4]	Intractable	6000
			[1, 5]	Intractable	6000
			[1, 6]	Intractable	6000
			[1, 7]	Intractable	6000
			[1, 8]	Intractable	6000
			[1, 9]	Intractable	6000
			[2, 3]	Intractable	6000
			[2, 4]	Intractable	6000
			[2, 5]	Intractable	6000
			[2, 6]	Intractable	6000
			[2, 7]	Intractable	6000
			[2, 8]	Intractable	6000
			[2, 9]	Intractable	6000
			[3, 4]	Intractable	6000
			[3, 5]	Intractable	6000
			[3, 6]	Intractable	6000
			[3, 7]	Intractable	6000
			[3, 8]	Intractable	6000
			[3, 9]	Intractable	6000
			[4, 5]	Intractable	6000
			[4, 6]	Intractable	6000
			[4, 7]	Intractable	6000
			[4, 8]	Intractable	6000
			[4, 9]	Intractable	6000
			[5, 6]	Intractable	6000
			[5, 7]	Intractable	6000
			[5, 8]	Intractable	6000
			[5, 9]	Intractable	6000
			[6, 7]	Intractable	6000
			[6, 8]	Intractable	6000

Table A.6 – *Continued from previous page*

Strategy				Solution	Time (s)
Branching		Fixing			
name	arg	name	arg		
			[6, 9]	Intractable	6000
			[7, 8]	Intractable	6000
			[7, 9]	Intractable	6000
			[8, 9]	Intractable	6000
Modulo	5	Naive	ϕ	Intractable	6000
		SelectAll	ϕ	[5, 17, 881, 1181, 1637, 14957, 24197, 131297, 184721, 6353021]	**299**
		ExcludeOne	[1]	Intractable	6000
			[2]	Intractable	6000
			[3]	Intractable	6000
			[4]	Intractable	6000
			[5]	Intractable	6000
			[6]	Intractable	6000
			[7]	Intractable	6000
			[8]	Intractable	6000
			[9]	Intractable	6000
		ExcludeTwo	[1, 2]	Intractable	6000
			[1, 3]	Intractable	6000
			[1, 4]	Intractable	6000
			[1, 5]	Intractable	6000
			[1, 6]	Intractable	6000
			[1, 7]	Intractable	6000
			[1, 8]	Intractable	6000
			[1, 9]	Intractable	6000
			[2, 3]	Intractable	6000
			[2, 4]	Intractable	6000
			[2, 5]	Intractable	6000
			[2, 6]	Intractable	6000
			[2, 7]	Intractable	6000
			[2, 8]	Intractable	6000
			[2, 9]	Intractable	6000
			[3, 4]	Intractable	6000
			[3, 5]	Intractable	6000
			[3, 6]	Intractable	6000
			[3, 7]	Intractable	6000
			[3, 8]	Intractable	6000
			[3, 9]	Intractable	6000
			[4, 5]	Intractable	6000
			[4, 6]	Intractable	6000
			[4, 7]	Intractable	6000
			[4, 8]	Intractable	6000
			[4, 9]	Intractable	6000
			[5, 6]	Intractable	6000
			[5, 7]	Intractable	6000
			[5, 8]	Intractable	6000
			[5, 9]	Intractable	6000
			[6, 7]	Intractable	6000

Table A.6 – *Continued from previous page*

Strategy				Solution	Time (s)
Branching		Fixing			
name	arg	name	arg		
			[6, 8]	Intractable	6000
			[6, 9]	Intractable	6000
			[7, 8]	Intractable	6000
			[7, 9]	Intractable	6000
			[8, 9]	Intractable	6000
`Modulo`	7	`Naive`	ϕ	Intractable	6000
		`SelectAll`	ϕ	[7, 19, 907, 1699, 1747, 3967, 4759, 16519, 23167, 4891507]	**221**
		`ExcludeOne`	[1]	Intractable	6000
			[2]	Intractable	6000
			[3]	Intractable	6000
			[4]	Intractable	6000
			[5]	Intractable	6000
			[6]	Intractable	6000
			[7]	Intractable	6000
			[8]	Intractable	6000
			[9]	Intractable	6000
		`ExcludeTwo`	[1, 2]	Intractable	6000
			[1, 3]	Intractable	6000
			[1, 4]	Intractable	6000
			[1, 5]	Intractable	6000
			[1, 6]	Intractable	6000
			[1, 7]	Intractable	6000
			[1, 8]	Intractable	6000
			[1, 9]	Intractable	6000
			[2, 3]	Intractable	6000
			[2, 4]	Intractable	6000
			[2, 5]	Intractable	6000
			[2, 6]	Intractable	6000
			[2, 7]	Intractable	6000
			[2, 8]	Intractable	6000
			[2, 9]	Intractable	6000
			[3, 4]	Intractable	6000
			[3, 5]	Intractable	6000
			[3, 6]	Intractable	6000
			[3, 7]	Intractable	6000
			[3, 8]	Intractable	6000
			[3, 9]	Intractable	6000
			[4, 5]	Intractable	6000
			[4, 6]	Intractable	6000
			[4, 7]	Intractable	6000
			[4, 8]	Intractable	6000
			[4, 9]	Intractable	6000
			[5, 6]	Intractable	6000
			[5, 7]	Intractable	6000
			[5, 8]	Intractable	6000
			[5, 9]	Intractable	6000

Table A.6 – *Continued from previous page*

Strategy				Solution	Time (s)
Branching		**Fixing**			
name	**arg**	**name**	**arg**		
			[6, 7]	Intractable	6000
			[6, 8]	Intractable	6000
			[6, 9]	Intractable	6000
			[7, 8]	Intractable	6000
			[7, 9]	Intractable	6000
			[8, 9]	Intractable	6000
`Modulo`	11	`Naive`	ϕ	Intractable	6000
		`SelectAll`	ϕ	[11, 23, 911, 1283, 2711, 2843, 3803, 24971, 28631, 406403]	**22**
		`ExcludeOne`	[1]	Intractable	6000
			[2]	Intractable	6000
			[3]	Intractable	6000
			[4]	Intractable	6000
			[5]	Intractable	6000
			[6]	Intractable	6000
			[7]	Intractable	6000
			[8]	Intractable	6000
			[9]	[11, 23, 911, 1283, 2711, 2843, 3803, 24971, 28631, 406403]	2664
		`ExcludeTwo`	[1, 2]	Intractable	6000
			[1, 3]	Intractable	6000
			[1, 4]	Intractable	6000
			[1, 5]	Intractable	6000
			[1, 6]	Intractable	6000
			[1, 7]	Intractable	6000
			[1, 8]	Intractable	6000
			[1, 9]	Intractable	6000
			[2, 3]	Intractable	6000
			[2, 4]	Intractable	6000
			[2, 5]	Intractable	6000
			[2, 6]	Intractable	6000
			[2, 7]	Intractable	6000
			[2, 8]	Intractable	6000
			[2, 9]	Intractable	6000
			[3, 4]	Intractable	6000
			[3, 5]	Intractable	6000
			[3, 6]	Intractable	6000
			[3, 7]	Intractable	6000
			[3, 8]	Intractable	6000
			[3, 9]	Intractable	6000
			[4, 5]	Intractable	6000
			[4, 6]	Intractable	6000
			[4, 7]	Intractable	6000
			[4, 8]	Intractable	6000
			[4, 9]	Intractable	6000
			[5, 6]	Intractable	6000
			[5, 7]	Intractable	6000
			[5, 8]	Intractable	6000

