Reformulation Techniques and Solution Approaches for Fractional 0-1 Programs and Applications

by

Erfan Mehmanchi

B.S. in Industrial Engineering, University of Tehran, 2011

M.S. in Industrial Engineering, Sharif University of Technology, 2013

Submitted to the Graduate Faculty of

the Swanson School of Engineering in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2019

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This dissertation was presented

by

Erfan Mehmanchi

It was defended on

September 13, 2019

and approved by

Oleg A. Prokopyev, Ph.D., Professor, Department of Industrial Engineering

Andrés Gómez, Ph.D., Assistant Professor, Department of Industrial & Systems Engineering, University of Southern California

Panos M. Pardalos, Ph.D., Distinguished Professor, Department of Industrial & Systems Engineering, University of Florida

Jayant Rajgopal, Ph.D., Professor, Department of Industrial Engineering

Bo Zeng, Ph.D., Associate Professor, Department of Industrial Engineering

Dissertation Director: Oleg A. Prokopyev, Ph.D., Professor, Department of Industrial Engineering

Reformulation Techniques and Solution Approaches for Fractional 0-1 Programs and Applications

Erfan Mehmanchi, PhD

University of Pittsburgh, 2019

Fractional binary programs (FPs) form a broad class of nonlinear integer optimization problems, where the objective is to optimize the sum of ratios of (linear) binary functions. FPs arise naturally in a number of important real-life applications such as scheduling, retail assortment, facility location, stochastic service systems, and machine learning, among others.

This dissertation studies methods that improve the performance of solution approaches for fractional binary programs in their general structure. In particular, we first explore the links between equivalent mixed-integer linear programming (MILP) and conic quadratic programming reformulations of FPs. Thereby, we show that integrating the ideas behind these two types of reformulations of FPs allows us to push further the limits of the current state-of-the-art results and tackle larger-size problems.

In practice, the parameters of an optimization problem are often subject to uncertainty. To deal with uncertainties in FPs, we extend the robust methodology to fractional binary programming. In particular, we study robust fractional binary programs (RFPs) under a wide-range of disjoint and joint uncertainty sets, where the former implies separate uncertainty sets for each numerator and denominator, and the latter accounts for different forms of inter-relatedness between them. We demonstrate that, unlike the deterministic case, single-ratio RFP is NP-hard under general polyhedral uncertainty sets. However, if the uncertainty sets are imbued with a certain structure - variants of the well-known budgeted uncertainty - the disjoint and joint single-ratio RFPs are polynomially-solvable when the deterministic counterpart is. We also propose MILP formulations for multiple-ratio RFPs and evaluate their performances by using real and synthetic data sets.

One interesting application of FPs arises in feature selection which is an essential pre-processing step for many machine learning and pattern recognition systems and involves identification of the most characterizing features from the data. Notably, correlation-based and mutual-information-based feature selection problems can be reformulated as single-ratio FPs. We study approaches that ensure globally optimal solutions for medium- and reasonably large-sized instances of the aforementioned problems, where the existing MILPs in the literature fail. We perform computational experiments with diverse classes of real data sets and report encouraging results.

Table of Contents

List of Tables

List of Figures

Preface

First and foremost, I would like to thank my advisor Dr. Oleg Prokopyev. I deeply appreciate his advice, ideas, and motivation throughout my PhD study. In particular, this work would not had been possible without his support and guidance.

Special thanks to Dr. Andrés Gómez for his invaluable insights. His contributions were fundamental for the successful completion of this dissertation.

I am grateful to the committee members Dr. Panos Pardalos, Dr. Jayant Rajgopal, and Dr. Bo Zeng for their suggestions, their time, and their encouragement.

I also want to thank my graduate fellow Colin Gillen for his contribution to the third chapter of this dissertation.

Above all, I would like to thank my family. I am forever indebted to them for everything they have done for me.

1.0 Introduction

Fractional 0-1 programs (FPs), also referred to as hyperbolic 0-1 programs [16, 43, 92], form a broad class of nonlinear integer optimization problems and involve minimization (maximization) of the sum of ratios of (linear) binary functions. Formally, FP is defined as

$$\text{(FP)} \qquad \min_{x \in X} \sum_{i \in I} \frac{a_{i0} + \sum_{j \in J} a_{ij} x_j}{b_{i0} + \sum_{j \in J} b_{ij} x_j},$$

where $I = \{1, ..., m\}$, $J = \{1, ..., n\}$ and $X \subseteq \mathbb{B}^n$ for $\mathbb{B} := \{0, 1\}$. If $m = 1$, then the problem is referred to as single-ratio, else it is multiple-ratio.

FPs have been the subject of many studies since they arise naturally in many practical contexts that involve optimization of efficiency measures (e.g., maximizing the ratio of return/investment or profit/time, see [17, 75, 84, 88]), probabilities, averages, and percentages, among others. Hence, fractional optimization models can be found in diverse application areas including but not limited to problems in data mining and machine learning (such as feature selection [37, 67, 68, 69] and biclustering [22, 93]), scheduling [83], retail assortment [28, 63, 89], set covering [3, 4], facility location [92], stochastic service systems [33, 42], finding diverse solutions to binary-linear programs [94], medical science [10], and so on. We refer the reader to a recent survey in [17] and the references therein for an overview of applications and solution methods for FPs.

1.1 Literature review

Constrained versions of either single or multiple-ratio FPs are NP-hard since linear binary programming that is known to be NP-hard [66] can be viewed as a special case of FP. The constrained (over feasible set X) single-ratio FP with a strictly positive denominator can be solved to optimality by repeatedly solving a sequence of optimization problems with a linear objective function over X via parametric algorithms, such as Newton's method [31] and binary-search [2, 53, 79]. Moreover, if solving such a linear optimization problem over X can be done in polynomial time, then single-ratio FP can be solved in polynomial time. Furthermore, Megiddo [60] shows that if a binary-linear problem admits a polynomial-time algorithm, then so does single-ratio FP. Nevertheless, the unconstrained multiple-ratio FP is NP-hard even for two ratios (or a ratio and a linear function) and strictly positive denominators, see, e.g., [44, 76, 77].

With respect to solution methodologies, typical approaches for solving single-ratio FP are centered around the parametric algorithms. A detailed discussion on these methods is provided in [79]. Additionally, specialized techniques have been proposed for special cases of single-ratio FP, including the minimum fractional spanning tree problem [95], the minimum cost-to-time cycle problem [27], the maximum mean-cut problem [59], the minimum fractional assignment problem [87], and the maximum clique ratio problem [65, 86].

These approaches do not naturally extend for multiple-ratio cases. Typical solution methods in the literature for solving multiple-ratio FPs are based on their reformulations as equivalent *mixed-integer linear programs* (MILPs). An early MILP formulation was given by [99] and later generalized by [92]. A different formulation was suggested by [54], and further discussed by [100] and [92]. Additionally, the work by [92] presents six other formulations. These MILPs mainly rely on the linearization of bilinear (product of a binary and a continuous variables) terms by introducing additional $O(nm)$ continuous variables and big-M constraints. Although the MILP formulations are commonly used, they do not handle well large-scale multiple-ratio FPs, see, e.g., [19, 35, 63], due in part to the weak relaxations caused by the big-M constraints, and also due to the large number of newly added variables and constraints.

Borrero et al. [16] recently proposed an alternative MILP reformulation based on performing *binary expansions* of certain integer-valued expressions. The formulation can substantially reduce the number of bilinear terms that require linearization, thus requiring much fewer variables and constraints than the original MILP formulations. As a consequence, the binarized formulation scales better to large instances; however, binary expansion also leads to weaker continuous relaxations, which in turn can hurt performance in branch-and-bound.

To deal with the weaknesses of MILPs, recently Şen et al. [85] proposed a *mixed-integer conic quadratic programming* (MICQP) reformulation for assortment optimization. Additionally, Atamtürk and Gómez [6] proposed another MICQP reformulation for FPs by explicitly involving submodular functions, and used extended polymatroid cuts [7, 57] to exploit the submodular structure and strengthen the formulations. Both the aforementioned MICQPs result in stronger convex relaxations than the standard MILP counterparts, as the latter require linearization of bilinear terms with big-M constraints.

Additionally, thanks to recent advances in commercial MICQP optimization softwares such as CPLEX [47] and Gurobi [39], conic based reformulations of FPs for small- and medium-sized problems can be solved with a better running time performance in comparison to standard MILP reformulations. However, the solvers still struggle with large-scale mixed-integer nonlinear optimization problems, and hence the performance of the MICQP reformulations degrades considerably in larger instances. Therefore, the researchers and practitioners are often forced to use either heuristic methods or resort to various modeling simplifications that substantially limit the quality of the obtained solutions as the resulting models do not adequately reflect the underlying fractional measures.

Furthermore, in many of the applications listed above, the parameters of optimization problems are often subject to uncertainty. The robust optimization paradigm is a natural approach for addressing such issues [9, 14]. Continuous robust fractional convex optimization is reasonably well studied in the literature, see, e.g., [38, 48, 49]. However, the literature on robust fractional 0-1 programs, in their general form, is rather sparse and it has been studied only for some classes of problems. For example, the work of [81] studies a single-ratio assortment optimization problem under the multinomial logit choice model, where only customer preferences are uncertain. Nevertheless, their results cannot be directly extended for more general classes of fractional problems including the cases when the revenues are subject to uncertainty or the choice model is mixed-multinomial logit.

1.2 Contributions and the structure of the dissertation

The main goal of this dissertation is to address the aforementioned shortcomings in the relevant literature. Our contribution is threefold. First, we improve solution methods for solving generally structured FPs with special focus on reasonably large-sized problems. Second, we propose solution approaches for solving FPs subject to uncertainty. Third, we study FPs in the application setting of feature selection problem.

To this end, Chapter 2 focuses on methods that potentially can improve the efficiency of solution approaches to solve multiple-ratio FPs. Our solution approaches do not completely rely on either mixed-integer linear or conic quadratic programming techniques, but a combination of both. In particular, we first explore the links between MILP- and MICQP-based equivalent reformulations of FPs. Then we enhance the best well-known MILP reformulations, see [54, 99], by exploiting the conic programming techniques. Alternatively, two MICQP reformulations of FP, see [6, 85], are further strengthened and improved via employing mixed-integer programming techniques. We show that combining the ideas behind these reformulations allows us to push further the limits of the current state-of-the-art results in the area and solve problems of larger sizes to optimality.

Chapter 3 is concerned with FPs under uncertainty. The aim is to extend the robust optimization methodology to fractional 0-1 programming in its general structure and to develop a modeling framework for solving robust fractional binary programs (RFPs) under various uncertainty sets. To this end, by understanding the theoretical properties of the models, and combining the ideas from deterministic FP and linear robust optimization new algorithms and reformulations are developed to solve RFPs exactly. Specifically, we consider both single- and multiple-ratio RFPs under various disjoint and joint uncertainty sets, where the former implies separate uncertainty sets for each numerator and denominator, and the latter accounts for different forms of inter-relatedness between them. Then it is demonstrated that single-ratio RFP, contrary to its deterministic counterpart, is NP-hard for a general polyhedral uncertainty set. However, if the uncertainty sets are modeled as a variant of the well-known budgeted uncertainty, then the disjoint and joint single-ratio

RFPs are polynomially-solvable when the deterministic counterpart is. Additionally, MILP reformulations are proposed for solving multiple-ratio RFPs.

Finally, Chapter 4 examines FPs in the context of feature selection, a fundamental problem in data mining and machine learning tasks, which is defined as the problem of selecting a small subset of relevant features to include in a statistical model. Feature selection is also critical for minimizing the classification errors [73] and forms an important class of data mining problems [56]. In particular, some feature selection optimization problems such as correlation feature-selection and minimal-redundancy-maximal-relevance can be modeled in the form of single-ratio (polynomial) fractional 0-1 programs, see [67, 68]. However, solving these problems is challenging for high-dimensional data sets. Thus, non-exact solution methods are usually applied [56, 64, 73]. The goal of Chapter 4 is to exploit the FPs' solution methods for the aforementioned classes of the feature selection problems in order to find more efficient solution approaches that can handle medium- and large-sized data sets.

2.0 Fractional 0-1 Programs: Links between Mixed-integer Linear and Conic Quadratic Formulations

2.1 Introduction

Recall the generally structured fractional binary programs (FPs) introduced in Chapter 1. In addition to the assumption that FP is in minimization form, we also assume that all data are non-negative integers, i.e., $a_{i0}, a_{ij}, b_{i0}, b_{ij} \in \mathbb{Z}_+$ for all $i \in I, j \in J$. Both assumptions are without loss of generality provided that the weaker (and commonly used) assumption $b_{i0} + \sum_{j \in J} b_{ij} x_j > 0$ for all $i \in I$ and $x \in \mathbb{B}^n$ holds, see Appendix A.1 for a discussion.

Contributions and the structure of the chapter. The main goal in this chapter is to develop formulations for generally structured fractional 0-1 programs that perform well for all instance sizes, with special focus on large instances where current methods fail. Specifically, our contribution is threefold:

(i) We perform a comprehensive review of MILP and MICQP formulations of FPs given in the literature and explore the relationships between them.

(ii) We show how to integrate MICQP and MILP formulations to obtain novel formulations that simultaneously have strong convex relaxations, and a limited number of variables and constraints.

(iii) By means of computational experiments, we demonstrate that the proposed formulations outperform existing alternatives formulations.

In order to achieve (i), in Section 2.2 we study the links between the classical MILP formulations LF and LEF, originally proposed in [99] and [54, 100], respectively; the binary-expansion MILP formulation LF_{log} developed in [16]; the MICQP formulations CF and CEF given in [6] and [85], respectively, as well as the MICQP formulation strengthened using polymatroids CF^{P}, also given in [6].