Table A.6 – *Continued from previous page*

Strategy				Solution	Time (s)
Branching		Fixing			
name	arg	name	arg		
			[5, 9]	Intractable	6000
			[6, 7]	Intractable	6000
			[6, 8]	Intractable	6000
			[6, 9]	Intractable	6000
			[7, 8]	Intractable	6000
			[7, 9]	Intractable	6000
			[8, 9]	Intractable	6000

Appendix B

Computation Results for IPA-SBB method

In this appendix, we report the computation results for all generated instances. Each instance is solved ten times, and we report the geometric mean and standard deviation for the number of nodes and computation time.

m	n	k	LB	UB	Nodes	Time (s)
10	2	2	0.10	0.10	1,083 $\pm$ 90	0.47 $\pm$ 0.05
10	2	3	0.02	0.02	13,190 $\pm$ 1,330	3.28 $\pm$ 0.33
10	3	2	0.01	0.01	2,079 $\pm$ 195	0.64 $\pm$ 0.05
10	3	3	0.00	0.00	83,647 $\pm$ 143	21.21 $\pm$ 0.29
14	2	2	0.29	0.29	32,691 $\pm$ 68	9.31 $\pm$ 0.16
14	2	3	0.40	0.40	35,060 $\pm$ 626	10.29 $\pm$ 0.43
14	3	2	0.11	0.11	40,197 $\pm$ 106	12.07 $\pm$ 0.54
14	3	3	0.00	0.00	366,724 $\pm$ 9,602	104.53 $\pm$ 2.98
18	2	2	1.64	1.64	31,802 $\pm$ 242	9.07 $\pm$ 0.57
18	2	3	0.76	0.76	201,748 $\pm$ 995	77.72 $\pm$ 0.67
18	3	2	0.37	0.37	803,623 $\pm$ 623	365.41 $\pm$ 5.10
18	3	3	0.00	0.01	5,078,815 $\pm$ 173,513	3600.07 $\pm$ 0.02
22	2	2	1.72	1.72	2,854 $\pm$ 0	0.95 $\pm$ 0.03
22	2	3	1.02	1.02	265,176 $\pm$ 1,394	144.04 $\pm$ 0.97
22	3	2	0.01	0.16	4,601,016 $\pm$ 103,325	3600.18 $\pm$ 0.14
22	3	3	0.00	0.05	4,592,308 $\pm$ 128,850	3600.21 $\pm$ 0.15

TABLE B.1: Results for the `LowDim` dataset under low noise. Each entry reports geometric mean $\pm$ standard deviation over ten runs.

m	n	k	LB	UB	Nodes	Time (s)
10	2	2	0.10	0.10	$1{,}298 \pm 73$	0.47 ± 0.05
10	2	3	0.02	0.02	$19{,}920 \pm 2{,}292$	4.57 ± 0.54
10	3	2	0.01	0.01	$1{,}883 \pm 117$	0.63 ± 0.04
10	3	3	0.00	0.00	$13{,}985 \pm 2{,}735$	3.56 ± 0.72
14	2	2	0.29	0.29	$27{,}531 \pm 94$	8.08 ± 0.07
14	2	3	0.57	0.57	$13{,}184 \pm 0$	3.85 ± 0.15
14	3	2	0.11	0.11	$22{,}619 \pm 612$	7.42 ± 0.38
14	3	3	0.00	0.00	$907{,}754 \pm 3{,}981$	309.87 ± 1.97
18	2	2	1.72	1.72	$135{,}428 \pm 0$	34.04 ± 0.90
18	2	3	1.15	1.15	$39{,}064 \pm 1$	14.65 ± 0.15
18	3	2	0.37	0.37	$756{,}989 \pm 1{,}884$	373.59 ± 5.62
18	3	3	0.01	0.01	$3{,}655{,}200 \pm 63{,}838$	2480.61 ± 129.60
22	2	2	1.72	1.72	446 ± 0	0.25 ± 0.01
22	2	3	1.56	1.56	$8{,}749 \pm 6$	6.01 ± 0.20
22	3	2	0.00	0.16	$4{,}111{,}212 \pm 81{,}203$	3600.26 ± 0.25
22	3	3	0.00	0.14	$3{,}389{,}661 \pm 56{,}817$	3600.28 ± 0.20

TABLE B.2: Results for the `LowDim` dataset under medium noise. Each entry reports geometric mean $\pm$ standard deviation over ten runs.

m	n	k	LB	UB	Nodes	Time (s)
10	2	2	0.10	0.10	$1{,}343 \pm 184$	0.50 ± 0.07
10	2	3	0.02	0.02	$17{,}831 \pm 2{,}519$	4.78 ± 0.70
10	3	2	0.02	0.02	$2{,}605 \pm 192$	1.01 ± 0.08
10	3	3	0.00	0.00	$1{,}362 \pm 523$	1.16 ± 0.19
14	2	2	0.29	0.29	$28{,}279 \pm 33$	8.75 ± 0.29
14	2	3	0.57	0.57	$14{,}972 \pm 0$	4.86 ± 0.17
14	3	2	0.11	0.11	$24{,}914 \pm 111$	8.91 ± 0.15
14	3	3	0.00	0.00	$930{,}370 \pm 4{,}621$	360.21 ± 2.24
18	2	2	1.97	1.97	$7{,}604 \pm 0$	2.57 ± 0.04
18	2	3	1.15	1.15	$25{,}320 \pm 0$	11.33 ± 0.22
18	3	2	0.37	0.37	$743{,}349 \pm 2{,}482$	368.92 ± 3.84
18	3	3	0.01	0.01	$3{,}551{,}596 \pm 25{,}475$	2406.63 ± 66.04
22	2	2	1.72	1.72	550 ± 0	0.31 ± 0.02
22	2	3	1.67	1.67	$5{,}008 \pm 0$	3.95 ± 0.08
22	3	2	0.01	0.16	$3{,}777{,}169 \pm 34{,}350$	3600.29 ± 0.25
22	3	3	0.00	0.27	$3{,}260{,}203 \pm 59{,}678$	3600.34 ± 0.26

TABLE B.3: Results for the `LowDim` dataset under high noise. Each entry reports geometric mean $\pm$ standard deviation over ten runs.

m	n	k	LB	UB	Nodes	Time (s)
10	2	4	0.00	0.00	88,180 ± 132	23.20 ± 0.49
10	4	2	0.00	0.00	4,687 ± 72	1.26 ± 0.06
11	2	4	0.00	0.00	326,186 ± 787	93.95 ± 1.01
11	2	5	0.00	0.00	3,078,900 ± 1,867	1510.76 ± 45.05
11	4	2	0.01	0.01	10,932 ± 94	3.39 ± 0.13
12	2	4	0.03	0.03	330,377 ± 1,150	103.17 ± 1.59
12	2	5	0.00	0.00	4,475,905 ± 4,353	1743.10 ± 15.18
12	4	2	0.02	0.02	40,597 ± 322	14.90 ± 0.67
12	5	2	0.00	0.00	17,969 ± 825	4.94 ± 0.26
13	2	4	0.05	0.05	172,473 ± 6,850	57.28 ± 2.58
13	2	5	0.00	0.00	6,477,451 ± 19,534	2,100.90 ± 65.67
13	3	4	0.00	0.00	2,452,465 ± 2,581,398	1801.75 ± 1895.75
13	4	2	0.01	0.01	18,170 ± 849	4.96 ± 0.37
13	4	3	0.00	0.00	9,541 ± 13,520	2.72 ± 3.76
13	5	2	0.00	0.00	24,291 ± 1,445	6.71 ± 0.48
14	2	4	0.03	0.03	1,787,774 ± 7,230	641.54 ± 7.26
14	2	5	0.00	0.02	8,143,975 ± 217,120	3,600.03 ± 0.02
14	3	4	0.00	0.00	4,239,816 ± 81,624	3600.19 ± 0.17
14	4	2	0.04	0.04	127,102 ± 92	44.52 ± 0.95
14	4	3	0.00	0.00	2,613,587 ± 828,517	1100.95 ± 362.94
14	5	2	0.00	0.00	131,442 ± 1,149	42.86 ± 0.80