In order to attain (ii), in Section 2.3 we show how to use binary expansions (emanated from MILPs) in MICQP formulations; and how to use conic strengthening (originally proposed in the context of CEF) and polymatroid cuts (originated from CF^P) to strengthen the formulations. More importantly, we show how to incorporate binary expansions and polymatroid strengthening in a single (either MILP or MICQP) formulation. Figure 1 shows the schematic representation of these ideas.

To achieve (*iii*), in Section 2.4, we conduct extensive computational results by using benchmark test instances and observe that the incorporation of improvements leads to formulations that perform better than the existing formulations in the literature.

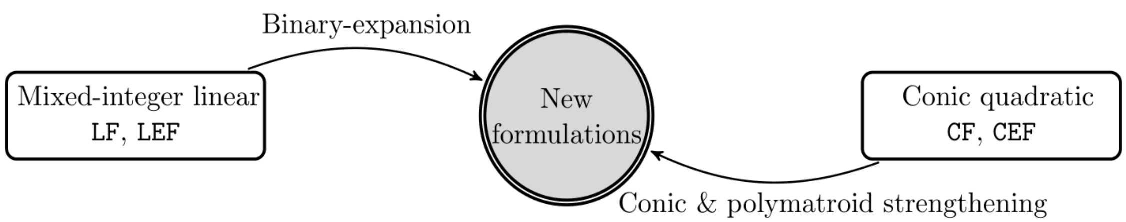


Figure 1: Schematic representation of the ideas in Chapter 2. We exploit binary-expansion technique (from MILP) and conic and polymatroid strengthening (from MICQP) to develop enhanced formulations for FPs.

In addition to the aforementioned formulations for FPs, several new formulations are developed in this chapter. We use the following naming conventions: names starting with "L" correspond to linear formulations, while names starting with "C" correspond to conic quadratic formulations; the letter "F" following the first letter indicates a compact formulation while the letters "EF" following the first letter indicate an *extended formulation*, i.e., a (usually stronger) formulation with additional variables and/or constraints; the subscript "log" indicates a formulation using binary expansions; finally, the superscript "P" indicates a strengthened formulation using polymatroid cuts. Table 1 provides a short summary of all formulations discussed in this chapter, and Figure 2 depicts the relationships between the convex relaxations of the formulations.

Table 1: Formulations studied in this chapter. No citation is given for new formulations. The symbols "+" and "⋆" denote that the corresponding formulation has a superior performance in medium- and large-size instances of our computations, respectively.

Formulation	Version	Linear-based				Conic			
		Without cut		With cut		Without cut		With cut	
Compact	Basic	LF	[99]	LF^P		CF	[6]	CF^P	(+) [6]
	Binary expansion	LF_{log}	[16]	$LF^P_{log}(\star)$		-		-	
Extended	Basic	LEF	[54]	LEF^P (+)		CEF (+)	[85]	CEF^P	
	Binary expansion	LEF_{log}	[16]	LEF^P_{log}		CEF_{log}		CEF^P_{log} $(\star)$	

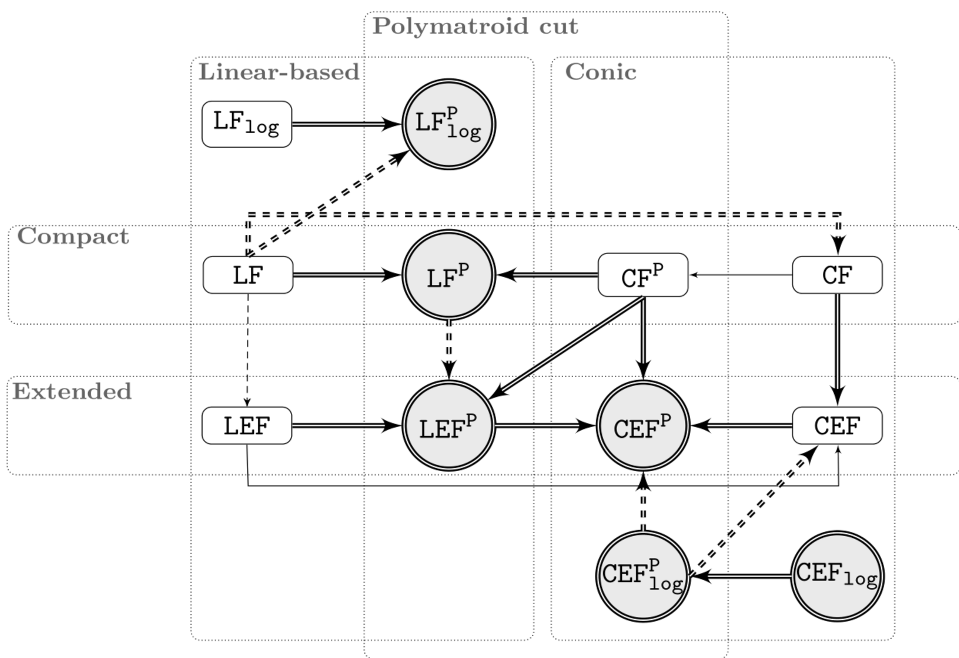


Figure 2: Relationships between the strengths of the convex relaxations of the formulations studied in Chapter 2. Single rectangular frames and single lines indicate existing formulations and shown relations in the literature, respectively. Double circle frames indicate new formulations, and double lines indicate relations shown in this chapter. The symbol S1 ⇒ S2 (or S1 → S2) indicates that formulation S2 has a stronger convex relaxation that formulation S1; this type of relations are demonstrated analytically in Section 2.2 and Section 2.3. Additionally, the symbol S1 ==⇒ S2 (or S1 ⇢ S2) indicates that S2 resulted in smaller root gaps than S1 in most of our computations; this type of relations are shown experimentally by performing computational results in Section 2.4.

2.2 Problem formulations

Herein, we review the MICQP and the (best-known) MILP reformulations of FPs existing in the literature, and describe their interrelatedness. Toward this goal, following our naming convention, in Section 2.2.1 we consider the compact formulations LF, CF and the strengthened version of CF with polymatroid cuts, i.e., CF^{P}. Then in Section 2.2.2 we discuss the extended formulations LEF and CEF involving more variables and/or constraints than LF and CF, respectively. Finally, in Section 2.2.3 we study the binary-expansion reformulations of MILPs.

2.2.1 Compact formulations

For $i \in I$ let

$$t_i := \frac{a_{i0} + \sum_{j\in J} a_{ij}x_j}{b_{i0} + \sum_{j\in J} b_{ij}x_j}. \tag{2.1}$$

Then the substitution of variable t_i for all $i \in I$ in FP yields

$$\min_{x\in X, t\geqslant 0} \quad \sum_{i\in I} t_i \tag{2.2a}$$

$$\text{s.t.} \quad b_{i0}t_i + \sum_{j\in J} b_{ij}x_jt_i \geqslant a_{i0} + \sum_{j\in J} a_{ij}x_j \qquad \forall i \in I. \tag{2.2b}$$

in which (2.2b) holds at equality at any optimal solution. Observe that constraint (2.2b) is nonlinear and non-convex (for $x \in [0,1]^n$) due to the presence of bilinear terms x_jt_i. In the following, we take two convexification procedures. The first uses a concave over-estimator of the left-hand side of inequality (2.2b), resulting in a MILP; see Section 2.2.1.1. The second uses a convex underestimator of the right-hand side of inequality (2.2b) chosen to ensure convexity of the ensuing constraint, resulting in a MICQP; see Section 2.2.1.2.

2.2.1.1 Compact MILP formulation (LF) The first approach is based on the linearization of x_jt_i, which can be accomplished by including additional variables and linear constraints [1, 92, 100]. Specifically, the concave envelope of x_jt_i, where $x_j \in \mathbb{B}$ and t_i is bounded, can be described with the bound constraints and the linear constraints $z_{ij} \leqslant t_i^U x_j$

and $z_{ij} \leqslant t_i + t_i^L(x_j - 1)$, where z_{ij} is a variable representing the hypograph of the bilinear term, and t_i^U and t_i^L are an upper bound and a lower bound on t_i, respectively. Note that under the data non-negativity assumption (see Appendix A.1) the presence of the concave envelope of $x_j t_i$ is sufficient for this linearization. Thus, problem FP can be formulated as the MILP [92, 99]:

$$\text{(LF)} \qquad \min \sum_{i \in I} t_i \tag{2.3a}$$

$$\text{s.t. } b_{i0} t_i + \sum_{j \in J} b_{ij} z_{ij} = a_{i0} + \sum_{j \in J} a_{ij} x_j \qquad \forall i \in I \tag{2.3b}$$

$$z_{ij} \leqslant t_i^U x_j, \; z_{ij} \leqslant t_i + t_i^L (x_j - 1) \qquad \forall i \in I, j \in J \tag{2.3c}$$

$$x \in X, \; t, z \geqslant 0. \tag{2.3d}$$

Formulation LF exploits the integrality restriction on x ($x \in \mathbb{B}^n$) to construct the concave overestimator of the left-hand side of (2.2b), but may be weak due to the used big-M constraints (2.3c). Classical big-M values used are $t_i^U = (a_{i0} + \sum_{j \in J} a_{ij})/b_{i0}$ and $t_i^L = a_{i0} / \left(b_{i0} + \sum_{j \in J} b_{ij}\right)$. Thus, LF is especially weak if either the entries a_{ij} and b_{ij} or the number of variables (n) are large.

2.2.1.2 Compact MICQP formulations (CF and CF$^{\mathbf{P}}$) An alternative approach to resolve the non-convexity of (2.2b) is using conic quadratic programming. For each $i \in I$, we define

$$r_i = b_{i0} + \sum_{j \in J} b_{ij} x_j, \tag{2.4}$$

and $R_i = \left\{ x \in \{0,1\}^n, (r_i, t_i) \in \mathbb{R}_+^2 \mid t_i r_i \geqslant a_{i0} + \sum_{j \in J} a_{ij} x_j \right\}$. Thus, problem (2.2) is equivalent to $\min_{x \in X, t, r \geqslant 0} \left\{ \sum_{i \in I} t_i \mid (2.4) \text{ and } (x, r_i, t_i) \in R_i, \; \forall i \in I \right\}$, that is still non-convex due to R_i.

A simple convex relaxation of R_i can be obtained by squaring the binary variables (and relaxing the integrality constraints), i.e., constraint (2.2b) can be written as $t_i r_i \geqslant a_{i0} + \sum_{j \in J} a_{ij} x_j = a_{i0} + \sum_{j \in J} a_{ij} x_j^2$, where the equality holds for $x_j \in \mathbb{B}$. Thus, problem (2.2) can be posed as the MICQP [6]:

$$\text{(CF)} \quad \min_{\substack{x \in X,\\ t,r \geqslant 0}} \quad \sum_{i \in I} t_i \tag{2.5a}$$

$$\text{s.t.} \quad t_i r_i \geqslant a_{i0} + \sum_{j \in J} a_{ij} x_j^2 \qquad \forall i \in I \tag{2.5b}$$

$$r_i = b_{i0} + \sum_{j \in J} b_{ij} x_j \qquad \forall i \in I. \tag{2.5c}$$

The nonlinear constraint (2.5b) is a rotated cone constraint, which can be directly used with off-the-shelf solvers for MICQP. Observe that, unlike LF, formulation CF does not involve big-M constraints. On the other hand, since $x_j^2 \leqslant x_j$ for $x_j \in [0,1]$, we see that squaring the variables may also lead to a weak relaxation. In fact, formulation CF only uses the upper bounds on x to construct the relaxation, but does not exploit the integrality constraints to derive stronger formulations.

A better convex relaxation of R_i can be obtained by using the strongest convex relaxation of R_i, i.e., $conv(R_i)$, see [6]:

$$(\text{CF}^{\text{P}}) \quad \min_{\substack{x \in X,\\ t,r \geqslant 0}} \quad \sum_{i \in I} t_i$$

$$\text{s.t.} \quad (x, r_i, t_i) \in conv(R_i) \qquad \forall i \in I$$

$$r_i = b_{i0} + \sum_{j \in J} b_{ij} x_j \qquad \forall i \in I.$$

Obviously, CF^{P} has a tighter convex relaxation than CF. However, formulation CF^{P} is much larger than CF, as it requires a factorial number of constraints to construct $conv(R_i)$. Specifically, let Σ denote the set of all permutations for set $\{1, \ldots, n\}$. For a given permutation $\sigma \coloneqq (\sigma(1), \ldots, \sigma(n)) \in \Sigma$, $i \in I$ and $j \in J$, define

$$\pi_{i,\sigma(j)} = \sqrt{\sum_{k=0}^{j} a_{i,\sigma(k)}} - \sqrt{\sum_{k=0}^{j-1} a_{i,\sigma(k)}},$$

where $a_{i,\sigma(0)} = a_{i0}$, and consider the *nonlinear extended polymatroid inequalities*

$$t_i r_i \geqslant \left(\sqrt{a_{i0}} + \sum_{j=1}^{n} \pi_{i,\sigma(j)} x_{\sigma(j)} \right)^2 \qquad \forall \sigma \in \Sigma, i \in I. \tag{2.6}$$

Proposition 1 ([6]). *The extended polymatroid inequalities and bound constraints describe* $conv(R_i)$*, i.e.,* $conv(R_i) = \left\{x \in [0,1]^n, (r_i, t_i) \in \mathbb{R}_+^2 \mid (2.6)\right\}$.