TABLE B.4: Results for the `HighDim` dataset under low noise. Each entry reports geometric mean ± standard deviation over ten runs.

m	n	k	LB	UB	Nodes	Time (s)
10	2	4	0.02	0.02	94,105 ± 101	23.89 ± 1.05
10	4	2	0.00	0.00	4,127 ± 142	1.31 ± 0.07
11	2	4	0.02	0.02	167,879 ± 374	48.57 ± 0.17
11	2	5	0.00	0.00	2,558,893 ± 745	1022.78 ± 22.25
11	4	2	0.01	0.01	10,106 ± 107	3.35 ± 0.11
12	2	4	0.14	0.14	646,000 ± 254	212.06 ± 2.10
12	2	5	0.00	0.00	5,208,020 ± 1,252	1784.91 ± 58.91
12	4	2	0.01	0.01	27,569 ± 344	9.52 ± 0.28
12	5	2	0.00	0.00	18,279 ± 321	5.08 ± 0.13
13	2	4	0.14	0.14	141,236 ± 154	49.87 ± 1.01
13	2	5	0.03	0.03	950,912 ± 117,911	437.51 ± 65.52
13	3	4	0.00	0.00	5,181 ± 2,982	2.21 ± 0.62
13	4	2	0.01	0.01	15,897 ± 181	4.41 ± 0.08
13	4	3	0.00	0.00	6,785 ± 2,827	2.15 ± 0.90
13	5	2	0.00	0.00	26,611 ± 820	7.58 ± 0.28
14	2	4	0.03	0.03	845,095 ± 9,188	285.63 ± 2.53
14	2	5	0.33	0.33	7,370,663 ± 1,593	3,022.08 ± 93.74
14	3	4	0.00	0.00	3,976,019 ± 99,848	3600.30 ± 0.12
14	4	2	0.05	0.05	139,896 ± 103	51.62 ± 1.51
14	4	3	0.00	0.00	39,547 ± 9,174	12.78 ± 2.99
14	5	2	0.00	0.00	122,970 ± 739	40.54 ± 0.68

TABLE B.5: Results for the `HighDim` dataset under medium noise. Each entry reports geometric mean ± standard deviation over ten runs.

m	n	k	LB	UB	Nodes	Time (s)
10	2	4	0.02	0.02	$120{,}654 \pm 431$	34.56 ± 0.95
10	4	2	0.00	0.00	$4{,}342 \pm 245$	1.64 ± 0.10
11	2	4	0.08	0.08	$110{,}679 \pm 409$	35.62 ± 0.21
11	2	5	0.00	0.00	$896{,}688 \pm 427{,}210$	400.24 ± 195.82
11	4	2	0.01	0.01	$10{,}046 \pm 42$	5.13 ± 1.01
12	2	4	0.19	0.19	$430{,}573 \pm 75$	147.13 ± 0.91
12	2	5	0.00	0.00	$3{,}648{,}314 \pm 5{,}166$	1783.81 ± 29.24
12	4	2	0.01	0.01	$26{,}390 \pm 413$	9.44 ± 0.23
12	5	2	0.00	0.00	$18{,}093 \pm 1{,}252$	5.86 ± 0.39
13	2	4	0.26	0.26	$100{,}577 \pm 480$	38.56 ± 0.53
13	2	5	0.03	0.03	$950{,}912 \pm 117{,}911$	437.51 ± 65.52
13	3	4	0.00	0.00	737 ± 85	0.87 ± 0.07
13	4	2	0.00	0.00	$16{,}439 \pm 1{,}253$	4.28 ± 0.28
13	4	3	0.00	0.00	$2{,}862 \pm 3{,}140$	1.71 ± 1.09
13	5	2	0.00	0.00	$24{,}799 \pm 1{,}431$	7.91 ± 0.39
14	2	4	0.03	0.03	$1{,}099{,}058 \pm 11{,}277$	350.37 ± 3.01
14	2	5	0.00	0.00	$1{,}546{,}761 \pm 255{,}985$	687.90 ± 114.21
14	3	4	0.00	0.00	$60{,}123 \pm 28{,}464$	22.31 ± 9.46
14	4	2	0.04	0.04	$106{,}588 \pm 9$	40.00 ± 0.82
14	4	3	0.00	0.00	$3{,}041 \pm 793$	2.92 ± 0.36
14	5	2	0.00	0.00	$125{,}776 \pm 961$	46.77 ± 0.73

TABLE B.6: Results for the `HighDim` dataset under high noise. Each entry reports geometric mean $\pm$ standard deviation over ten runs.

References

Tobias Achterberg, Thorsten Koch, and Alexander Martin. Branching rules revisited. *Operations Research Letters*, 33(1):42–54, January 2005. ISSN 0167-6377. .

Sam Aflaki and Serguei Netessine. Strategic investment in renewable energy sources: The effect of supply intermittency. *Manufacturing & Service Operations Management*, 19(3):489–507, July 2017. ISSN 1526-5498. .

Charu C. Aggarwal and Philip S. Yu. Finding generalized projected clusters in high dimensional spaces. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, SIGMOD/PODS00, pages 70–81. ACM, May 2000. .

Charu C. Aggarwal, Joel L. Wolf, Philip S. Yu, Cecilia Procopiuc, and Jong Soo Park. Fast algorithms for projected clustering. *ACM SIGMOD Record*, 28(2):61–72, June 1999. ISSN 0163-5808. .

Seyed Zeinab Aliahmadi, Farnaz Barzinpour, and Mir Saman Pishvaee. A novel bi-objective credibility-based fuzzy model for municipal waste collection with hard time windows. *Journal of Cleaner Production*, 296:126364, May 2021. ISSN 0959-6526. .

Michael G. Allingham and Agnar Sandmo. Income tax evasion: a theoretical analysis. *Journal of Public Economics*, 1(3–4):323–338, November 1972. ISSN 0047-2727. .

Edoardo Amaldi and Stefano Coniglio. A distance-based point-reassignment heuristic for the k-hyperplane clustering problem. *European Journal of Operational Research*, 227(1):22–29, May 2013. ISSN 0377-2217. .

Semih Atakan, Harsha Gangammanavar, and Suvrajeet Sen. Towards a sustainable power grid: Stochastic hierarchical planning for high renewable integration. *European Journal of Operational Research*, 302(1):381–391, October 2022. ISSN 0377-2217. .

Charles Audet, Pierre Hansen, Alejandro Karam, Chi To Ng, and Sylvain Perron. Exact l 2-norm plane separation. *Optimization Letters*, 2(4):483–495, February 2008. ISSN 1862-4480. .

Jean-Claude Bajard, Julien Eynard, M. Anwar Hasan, and Vincent Zucca. *A Full RNS Variant of FV Like Somewhat Homomorphic Encryption Schemes*, pages 423–442. Springer International Publishing, 2017. ISBN 9783319694535. .

Kyri Baker and Bridget Toomey. Efficient relaxations for joint chance constrained ac optimal power flow. *Electric Power Systems Research*, 148:230–236, July 2017. ISSN 0378-7796. .

Egon Balas. *Disjunctive Programming*, pages 3–51. Elsevier, 1979. ISBN 9780080867670. .

Egon Balas, Sebastián Ceria, and Gérard Cornuéjols. Mixed 0-1 programming by lift-and-project in a branch-and-cut framework. *Management Science*, 42(9):1229–1246, September 1996. ISSN 1526-5501. .

Antal Balog. Linear equations in primes. *Mathematika*, 39(2):367–378, 1992. .

Pietro Belotti, Sonia Cafieri, Jon Lee, and Leo Liberti. *Feasibility-Based Bounds Tightening via Fixed Points*, pages 65–76. Springer Berlin Heidelberg, 2010. ISBN 9783642174582. .

Pietro Belotti, Christian Kirches, Sven Leyffer, Jeff Linderoth, James Luedtke, and Ashutosh Mahajan. Mixed-integer nonlinear optimization. *Acta Numerica*, 22:1–131, April 2013. ISSN 1474-0508. .

J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252, December 1962. ISSN 0945-3245. .

Kristin P. Bennett and O. L. Mangasarian. Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods and Software*, 1(1):23–34, January 1992. ISSN 1029-4937. .

Timo Berthold and Gregor Hendel. Learning to scale mixed-integer programs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(5):3661–3668, May 2021. . URL https://ojs.aaai.org/index.php/AAAI/article/view/16482.

Dimitris Bertsimas and Ryan Cory-Wright. A scalable algorithm for sparse portfolio selection. *INFORMS Journal on Computing*, 34(3):1489–1511, May 2022. ISSN 1526-5528. .

Robert E. Bixby. *A brief history of linear and mixed-integer programming computation*, pages 107–121. EMS Press, January 2012. ISBN 9783985475407. .

Lars Blackmore, Masahiro Ono, and Brian C. Williams. Chance-constrained optimal path planning with obstacles. *IEEE Transactions on Robotics*, 27(6):1080–1094, December 2011. ISSN 1941-0468. .

Carlo Bonferroni. Teoria statistica delle classi e calcolo delle probabilita. *Pubblicazioni del R istituto superiore di scienze economiche e commericiali di firenze*, 8:3–62, 1936.

P.S. Bradley and O.L. Mangasarian. k-plane clustering. *Journal of Global Optimization*, 16(1):23–32, 2000. ISSN 0925-5001. .

Y.P. Cai, G.H. Huang, Z.F. Yang, Q.G. Lin, and Q. Tan. Community-scale renewable energy systems planning under uncertainty—an interval chance-constrained programming approach. *Renewable and Sustainable Energy Reviews*, 13(4):721–735, May 2009. ISSN 1364-0321. .

Matthew D. Cain and Stephen B. McKeon. CEO personal risk-taking and corporate policies. *SSRN Electronic Journal*, 2012. ISSN 1556-5068. .

A. Charnes and W. W. Cooper. The strong Minkowski Farkas-Weyl theorem for vector spaces over ordered fields. *Proceedings of the National Academy of Sciences*, 44(9):914–916, September 1958. ISSN 1091-6490. .

A. Charnes, W. W. Cooper, and G. H. Symonds. Cost horizons and certainty equivalents: An approach to stochastic programming of heating oil. *Management Science*, 4(3):235–263, April 1958. ISSN 1526-5501. .

Abraham Charnes and William W Cooper. Chance-constrained programming. *Management science*, 6(1): 73–79, 1959.

Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. *Homomorphic Encryption for Arithmetic of Approximate Numbers*, pages 409–437. Springer International Publishing, 2017. ISBN 9783319706948. .

Rohan Chindooroy, Patrice Muller, and Giovanni Notaro. Company survival following rescue and restructuring State aid. *European Journal of Law and Economics*, 24(2):165–186, September 2007. ISSN 1572-9990. .

V. Chvátal. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete Mathematics*, 4(4): 305–337, April 1973. ISSN 0012-365X. .

Shawn Cole, Xavier Giné, Jeremy Tobacman, Petia Topalova, Robert Townsend, and James Vickery. Barriers to household risk management: Evidence from India. Policy Research Working Paper WPS5504, The World Bank, 2010. URL http://hdl.handle.net/10986/3987. License: CC BY 3.0 IGO.

Michele Conforti, Gérard Cornuéjols, and Giacomo Zambelli. *Integer programming*. Springer International Publishing, 2014. ISBN 9783319110080. .

Roy L. Crum, Dan J. Laughhunn, and John W. Payne. Risk-seeking behavior and its implications for financial models. *Financial Management*, 10(5):20, 1981. ISSN 0046-3892. .

G. Dantzig, R. Fulkerson, and S. Johnson. Solution of a large-scale Traveling-Salesman problem. *Journal of the Operations Research Society of America*, 2(4):393–410, 1954. ISSN 00963984. URL http://www.jstor.org/stable/166695.

Pranav Dass, Harish Sharma, Jagdish Chand Bansal, and Kendall E. Nygard. Meta heuristics for prime factorization problem. In *2013 World Congress on Nature and Biologically Inspired Computing*, pages 126–131. IEEE, August 2013. .

M. W. Dawande and J. N. Hooker. Inference-based sensitivity analysis for mixed integer/linear programming. *Oper. Res.*, 48(4):623–634, August 2000. ISSN 1526-5463. .

D. A. Dawson and D. Sankoff. An inequality for probabilities. *Proceedings of the American Mathematical Society*, 18(3):504–507, June 1967. ISSN 1088-6826. .

Stephan Dempe and Alain Zemkoho, editors. *Bilevel Optimization: Advances and Next Challenges*. Springer International Publishing, 2020. ISBN 9783030521196. .

Department for Environment, Food & Rural Affairs. Residual waste infrastructure capacity note. resreport, GOV.UK, December 2024. URL https://www.gov.uk/government/publications/residual-waste-infrastructure-capacity-note/residual-waste-infrastructure-capacity-note.

Nameirakpam Dhanachandra, Khumanthem Manglem, and Yambem Jina Chanu. Image segmentation using k -means clustering algorithm and subtractive clustering algorithm. *Procedia Computer Science*, 54:764–771, 2015. ISSN 1877-0509. .

S.C. Douglas, S. Amari, and S.-Y. Kung. On gradient adaptation with unit-norm constraints. *IEEE Transactions on Signal Processing*, 48(6):1843–1847, June 2000. ISSN 1053-587X. .

Marco A. Duran and Ignacio E. Grossmann. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming*, 36(3):307–339, October 1986. ISSN 1436-4646. .

Maklawe Essonanawe Edjabou, Giorgia Faraca, Alessio Boldrin, and Thomas Fruergaard Astrup. Temporal and geographical patterns of solid waste collected at recycling centres. *Journal of Environmental Management*, 245:384–397, September 2019. ISSN 0301-4797. .

Love Ekenberg, Magnus Boman, and Joanne Linnerooth-Bayer. General risk constraints. *Journal of Risk Research*, 4(1):31–47, January 2001. ISSN 1466-4461. .

Maria Elbek and Sanne Wøhlk. A variable neighborhood search for the multi-period collection of recyclable materials. *European Journal of Operational Research*, 249(2):540–550, March 2016. ISSN 0377-2217. .

Erhan Erkut and Susan Neuman. Analytical models for locating undesirable facilities. *European Journal of Operational Research*, 40(3):275–291, June 1989. ISSN 0377-2217. .

L. Ferrer-Martí, B. Domenech, A. García-Villoria, and R. Pastor. A MILP model to design hybrid wind–photovoltaic isolated rural electrification projects in developing countries. *European Journal of Operational Research*, 226(2):293–300, April 2013. ISSN 0377-2217. .

FICO. Fico Xpress optimization :: Xpress-optimizer 50.0.3 :: Chapter 7: Control parameters. https://www.fico.com/fico-xpress-optimization/docs/latest/solver/optimizer/HTML/chapter7.html, 2025. Accessed on 2025-07-20.