Remark 1. In order to avoid adding all $m \cdot (n!)$ constraints of the form (2.6), Atamtürk and Gómez [6] add constraint (2.5b) – which is redundant for CF$^{\text{P}}$– to the formulation, and add a small number of constraints (2.6) in a cutting surface fashion. The separation of such constraints can be done in $O(n \log n)$ using the greedy algorithm for optimization over polymatroids [32]. □

Remark 2. Inequalities (2.6) can be implemented in a lifted formulation using a single three-dimensional rotated cone inequality and $n!$ linear inequalities – which can be added as cutting planes. Specifically, $(x, r_i, t_i) \in conv(R_i)$ if and only if there exists $s_i \in \mathbb{R}_+$ such that

$$t_i r_i \geqslant s_i^2, \text{ and } \sqrt{a_{i0}} + \sum_{j=1}^{n} \pi_{i,\sigma(j)} x_{\sigma(j)} \leqslant s_i, \; \forall \sigma \in \Sigma.$$

Such a representation is preferable when using current off-the-shelf MICQP solvers, see [6] for further discussions. □

2.2.2 Extended formulations

Unlike compact formulations, which are based on convexifications of either the right-hand side or the left-hand side of (2.2b), extended formulations simultaneously consider both sides of (2.2b). Let

$$y_i := \frac{1}{b_{i0} + \sum_{j \in J} b_{ij} x_j} = \frac{1}{r_i} \qquad \forall i \in I,$$

where r_i is given by (2.4). Then the substitution of variable y_i for all $i \in I$ in FP yields

$$\min_{x \in X, t, y \geqslant 0} \quad \sum_{i \in I} t_i \tag{2.7a}$$

$$\text{s.t.} \quad t_i \geqslant a_{i0} y_i + \sum_{j \in J} a_{ij} x_j y_i \qquad \forall i \in I \tag{2.7b}$$

$$b_{i0} y_i + \sum_{j \in J} b_{ij} x_j y_i \geqslant 1 \qquad \forall i \in I, \tag{2.7c}$$

where t_i is given by (2.1). Both constraints (2.7b) and (2.7c) hold at equality at any optimal solution.

Observe that (2.7b) and (2.7c) use non-convex bilinear terms $x_j y_i$. In order to resolve the non-convexity, we first review LEF, a classical MILP formulation based on formulation (2.7), see Section 2.2.2.1. Then we review the conic quadratic formulation CEF, which is a strengthening of the LEF. Moreover, we demonstrate that CEF is also a strengthening of CF, see Section 2.2.2.2 – in contrast, although LEF has been observed to be stronger than LF in practice, it does not theoretically dominate LF.

2.2.2.1 Extended MILP formulation (LEF) The first approach is based on the linearization of $x_j y_i$. Unlike the approach discussed in Section 2.2.1.1, both the concave and convex envelopes of the bilinear terms need to be constructed, requiring four linear inequalities per term. Letting y_i^U and y_i^L be upper and lower bounds on variable y_i, and letting $\bar{z}_{ij} := x_j y_i$, we find the MILP formulation [54]:

$$
\text{(LEF)} \quad \min \sum_{i \in I} t_i \tag{2.8a}
$$

$$
\text{s.t. } t_i = a_{i0} y_i + \sum_{j \in J} a_{ij} \bar{z}_{ij} \qquad \forall i \in I \tag{2.8b}
$$

$$
b_{i0} y_i + \sum_{j \in J} b_{ij} \bar{z}_{ij} = 1 \qquad \forall i \in I \tag{2.8c}
$$

$$
y_i^L x_j \leqslant \bar{z}_{ij} \leqslant y_i^U x_j \qquad \forall i \in I, j \in J \tag{2.8d}
$$

$$
y_i + y_i^U (x_j - 1) \leqslant \bar{z}_{ij} \leqslant y_i + y_i^L (x_j - 1), \qquad \forall i \in I, j \in J \tag{2.8e}
$$

$$
x \in X, \ t, y, \bar{z} \geqslant 0. \tag{2.8f}
$$

Classical big-M values used are $y_i^U = 1/b_{i0}$ and $y_i^L = 1/(b_{i0} + \sum_{j \in J} b_{ij})$. Thus, LEF is especially weak if either the entries b_{ij} or the number of variables (n) are large (but is not sensitive to the values a_{ij}).

2.2.2.2 Extended MICQP formulation (CEF) Şen et al. [85] recently proposed a conic strengthening of LEF in the context of the assortment problem under multinomial logit choice model, but we show that the strengthening can be used for generally structured fractional binary programs. In particular, since $\bar{z}_{ij} = x_j y_i$ for $x_j \in \mathbb{B}$ and $r_i = 1/y_i$, it follows that the constraint $\bar{z}_{ij} r_i \geqslant x_j$ is valid for LEF; squaring the binary variables, one obtains a convex (rotated cone) constraint that can be used to strengthen the formulations. Moreover, constraint (2.7c) is in fact conic quadratic representable ($y_i r_i \geqslant 1$). Thus, we obtain the formulation:

(CEF)
$$
\begin{aligned}
& \min \sum_{i \in I} t_i && && (2.9a)\\
& \text{s.t. } t_i = a_{i0} y_i + \sum_{j \in J} a_{ij} \bar{z}_{ij} && \forall i \in I && (2.9b)\\
& b_{i0} y_i + \sum_{j \in J} b_{ij} \bar{z}_{ij} = 1 && \forall i \in I && (2.9c)\\
& y_i^L x_j \leqslant \bar{z}_{ij} \leqslant y_i^U x_j && \forall i \in I, j \in J && (2.9d)\\
& y_i + y_i^U (x_j - 1) \leqslant \bar{z}_{ij} \leqslant y_i + y_i^L (x_j - 1), && \forall i \in I, j \in J && (2.9e)\\
& r_i = b_{i0} + \sum_{j \in J} b_{ij} x_j && \forall i \in I && (2.9f)\\
& \bar{z}_{ij} r_i \geqslant x_j^2 && \forall i \in I, j \in J && (2.9g)\\
& y_i r_i \geqslant 1 && \forall i \in I && (2.9h)\\
& x \in X, \ t, y, r, \bar{z} \geqslant 0. && && (2.9i)
\end{aligned}
$$

Formulation CEF generalizes the conic quadratic formulation of [85] - developed for the assortment problem under multinomial logit choice model - for the general fractional binary program FP. Formulation CEF is stronger than LEF as it includes additional constraints. As we now show, formulation CEF is also stronger than CF.

Proposition 2. *The natural convex relaxation of* CEF *is stronger than the relaxation of* CF.

Proof. We start from formulation CF. For each $i \in I$ divide both sides of (2.5b) by $r_i > 0$, leading to the equivalent representation

$$t_i \geqslant \frac{a_{i0}}{r_i} + \sum_{j \in J} a_{ij} \frac{x_j^2}{r_i}.$$

Using the substitutions $y_i \geqslant \frac{1}{r_i}$ and $\bar{z}_{ij} \geqslant \frac{x_j^2}{r_i}$ for all $i \in I, j \in J$ we can write CF as

$$
\begin{aligned}
& \min_{\substack{x_j \in X, \\ t_i, r_i, y_i, \bar{z}_{ij} \geqslant 0}} \quad \sum_{i \in I} t_i && && (2.10a)\\
& \text{s.t.} \quad t_i \geqslant a_{i0} y_i + \sum_{j \in J} a_{ij} \bar{z}_{ij} && \forall i \in I && (2.10b)
\end{aligned}
$$

$$y_i r_i \geqslant 1 \qquad \forall i \in I \quad (2.10c)$$

$$\bar{z}_{ij} r_i \geqslant x_j^2 \qquad \forall i \in I, j \in J \quad (2.10d)$$

$$r_i = b_0 + b_{ij} x_j \qquad \forall i \in I. \quad (2.10e)$$

Observe that none of the transformations discussed exploit the integrality constraints, thus formulation (2.10) above has the same continuous relaxation as CF. If formulation (2.10) is strengthened using constraints (2.9c), (2.9d), and (2.9e), then one obtains precisely CEF, thus proving the proposition. □

Remark 3 (Extended formulation of CF)**.** Formulations CF and (2.10) are equivalent, in the sense that their natural convex relaxations (by relaxing integrality constraints in x) coincide. However, formulation (2.10) requires $m + nm$ additional variables. Moreover, (2.10) has $m + nm$ three-dimensional rotated cone constraints, while formulation CF has m $(n+2)$-dimensional rotated cone constraints. The extended formulation (2.10) is preferable in the context of branch-and-bound, as the corresponding linear outer approximations are stronger, see [97]. In fact, modern conic quadratic branch-and-bound solvers will automatically reformulate CF into a form similar to (2.10) in the presolve process. □

2.2.3 MILP binary-expansion formulation ($\mathbf{LF_{log}}$)

Under the data integrality assumption, the binary-expansion technique attempts to reduce the number of bilinear terms ($x_j t_i$ or $x_j y_i$) that need to be linearized in LF or LEF. Specifically, for the binary-expansion reformulation of LF, let $\theta_i^b := \lfloor \log_2 \left(\sum_{j \in J} b_{ij} \right) \rfloor + 1$, then by using the substitution $\sum_{j \in J} b_{ij} x_j = \sum_{k=1}^{\theta_i^b} 2^{k-1} w_{ik}^b$ in problem (2.2) we get

$$\min \quad \sum_{i \in I} t_i \quad (2.11a)$$

$$\text{s.t.} \quad b_{i0} t_i + \sum_{k=1}^{\theta_i^b} 2^{k-1} w_{ik}^b t_i = a_{i0} + \sum_{j \in J} a_{ij} x_j \qquad \forall i \in I \quad (2.11b)$$

$$\sum_{j \in J} b_{ij} x_j = \sum_{k=1}^{\theta_i^b} 2^{k-1} w_{ik}^b \qquad \forall i \in I \quad (2.11c)$$

$$x \in X, w_{ik}^b \in \mathbb{B}, t_i \geqslant 0 \qquad \forall i \in I, k \in \{1, \ldots, \theta_i^b\}. \quad (2.11d)$$

Observe that, since $x_j \in \mathbb{B}$, the left-hand side of constraint (2.11c) is integer for any feasible solution of (2.11), and thus constraint (2.11c) can always be satisfied at equality. Using a similar linearization as the one described in Section 2.2.1.1 to linearize the product terms $w_{ij}^b t_i$, we obtain the MILP formulation [16]:

$$
\begin{aligned}
(\mathrm{LF_{log}}) \qquad & \min \quad \sum_{i \in I} t_i && && (2.12a)\\
& \text{s.t.} \quad b_{i0} t_i + \sum_{k=1}^{\theta_i^b} 2^{k-1} z_{ik}^b = a_{i0} + \sum_{j \in J} a_{ij} x_j, && \forall i \in I && (2.12b)\\
& \sum_{j \in J} b_{ij} x_j = \sum_{k=1}^{\theta_i^b} 2^{k-1} w_{ik}^b, && \forall i \in I && (2.12c)\\
& z_{ik}^b \leqslant t_i^U w_{ik}^b,\ z_{ik}^b \leqslant t_i + t_i^L (z_{ij}^b - 1) && \forall i \in I, k \in \{1, \ldots, \theta_i^b\} && (2.12d)\\
& x \in X, w_{ik}^b \in \mathbb{B}, z_{ik}^b \geqslant 0, t_i \geqslant 0 && \forall i \in I, k \in \{1, \ldots, \theta_i^b\}. && (2.12e)
\end{aligned}
$$

When $\theta_i^b << n$, which is the case when n is large and the coefficients b_{ij} are small, formulation $\mathrm{LF_{log}}$ requires substantially less (continuous) variables and big-M constraints than LF, but the strength of the continuous relaxation of $\mathrm{LF_{log}}$ is weaker. Nonetheless, by performing computation results, see Section 2.4, we observe that for large instances formulation $\mathrm{LF_{log}}$ results in much more branch-and-bound nodes explored and better performance overall.

Remark 4. It is also possible to develop a binary-expansion reformulation for LEF. However, based on the results in [16, 61] such a formulation performs poorly. Thus, we omit $\mathrm{LEF_{log}}$ from Figure 2 and the discussion in this chapter for the sake of brevity. □

In Example 1 below, we evaluate the formulations discussed in Section 2.2 for a specific instance.

Example 1. Consider unconstrained ($X = \mathbb{B}^n$) two-ratio ($m = 2$) five-variate ($n = 5$) fractional 0-1 program

$$
\min_{x \in \mathbb{B}^5} \left\{ \frac{1 + x_1 + x_2 + 2x_3 + 2x_4 + x_5}{2 + x_1 + x_2 + x_3 + x_4 + x_5} + \frac{2 + 2x_1 + 3x_2 + x_3 + x_4}{1 + 2x_1 + 2x_2 + 3x_3} \right\}, \qquad (2.13)
$$

which has the optimal objective function value 1.75.

(i) The objective function values of convex relaxations, computed by CPLEX 12.7.1 [47], for the basic reformulations of (2.13), i.e., LF, CF, LEF, and CEF are: 0.482, 1.236, 1.484, and 1.639, respectively.

(ii) For permutation $\sigma = (1, 2, 3, 4, 5)$, polymatroid inequalities (2.6) for the first and second ratios are, respectively,

$$t_1 r_1 \geqslant \Big(1 + (\sqrt{2} - 1)x_1 + (\sqrt{3} - \sqrt{2})x_2 + (\sqrt{5} - \sqrt{3})x_3 + (\sqrt{7} - \sqrt{5})x_4 + (\sqrt{8} - \sqrt{7})x_5\Big)^2, \text{ and} \tag{2.14a}$$

$$t_2 r_2 \geqslant \Big(2 + (\sqrt{4} - \sqrt{2})x_1 + (\sqrt{7} - \sqrt{4})x_2 + (\sqrt{8} - \sqrt{7})x_3 + (\sqrt{9} - \sqrt{8})x_4 + 0x_5\Big)^2. \tag{2.14b}$$

If we add (2.14a) and (2.14b) to CF (without (2.5b)), then the objective function value of the convex relaxation of the resulting formulation is improved to 1.349. Additionally, if inequalities (2.6) for all 5! and 4! permutations of the first and second ratios' numerators indices (in total 144 rotated cone constraints) are added to CF (without (2.5b)), then the resulting formulation is CF^{P} with an improved relaxation objective function value equal to 1.697. Thus, CF^{P} results in the best convex relaxation among the formulations of Section 2.2 in this particular instance.