Avi Fiegenbaum and Howard Thomas. Attitudes toward risk and the risk–return paradox: Prospect theory explanations. *Academy of Management journal*, 31(1):85–106, 1988.

Kenji Nose Filho and João Marcos Travassos Romano. On ℓ_p-norm sparse blind deconvolution. *2014 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2014. URL https://api.semanticscholar.org/CorpusID:1976602.

A. S. Fotheringham. A new set of spatial-interaction models: The theory of competing destinations. *Environment and Planning A: Economy and Space*, 15(1):15–36, January 1983. ISSN 1472-3409. .

Vishal Gaur and Sridhar Seshadri. Hedging inventory risk through market instruments. *Manufacturing & Service Operations Management*, 7(2):103–120, April 2005. ISSN 1526-5498. .

Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, STOC '09, pages 169–178. ACM, May 2009. .

A. M. Geoffrion. Generalized Benders decomposition. *Journal of Optimization Theory and Applications*, 10 (4):237–260, October 1972. ISSN 1573-2878. .

Gunther Glenk and Stefan Reichelstein. Synergistic value in vertically integrated power-to-gas energy systems. *Production and Operations Management*, 29(3):526–546, March 2020. ISSN 1937-5956. .

Mehdi Golari, Neng Fan, and Tongdan Jin. Multistage stochastic optimization for production-inventory planning with intermittent renewable energy. *Production and Operations Management*, 26(3):409–425, March 2017. ISSN 1937-5956. .

Ralph E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society*, 64(5):275–278, September 1958.

Iratxe Gonzalez Aparicio, Andreas Zucker, Francesco Careri, Fabio Monforti Ferrario, Thomas Huld, and Jake Badger. Wind hourly generation time series at country, NUTS 1, NUTS 2 level and bidding zones. European Commission, Joint Research Centre (JRC), 2016. [Dataset] http://data.europa.eu/89h/jrc-emhires-wind-generation-time-series.

Iratxe Gonzalez Aparicio, Thomas Huld, Francesco Careri, Fabio Monforti Ferrario, and Andreas Zucker. Solar hourly generation time series at country, NUTS 1, NUTS 2 level and bidding zones. European Commission, Joint Research Centre (JRC), 2017. [Dataset] http://data.europa.eu/89h/jrc-emhires-solar-generation-time-series.

Gov.uk. Crime and policing bill: retail crime factsheet. techreport, Home Office, Ministry of Justice, July 2025. URL https://www.gov.uk/government/publications/crime-and-policing-bill-2025-factsheets/crime-and-policing-bill-retail-crime-factsheet.

Andrew Granville. A note on sums of primes. *Can. Math. Bulletin*, 33(4):452–454, 1990. .

Xiaogang Guo, Zhejing Bao, Zhijie Li, and Wenjun Yan. Adaptively constrained stochastic model predictive control for the optimal dispatch of microgrid. *Energies*, 11(1):243, January 2018. ISSN 1996-1073. .

Yan Ru Guo and Yan Qin Bai. Two-dimensional k-subspace clustering and its applications on image recognition. *International Journal of Machine Learning and Cybernetics*, 14(8):2671–2683, February 2023. ISSN 1868-808X. .

Gurobi Optimization, LLC. Gurobi optimizer reference manual, 2024. URL https://www.gurobi.com/documentation/current/refman/ub.html.

Oktay Günlük and Yves Pochet. Mixing mixed-integer inequalities. *Mathematical Programming*, 90(3): 429–457, May 2001. ISSN 0025-5610. .

Nils H. Hakansson and Kenneth J. Arrow. Essays in the theory of risk-bearing. *The Journal of Finance*, 27 (5):1193, December 1972. ISSN 0022-1082. .

Hampshire County Council. Proposals relating to Household Wasteand Recycling Centres (HWRCs). resreport, Hampshire County Council, March 2024. URL https://www.hants.gov.uk/aboutthecouncil/haveyoursay/consultations/future-services-consultation/household-waste-recycling-centres.

Jialin Han, Jiaxiang Zhang, Haoyue Guo, and Ning Zhang. Optimizing location-routing and demand allocation in the household waste collection system using a branch-and-price algorithm. *European Journal of Operational Research*, 316(3):958–975, August 2024. ISSN 0377-2217. .

M.K. Harder, N. Stantzos, R. Woodard, and A. Read. Development of a new quality fair access best value performance indicator (BVPI) for recycling services. *Waste Management*, 28(2):299–309, January 2008. ISSN 0956-053X. .

G. H. Hardy and J. E. Littlewood. Some problems of 'partitio numerorum'; III: On the expression of a number as a sum of primes. *Acta Mathematica*, 44(1):1–70, 1923. .

David Harvey. Faster arithmetic for number-theoretic transforms. *Journal of Symbolic Computation*, 60: 113–119, January 2014. ISSN 0747-7171. .

L.C. Hendry and B.G. Kingsman. Production planning systems and their applicability to make-to-order companies. *European Journal of Operational Research*, 40(1):1–15, May 1989. ISSN 0377-2217. .

Fatemeh Hirbod, Tourandokht Karimi, Zahra Mohammadnazari, Masoud Rabbani, and Amir Aghsami. Municipal solid waste management using multiple disposal location-arc routing and waste segregation approach: a real-life case study in England. *Journal of Industrial and Production Engineering*, 41(1):81–100, October 2023. ISSN 2168-1023. .

Shanshan Hu, Gilvan C. Souza, Mark E. Ferguson, and Wenbin Wang. Capacity investment in renewable energy technology with supply intermittency: Data granularity matters! *Manufacturing & Service Operations Management*, 17(4):480–494, October 2015. ISSN 1526-5498. .

Michael Iacono, Kevin Krizek, and Ahmed M. El-Geneidy. Access to destinations: How close is close enough? Estimating accurate distance decay functions for multiple modes and different purposes. resreport, Minnesota Department of Transportation, May 2008. URL https://hdl.handle.net/11299/151329.

Zhan Jiang, Kenneth A. Kim, and Hao Zhang. The effects of corporate bailout on firm performance: International evidence. *Journal of Banking & Finance*, 43:78–96, June 2014. ISSN 0378-4266. .

William C. Jordan and Stephen C. Graves. Principles on the benefits of manufacturing process flexibility. *Management Science*, 41(4):577–594, April 1995. ISSN 1526-5501. .

Michael Jünger, Thomas M. Liebling, Denis Naddef, George Nemhauser, William R. Pulleyblank, Gerhard Reinelt, Giovanni Rinaldi, and Laurence A. Wolsey, editors. *50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art*. Springer Berlin Heidelberg, November 2009. ISBN 9783540682790. .

Daniel Kahneman and Amos Tversky. *Prospect theory: An analysis of decision under risk*, pages 183–214. Cambridge University Press, March 1988. .

Prasit Kailomsom and Charoenchai Khompatraporn. A multi-objective optimization model for multi-facility decisions of infectious waste transshipment and disposal. *Sustainability*, 15(6):4808, March 2023. ISSN 2071-1050. .

Kenichi Kanatani. *Statistical optimization for geometric computation*. Dover Publ., Mineola, NY, dover ed. edition, 2005. ISBN 0486443086. Unabridged and slightly corrected republication of the 2nd (1999) impression of the work, originally published in the "Machine intelligence and pattern recognition" series by Elsevier Science, Amsterdam, in 1996. An errata list has been added to the present edition on pp. xv-xviii.

Tushar Kansal, Suraj Bahuguna, Vishal Singh, and Tanupriya Choudhury. Customer segmentation using k-means clustering. In *2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS)*, pages 135–139. IEEE, December 2018. .

Richard M. Karp. *Reducibility among Combinatorial Problems*, pages 85–103. Springer US, 1972. ISBN 9781468420012. .