(iii) By using the binary-expansion technique, constraint (2.2b) in model (2.2) for the first and second ratios, i.e.,

$$2t_1 + (x_1 + x_2 + x_3 + x_4 + x_5)t_1 \geqslant 1 + x_1 + x_2 + 2x_3 + 2x_4 + x_5, \text{ and} \tag{2.15a}$$

$$t_2 + (2x_1 + 2x_2 + 3x_3)t_2 \geqslant 2 + 2x_1 + 3x_2 + x_3 + x_4, \tag{2.15b}$$

can be replaced, respectively, by

$$2t_1 + (2^0 w^b_{11} + 2^1 w^b_{12} + 2^2 w^b_{13})t_1 \geqslant 1 + x_1 + x_2 + 2x_3 + 2x_4 + x_5, \text{ and} \tag{2.16a}$$

$$t_2 + (2^0 w^b_{21} + 2^1 w^b_{22} + 2^2 w^b_{23})t_2 \geqslant 2 + 2x_1 + 3x_2 + x_3 + x_4. \tag{2.16b}$$

Note that instead of linearizing 8 bilinear terms ($x_j t_i$) in the left-hand sides of (2.15a) and (2.15b), which results in LF, only 6 bilinear terms ($w^b_{ik} t_i$) are required to be linearized in the left-hand sides of (2.16a) and (2.16b), which lead to formulation LF_{log}. Recall that fewer linearizations implies fewer number of additional continuous variables and big-M constraints.

However, $\text{LF}_{\log}$ has a weaker convex relaxation objective value than LF (0.405 vs. 0.482). Thus, $\text{LF}_{\log}$ results in the worst convex relaxation in this particular instance, but also in the smallest and easiest to solve convex relaxation. □

2.3 Enhancements

None of the formulations presented in Section 2.2 consistently outperforms the others. MICQP formulations are in general stronger and perform best in small- and medium-size problems; however, due to the difficulties of optimization solvers to handle the nonlinear convex relaxations, they may fail to adequately process the root node in larger instances. In contrast, the binarized MILPs tend to perform better than MICQPs in larger instances thanks to the reduced formulation size and linear convex relaxations; however, they do not perform as well in small instances. Finally, MILP formulations perform somewhat in between the MICQPs and binarized MILPs.

In this section, we aim to further improve the performance of the existing formulations for FPs. First, from the analysis in Section 2.2, it becomes apparent how to "mix" the ideas behind these formulations to improve their performance, see Section 2.3.1. Then, in Section 2.3.2, we develop binary-expansion techniques for conic quadratic formulations. By using the proposed improvements, we obtain strong formulations of moderate sizes, which perform well across all problem sizes and are particularly effective in larger instances.

2.3.1 "Mixing" formulations ($\mathbf{CEF^P}, \mathbf{LF^P}, \mathbf{LEF^P}$, and $\mathbf{LF^P_{log}}$)

Herein, we employ polymatroid cuts in CEF. Then, more interestingly, we make MILP formulations LF, LEF, and $\text{LF}_{\log}$ able to benefit from polymatroid strengthening, as well.

First, note that neither CEF nor CF^P theoretically dominates the other in terms of strength of the continuous relaxations. Moreover, in our computations (see Section 2.4), neither consistently dominates the other. Nonetheless, we can obtain a stronger new formulation simply by adding the nonlinear extended polymatroid inequalities to CEF, i.e.,

$$(\mathrm{CEF^P}):\ \min_{x,y,\bar{z},t,r}\left\{\sum_{i\in I} t_i \mid\ (2.9\mathrm{b})-(2.9\mathrm{i}),\ (x,r_i,t_i)\in conv(R_i)\ \forall i\in I\right\}.$$

Clearly, $\mathrm{CEF^P}$ is stronger than CEF and based on Proposition 2, it is also stronger than $\mathrm{CF^P}$. Indeed, formulation $\mathrm{CEF^P}$ results in the best convex relaxations among the formulations presented in this chapter. However, due to its size, it is impractical in larger instances. We address this issue by using the binary-expansion idea in Section 2.3.2. We also point out that several approaches to strengthen the MILP formulations have been proposed in the literature, see, e.g., [63, 92]. Clearly, such approaches can naturally be used with any of the formulations present in Section 2.2, or the new formulations introduced in this section.

Second, as pointed out in Remark 1, previous implementations of $\mathrm{CF^P}$ also added constraints (2.5b), large-dimensional conic quadratic constraints which substantially increases the computational burden of solving the convex relaxations, despite the recent advances in off-the-shelf optimization solvers. An alternative is to use the nonlinear extended polymatroid constraints with formulation LF, i.e.,

$$(\mathrm{LF^P}):\ \min_{x,y,z,t,r}\left\{\sum_{i\in I} t_i \mid\ (2.3\mathrm{b})-(2.3\mathrm{d}),\ r_i=b_{i0}+\sum_{j\in J} b_{ij}x_j,\ (x,r_i,t_i)\in conv(R_i)\ \forall i\in I\right\}.$$

Clearly, $\mathrm{LF^P}$ dominates both LF and $\mathrm{CF^P}$ in terms of the strength of the convex relaxation (the second domination statement holds only if all inequalities (2.6) are added. Nonetheless, $\mathrm{LF^P}$ is able to achieve excellent convex relaxations with a modest number of cuts.) More importantly, using the extended formulation described in Remark 2, $\mathrm{LF^P}$ requires only m three-dimensional rotated cone constraints, which are much easier to handle than m $(n+2)$-dimensional conic constraints of $\mathrm{CF^P}$. Alternatively, efficient polyhedral outer-approximations of the rotated cone constraint can be easily constructed [8, 96], and $\mathrm{LF^P}$ can be implemented in a pure MILP framework.

Similarly, one can use the nonlinear extended polymatroid constraints with formulations LEF and $\mathrm{LF_{log}}$, yielding

$$(\mathrm{LEF^P}):\min_{x,\bar{z},t,y,r}\left\{\sum_{i\in I} t_i \mid (2.8\mathrm{b})-(2.8\mathrm{f}), r_i=b_{i0}+\sum_{j\in J} b_{ij}x_j, (x,r_i,t_i)\in conv(R_i)\ \forall i\in I\right\},\ \text{and}$$

$$(\mathrm{LF}^{\mathrm{P}}_{\log}): \min_{x,w^b,z^b,t,r} \left\{ \sum_{i\in I} t_i \mid (2.12\mathrm{b}) - (2.12\mathrm{e}), r_i = b_{i0} + \sum_{j\in J} b_{ij}x_j, (x, r_i, t_i) \in conv(R_i) \ \forall i \in I \right\}.$$

Formulation $\mathrm{LEF}^{\mathrm{P}}$ has (in our computations) a stronger convex relaxation than LF^{P} while maintaining easy-to-solve convex relaxations in small and medium instances. Formulation $\mathrm{LF}^{\mathrm{P}}_{\log}$ has a small size, but has weaker convex relaxations than LF^{P} and $\mathrm{LEF}^{\mathrm{P}}$.