Michal Kaut, Hajnalka Vaagen, and Stein W. Wallace. The combined impact of stochastic and correlated activity durations and design uncertainty on project plans. *International Journal of Production Economics*, 233:108015, March 2021. ISSN 0925-5273. .

Markelle Kelly, Rachel Longjohn, and Kolby Nottingham. The UCI Machine Learning Repository, 2025. URL https://archive.ics.uci.edu. Accessed: 2025-10-14.

William R. Kerr, Ramana Nanda, and Matthew Rhodes-Kropf. Entrepreneurship as experimentation. *Journal of Economic Perspectives*, 28(3):25–48, August 2014. ISSN 0895-3309. .

Karlo Knezevic. Generating prime numbers using genetic algorithms. In *2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO)*, pages 1224–1229. IEEE, September 2021. .

H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2 (1–2):83–97, March 1955. ISSN 1931-9193. .

Nirmal Kumar. An alternative computational optimization technique to solve linear and nonlinear Diophantine equations using discrete WQPSO algorithm. *Soft Comput.*, 26(22):12531–12544, June 2022. ISSN 1433-7479. .

R. Kumari, Sheetanshu, M. K. Singh, R. Jha, and N.K. Singh. Anomaly detection in network traffic using k-mean clustering. In *2016 3rd International Conference on Recent Advances in Information Technology (RAIT)*, pages 387–393. IEEE, March 2016. .

Howard Kunreuther. Mitigating disaster losses through insurance. *Journal of Risk and Uncertainty*, 12 (2–3):171–187, May 1996. ISSN 1573-0476. .

A. H. Land and A. G. Doig. An automatic method of solving discrete programming problems. *Econometrica*, 28(3):497, July 1960. ISSN 0012-9682. .

Miguel A Lejeune and Siqian Shen. Multi-objective probabilistically constrained programs with variable risk: Models for multi-portfolio financial optimization. *European Journal of Operational Research*, 252(2): 522–539, 2016.

David M. Levinson and Ajay Kumar. The rational locator: Why travel times have remained stable. *Journal of the American Planning Association*, 60(3):319–332, September 1994. ISSN 1939-0130. .

Changjun Li, Yi Zhang, Pierre Nouvellet, Joseph O. Okoro, Wang Xiao, and Marie K. Harder. Distance is a barrier to recycling – or is it? Surprises from a clean test. *Waste Management*, 108:183–188, May 2020. ISSN 0956-053X. .

Chong Li, Xinfeng Ge, Yuan Zheng, Chang Xu, Yan Ren, Chenguang Song, and Chunxia Yang. Techno-economic feasibility study of autonomous hybrid wind/PV/battery power system for a household in Urumqi, China. *Energy*, 55:263–272, June 2013. ISSN 0360-5442. .

Li-Ming Liu, Yan-Ru Guo, Zhen Wang, Zhi-Min Yang, and Yuan-Hai Shao. k-proximal plane clustering. *International Journal of Machine Learning and Cybernetics*, 8(5):1537–1554, April 2016. ISSN 1868-808X. .

James Luedtke, Shabbir Ahmed, and George L. Nemhauser. An integer programming approach for linear programs with probabilistic constraints. *Mathematical Programming*, 122(2):247–272, October 2008. ISSN 1436-4646. .

Zhi-quan Luo, Wing-kin Ma, Anthony So, Yinyu Ye, and Shuzhong Zhang. Semidefinite relaxation of quadratic optimization problems. *IEEE Signal Processing Magazine*, 27(3):20–34, May 2010. ISSN 1053-5888. .

O.L. Mangasarian. Arbitrary-norm separating plane. *Operations Research Letters*, 24(1–2):15–23, February 1999. ISSN 0167-6377. .

OL Mangasarian and Michael C Ferris. Uniqueness of integer solution of linear equations. *Optimization Letters*, 4:559–565, 2010. .

Harry Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, March 1952. ISSN 1540-6261. .

M. Martin, I.D. Williams, and M. Clark. Social, cultural and structural influences on household waste recycling: A case study. *Resources, Conservation and Recycling*, 48(4):357–395, October 2006. ISSN 0921-3449. .

S. J. Maynard and T. Cherrett. Transport impacts of household waste recycling centres. *Proceedings of the Institution of Civil Engineers - Waste and Resource Management*, 159(1):13–21, February 2006. ISSN 1747-6534. .

Garth P. McCormick. Computability of global solutions to factorable nonconvex programs: Part i — convex underestimating problems. *Mathematical Programming*, 10(1):147–175, December 1976. ISSN 1436-4646. .

Nimrod Megiddo and Arie Tamir. On the complexity of locating linear facilities in the plane. *Operations Research Letters*, 1(5):194–197, November 1982. ISSN 0167-6377. .

Harshit Mehrotra and Saibal Kumar Pal. Using chaos in grey wolf optimizer and application to prime factorization. In *International Conference on Soft Computing for Problem Solving*, 2018. . URL https://api.semanticscholar.org/CorpusID:48361564.

Kent D. Miller. Risk and rationality in entrepreneurial processes. *Strategic Entrepreneurship Journal*, 1(1–2): 57–74, November 2007. ISSN 1932-443X. .

Himanshu Mittal, Avinash Chandra Pandey, Mukesh Saraswat, Sumit Kumar, Raju Pal, and Garv Modwel. A comprehensive survey of image segmentation: clustering methods, performance parameters, and benchmark datasets. *Multimedia Tools and Applications*, 81(24):35001–35026, February 2021. ISSN 1573-7721. .

Sobhan Mostafayi Darmian, Sahar Moazzeni, and Lars Magnus Hvattum. Multi-objective sustainable location-districting for the collection of municipal solid waste: Two case studies. *Computers & Industrial Engineering*, 150:106965, December 2020. ISSN 0360-8352. .

Sanjoy Kumar Nandi and Himangshu Ranjan Ghosh. Prospect of wind–PV-battery hybrid power system as an alternative to grid extension in Bangladesh. *Energy*, 35(7):3040–3047, July 2010. ISSN 0360-5442. .

Arkadi Nemirovski and Alexander Shapiro. Convex approximations of chance constrained programs. *SIAM Journal on Optimization*, 17(4):969–996, January 2007. ISSN 1095-7189. .

Jorge Nocedal and Stephen J. Wright. *Numerical optimization*. Springer series in operations research and financial engineering. Springer, New York, NY, second edition edition, 2006. ISBN 9780387400655.

B. K. Pagnoncelli, S. Ahmed, and A. Shapiro. Sample average approximation method for chance constrained programming: Theory and applications. *Journal of Optimization Theory and Applications*, 142(2):399–416, March 2009a. ISSN 1573-2878. .

Bernardo K Pagnoncelli, Shabbir Ahmed, and Alexander Shapiro. Computational study of a chance constrained portfolio selection problem. *Journal of Optimization Theory and Applications*, 142(2):399–416, 2009b.

Panos M. Pardalos and Stephen A. Vavasis. Quadratic programming with one negative eigenvalue is NP-hard. *Journal of Global Optimization*, 1(1):15–22, 1991. ISSN 1573-2916. .

Jiming Peng and Yu Wei. Approximating k-means-type clustering via semidefinite programming. *SIAM Journal on Optimization*, 18(1):186–205, January 2007. ISSN 1095-7189. .

Jan-Hendrik Piel, Julian F.H. Hamann, André Koukal, and Michael H. Breitner. Promoting the system integration of renewable energies: Toward a decision support system for incentivizing spatially diversified deployment. *Journal of Management Information Systems*, 34(4):994–1022, October 2017. ISSN 1557-928X. .

Carl Pomerance, András Sárközy, and C. Stewart. On divisors of sums of integers. III. *Pac. J. Math.*, 133, June 1988. .

Thierry Post and Haim Levy. Does risk seeking drive stock prices? a stochastic dominance analysis of aggregate investor preferences and beliefs. *Review of Financial Studies*, 18(3):925–953, 2005. ISSN 1465-7368. .

C. Pozo, G. Guillén-Gosálbez, A. Sorribas, and L. Jiménez. A spatial branch-and-bound framework for the global optimization of kinetic models of metabolic networks. *Industrial & Engineering Chemistry Research*, 50(9):5225–5238, December 2010. ISSN 1520-5045. .

A. Alan B. Pritsker, Lawrence J. Waiters, and Philip M. Wolfe. Multiproject scheduling with limited resources: A zero-one programming approach. *Management Science*, 16(1):93–108, September 1969. ISSN 1526-5501. .

András Prékopa. Boole-Bonferroni inequalities and linear programming. *Operations Research*, 36(1): 145–162, February 1988. ISSN 1526-5463. .

András Prékopa. *Stochastic Programming*. Springer Netherlands, 1995. ISBN 9789401730877. .

Guo Pu, Lijuan Wang, Jun Shen, and Fang Dong. A hybrid unsupervised clustering-based anomaly detection method. *Tsinghua Science and Technology*, 26(2):146–153, April 2021. ISSN 1007-0214. .

Qiang Qiu and Guillermo Sapiro. Learning transformations for clustering and classification. *J. Mach. Learn. Res.*, 16(1):187–225, 2015.

Ignacio Quesada and Ignacio E. Grossmann. Global optimization algorithm for heat exchanger networks. *Industrial & Engineering Chemistry Research*, 32(3):487–499, March 1993. ISSN 1520-5045. .

Emma B. Rasiel, Kevin P. Weinfurt, and Kevin A. Schulman. Can prospect theory explain risk-seeking behavior by terminally ill patients? *Medical Decision Making*, 25(6):609–613, November 2005. ISSN 1552-681X. .

Line Roald and Goran Andersson. Chance-constrained AC optimal power flow: Reformulations and efficient algorithms. *IEEE Transactions on Power Systems*, 33(3):2906–2918, May 2018. ISSN 1558-0679. .

R Tyrrell Rockafellar and Johannes O Royset. On buffered failure probability in design and optimization of structures. *Reliability engineering & system safety*, 95(5):499–510, 2010.

R. Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *The Journal of Risk*, 2(3):21–41, 2000. ISSN 1465-1211. .

Diana Romero, Amy Kwan, Justin Swearingen, Sue Nestler, and Neal Cohen. Impact of the closure of a large urban medical center: A qualitative assessment (Part I). *Journal of Community Health*, 37(5): 982–994, March 2012. ISSN 1573-3610. .

Jim Roth, Michael J. McCord, and Dominic Liber. The landscape of microinsurance in the world's 100 poorest countries. techreport, The Microinsurance Centre, LLC, April 2007. URL https://microinsurancecentre.milliman.com/en/Insight/the-landscape-of-microinsurance-in-the-worlds-100-poorest-countries-in-english.

Andrzej Ruszczyński. Advances in risk-averse optimization. In *Theory driven by influential applications*, pages 168–190. INFORMS, 2013.

Hong S. Ryoo and Nikolaos V. Sahinidis. A branch-and-reduce approach to global optimization. *Journal of Global Optimization*, 8(2):107–138, March 1996. ISSN 1573-2916. .

Nikolaos V. Sahinidis. BARON: A general purpose global optimization software package. *Journal of Global Optimization*, 8(2):201–205, March 1996. ISSN 1573-2916. .

Nikolaos V. Sahinidis. *Global Optimization and Constraint Satisfaction: The Branch-and-Reduce Approach*, pages 1–16. Springer Berlin Heidelberg, 2003. ISBN 9783540399018. .

Muhammad Salman, Syed Abdul Rahman Kashif, Muhammad Salman Fakhar, Akhtar Rasool, and Abdulkerim Sherefa Hussen. Optimizing power generation in a hybrid solar wind energy system using a DFIG-based control approach. *Scientific Reports*, 15(1), March 2025. ISSN 2045-2322. .

Paul Salze, Arnaud Banos, Jean-Michel Oppert, Hélène Charreire, Romain Casey, Chantal Simon, Basile Chaix, Dominique Badariotti, and Christiane Weber. Estimating spatial accessibility to facilities on the regional scale: an extended commuting-based interaction potential model. *International Journal of Health Geographics*, 10(1):2, 2011. ISSN 1476-072X. .

Nikola Samardzic and Daniel Sanchez. BitPacker: Enabling high arithmetic efficiency in fully homomorphic encryption accelerators. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, ASPLOS '24, pages 137–150. ACM, April 2024. .

Y. S. Sathe, Meena Pradhan, and S. P. Shah. Inequalities for the probability of the occurrence of at least m out of n events. *Journal of Applied Probability*, 17(4):1127–1132, December 1980. ISSN 1475-6072. .

M. W. P. Savelsbergh. Preprocessing and probing techniques for mixed integer programming problems. *ORSA Journal on Computing*, 6(4):445–454, November 1994. ISSN 2326-3245. .

Malena Schmidt and Bismark Singh. Selectively closing recycling centers in Bavaria: Reforming waste-management policy to reduce disparity. *Networks*, 84(2):148–160, April 2024. ISSN 1097-0037. .

Christian Schmitt and Bismark Singh. Quadratic optimization models for balancing preferential access and fairness: Formulations and optimality conditions. *INFORMS Journal on Computing*, 36(5): 1150–1167, September 2024. ISSN 1526-5528. .

Ashish Sen and Tony E. Smith. *Gravity models of spatial interaction behavior*. Springer Berlin Heidelberg, 1995. ISBN 9783642798801. .

Antonino Sgalambro, Serena Fugaro, and Filippo Santarelli. A soft-constrained multi-objective facility location approach for designing a network of household waste recycling centres in South Yorkshire. *Journal of the Operational Research Society*, pages 1–30, July 2025. ISSN 1476-9360. .

Yuan-Hai Shao, Lan Bai, Zhen Wang, Xiang-Yu Hua, and Nai-Yang Deng. Proximal plane clustering via eigenvalues. In *Proceedings of the International Conference on Information Technology and Quantitative Management (ITQM 2013)*, 2013. URL https://api.semanticscholar.org/CorpusID:34554212.

Saber Shiripour, Iraj Mahdavi, M. Amiri-Aref, M. Mohammadnia-Otaghsara, and Nezam Mahdavi-Amiri. Multi-facility location problems in the presence of a probabilistic line barrier: a mixed integer quadratic programming model. *International Journal of Production Research*, 50(15): 3988–4008, August 2012. ISSN 1366-588X. .

W. Sierpiński. *250 problems in elementary number theory*. Modern analytic and computational methods in science and mathematics. American Elsevier Publishing Company, 1970. ISBN 9780444000712.

Bismark Singh. Tighter reformulations using classical dawson and sankoff bounds for approximating two-stage chance-constrained programs. *Optimization Letters*, 15(2):327–336, May 2020. ISSN 1862-4480. .

Bismark Singh. Supermodularity, curvature, and convex relaxations in a class of quadratic binary optimization problems. *peer-review*, October 2025. URL https://optimization-online.org/?p=32538.

Bismark Singh and Bernard Knueven. Lagrangian relaxation based heuristics for a chance-constrained optimization model of a hybrid solar-battery storage system. *Journal of Global Optimization*, 80(4): 965–989, June 2021. ISSN 1573-2916. .

Bismark Singh and David Pozo. A guide to solar power forecasting using ARMA models. In *2019 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe)*, pages 1–4. IEEE, September 2019. .

Bismark Singh and Jean-Paul Watson. Approximating two-stage chance-constrained programs with classical probability bounds. *Optimization Letters*, 13(6):1403–1416, March 2019. ISSN 1862-4480. .

Bismark Singh, David P. Morton, and Surya Santoso. An adaptive model with joint chance constraints for a hybrid wind-conventional generator system. *Computational Management Science*, 15(3-4):563–582, April 2018. ISSN 1619-6988. .

E.M.B. Smith and C.C. Pantelides. A symbolic reformulation/spatial branch-and-bound algorithm for the global optimisation of nonconvex MINLPs. *Computers & Chemical Engineering*, 23(4–5):457–478, May 1999. ISSN 0098-1354. .

Arthur Snow and Ronald S. Warren. Tax evasion under random audits with uncertain detection. *Economics Letters*, 88(1):97–100, July 2005. ISSN 0165-1765. .

Andrea Spinelli, Francesca Maggioni, Tânia Rodrigues Pereira Ramos, Ana Paula Barbosa-Póvoa, and Daniele Vigo. A rolling horizon heuristic approach for a multi-stage stochastic waste collection problem. *European Journal of Operational Research*, 323(1):276–296, May 2025. ISSN 0377-2217. .

Nur Sunar and John R. Birge. Strategic commitment to a production schedule with uncertain supply and demand: Renewable energy in day-ahead electricity markets. *Management Science*, 65(2):714–734, February 2019. ISSN 1526-5501. .

J. J. Sylvester. On Tchebycheff's theory of the totality of the prime numbers comprised within given limits. *American Journal of Mathematics*, 4(1/4):230, 1881. ISSN 0002-9327. .

Kayalvily Tabianan, Shubashini Velu, and Vinayakumar Ravi. K-means clustering approach for intelligent customer segmentation using customer purchase behavior data. *Sustainability*, 14(12):7243, June 2022. ISSN 2071-1050. .

Trung Hieu Tran, Thu Ba T. Nguyen, Hoa Sen T. Le, and Duc Chinh Phung. Formulation and solution technique for agricultural waste collection and transport network design. *European Journal of Operational Research*, 313(3):1152–1169, March 2024. ISSN 0377-2217. .

P. Tseng. Nearest q-flat to m points. *Journal of Optimization Theory and Applications*, 105(1):249–252, April 2000. ISSN 1573-2878. .

Michael P. Vitus and Claire J. Tomlin. On feedback design and risk allocation in chance constrained control. In *IEEE Conference on Decision and Control and European Control Conference*, pages 734–739. IEEE, December 2011. .

Dingding Wang, Chris Ding, and Tao Li. *K-Subspace Clustering*, pages 506–521. Springer Berlin Heidelberg, 2009. ISBN 9783642041747. .

Juyoung Wang, Mucahit Cevik, Saman Hassanzadeh Amin, and Amir Ali Parsaee. Mixed-integer linear programming models for the paint waste management problem. *Transportation Research Part E: Logistics and Transportation Review*, 151:102343, July 2021. ISSN 1366-5545. .

Qianfan Wang, Yongpei Guan, and Jianhui Wang. A chance-constrained two-stage stochastic program for unit commitment with uncertain wind power output. *IEEE Transactions on Power Systems*, 27(1): 206–215, February 2012. ISSN 1558-0679. .

Yilin Wang, Yankui Liu, and Huili Pei. Designing a new robust solid waste recycling network under uncertainty: A case study about circular economy transition. *Socio-Economic Planning Sciences*, 96: 102066, December 2024. ISSN 0038-0121. .

David G. Wells. *Prime numbers: The most mysterious figures in math.* Wiley Hoboken, Hoboken, NJ, 2005. ISBN 0471462349.

Narong Wichapa and Porntep Khokhajaikiat. Using the hybrid fuzzy goal programming model and hybrid genetic algorithm to solve a multi-objective location routing problem for infectious waste disposal. *Journal of Industrial Engineering and Management*, 10(5):853, November 2017a. ISSN 2013-8423. .

Narong Wichapa and Porntep Khokhajaikiat. Solving multi-objective facility location problem using the fuzzy analytical hierarchy process and goal programming: a case study on infectious waste disposal centers. *Operations Research Perspectives*, 4:39–48, 2017b. ISSN 2214-7160. .

I.D. Williams and C. Taylor. Maximising household waste recycling at civic amenity sites in lancashire, england. *Waste Management*, 24(9):861–874, January 2004. ISSN 0956-053X. .

Michael Wolff, Tristan Becker, and Grit Walther. Long-term design and analysis of renewable fuel supply chains — an integrated approach considering seasonal resource availability. *European Journal of Operational Research*, 304(2):745–762, January 2023. ISSN 0377-2217. .

Laurence A. Wolsey. *Integer programming*. Wiley, New York, 1998. ISBN 0471283665. .

R. Woodard, M. Bench, M.K. Harder, and N. Stantzos. The optimisation of household waste recycling centres for increased recycling—a case study in Sussex, UK. *Resources, Conservation and Recycling*, 43 (1):75–93, December 2004. ISSN 0921-3449. .

Shenming Xie, Ying Terk Lim, Huiwen Wang, Wen Yi, and Maxwell Fordjour Antwi-Afari. Location and capacity optimization of waste recycling centers: Mathematical models and numerical experiments. *Applied Sciences*, 14(16):7039, August 2024. ISSN 2076-3417. .

Jing Xu, Alan T. Murray, Richard L. Church, and Ran Wei. Service allocation equity in location coverage analytics. *European Journal of Operational Research*, 305(1):21–37, February 2023. ISSN 0377-2217. .

Vinay Yadav, A.K. Bhurjee, Subhankar Karmakar, and A.K. Dikshit. A facility location model for municipal solid waste management system under uncertain environment. *Science of The Total Environment*, 603–604:760–771, December 2017. ISSN 0048-9697. .

Linfeng Yang, Jinbao Jian, Yuanyuan Wang, and Zhaoyang Dong. Projected mixed integer programming formulations for unit commitment problem. *International Journal of Electrical Power & Energy Systems*, 68:195–202, June 2015. ISSN 0142-0615. .

Arsalan Yousefloo, Reza Babazadeh, Mehrdad Mohammadi, Amir Pirayesh, and Alexandre Dolgui. Design of a robust waste recycling network integrating social and environmental pillars of sustainability. *Computers & Industrial Engineering*, 176:108970, February 2023. ISSN 0360-8352. .

Hao Yu and Wei Deng Solvang. A multi-objective location-allocation optimization for sustainable management of municipal solid waste. *Environment Systems and Decisions*, 37(3):289–308, February 2017. ISSN 2194-5411. .

Zati Aqmar Zaharudin, Andrew Brint, Andrea Genovese, and Carmela Piccolo. A spatial interaction model for the representation of user access to household waste recycling centres. *Resources, Conservation and Recycling*, 168:105438, May 2021. ISSN 0921-3449. .

Zati Aqmar Zaharudin, Andrew Brint, and Andrea Genovese. A multi-period model for reorganising urban household waste recycling networks. *Socio-Economic Planning Sciences*, 84:101396, December 2022. ISSN 0038-0121. .

M. Hosein Zare, Juan S. Borrero, Bo Zeng, and Oleg A. Prokopyev. A note on linearized reformulations for a class of bilevel linear integer problems. *Annals of Operations Research*, 272(1–2):99–117, November 2017. ISSN 1572-9338. .

Teng Zhang, Arthur Szlam, and Gilad Lerman. Median k-flats for hybrid linear modeling with many outliers. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 234–241. IEEE, September 2009. .