By comparing LF^{P} and $\mathrm{LEF}^{\mathrm{P}}$ with CF^{P} we can conclude that the former both are stronger than CF^{P} as depicted in Figure 2. Based on the discussion given in Section 2.2.2.2, we also conclude that $\mathrm{CEF}^{\mathrm{P}}$ is stronger than $\mathrm{LEF}^{\mathrm{P}}$.

Standout formulation. Formulation $\mathrm{LF}^{\mathrm{P}}_{\log}$ is one of the best formulations in our computations. It was observed in [16] (and corroborated in our experiments) that while the continuous relaxation of $\mathrm{LF}_{\log}$ is weaker than LF, which in turn is much weaker than LEF, it may result in better performance due to the faster exploration of the branch-and-bound tree. With the inclusion of the nonlinear polymatroid inequalities, formulation $\mathrm{LF}^{\mathrm{P}}_{\log}$ has a convex relaxation strength similar to CF^{P}, which is substantially stronger than LF and was also observed to be stronger than LEF [6]. Moreover, using $\mathrm{LF}^{\mathrm{P}}_{\log}$ results in small formulations with a few nonlinearities, thus allowing for a much faster exploration of the branch-and-bound tree than CF^{P}, and performing well across all instance sizes. Intuitively, formulation $\mathrm{LF}^{\mathrm{P}}_{\log}$ benefits both from the advantages of the conic formulations (strength) and binarization ideas (speed).

Remark 5. We need to point out that $conv(R_i)$ is implemented in this chapter using rotated cone constraints instead of explicit polyhedral outer approximations. Hence, LF^{P}, $\mathrm{LEF}^{\mathrm{P}}$ and $\mathrm{LF}^{\mathrm{P}}_{\log}$ are in fact MICQPs, see also Remarks 1 and 2; however, in contrast to other MICQPs in this chapter, they involve only a small number of "easy" 3-dimensional rotated cone constraints. □

2.3.2 Enhancements on CEF

Next, we develop a binary-expansion reformulation for the conic quadratic program CEF, which we call $\mathrm{CEF}_{\log}$, see Section 2.3.2.1. Then we extend the notion of polymatroid cuts to the binary-expansion space in order to further strengthen $\mathrm{CEF}_{\log}$, see Section 2.3.2.2.

2.3.2.1 MICQP binary-expansion formulation ($\mathbf{CEF_{log}}$) As pointed out earlier, the MICQP reformulations of FPs do not require the linearization of bilinear terms. Nevertheless, we demonstrate that binarization technique – developed in Section 2.2.3 for MILPs – still can be employed to reduce the number of variables and rotated quadratic cone constraints in CEF as shown below. Let $\theta_i^a := \lfloor \log_2(\sum_{j\in J} a_{ij}) \rfloor + 1$ and, by using the substitution $\sum_{j\in J} a_{ij}x_j = \sum_{k=1}^{\theta_i^a} 2^{k-1} w_{ik}^a$, we can rewrite (2.7) as

$$
\begin{aligned}
\min \quad & \sum_{i\in I} t_i && && (2.17a)\\
\text{s.t.} \quad & t_i \geqslant a_{i0}y_i + \sum_{k=1}^{\theta_i^a} 2^{k-1} w_{ik}^a y_i && \forall i \in I && (2.17b)\\
& \sum_{j\in J} a_{ij}x_j = \sum_{k=1}^{\theta_i^a} 2^{k-1} w_{ik}^a && \forall i \in I && (2.17c)\\
& r_i y_i \geqslant 1 && \forall i \in I && (2.17d)\\
& r_i = b_{i0} + \sum_{j\in J} b_{ij}x_j && \forall i \in I && (2.17e)\\
& x \in X, y_i \geqslant 0, w_{ik}^a \in \mathbb{B} && \forall i \in I, k \in \{1,\ldots,\theta_i^a\}. && (2.17f)
\end{aligned}
$$

Then introducing variables $z_{ik}^a := w_{ik}^a y_i = w_{ik}^a / r_i$ and exploiting the fact that $\left(w_{ik}^a\right)^2 = w_{ik}^a$ for $w_{ik}^a \in \mathbb{B}$, problem (2.17) can be convexified as

$$
\begin{aligned}
\min \quad & \sum_{i\in I} t_i && && (2.18a)\\
\text{s.t.} \quad & t_i \geqslant a_{i0}y_i + \sum_{k=1}^{\theta_i^a} 2^{k-1} z_{ik}^a && \forall i \in I && (2.18b)\\
& z_{ik}^a r_i \geqslant \left(w_{ik}^a\right)^2 && \forall i \in I, k \in \{1,\ldots,\theta_i^a\} && (2.18c)\\
& \sum_{j\in J} a_{ij}x_j = \sum_{k=1}^{\theta_i^a} 2^{k-1} w_{ik}^a && \forall i \in I && (2.18d)\\
& r_i y_i \geqslant 1 && \forall i \in I && (2.18e)\\
& r_i = b_{i0} + \sum_{j\in J} b_{ij}x_j && \forall i \in I && (2.18f)\\
& x \in X, y_i \geqslant 0, w_{ik}^a \in \mathbb{B}, z_{ik}^a \geqslant 0 && \forall i \in I, k \in \{1,\ldots,\theta_i^a\}. && (2.18g)
\end{aligned}
$$

Formulation (2.18) can be further strengthened by adding the linearization constraints $z_{ij}^a \geqslant y_i^L w_{ik}^a$, and $z_{ij}^a \geqslant y_i + y_i^U(w_{ik}^a - 1)$. The resulting conic quadratic binary-expansion reformulation is

$$
\begin{aligned}
(\text{CEF}_{\log}) \qquad \min \quad & \sum_{i\in I} t_i && && (2.19a)\\
\text{s.t.} \quad & t_i \geqslant a_{i0}y_i + \sum_{k=1}^{\theta_i^a} 2^{k-1} z_{ik}^a && \forall i \in I && (2.19b)\\
& r_i = b_{i0} + \sum_{j\in J} b_{ij}x_j && \forall i \in I && (2.19c)\\
& z_{ik}^a r_i \geqslant (w_{ik}^a)^2 && \forall i \in I, k \in \{1,\dots,\theta_i^a\} && (2.19d)\\
& y_i r_i \geqslant 1 && \forall i \in I && (2.19e)\\
& \sum_{j\in J} a_{ij}x_j = \sum_{k=1}^{\theta_i^a} 2^{k-1} w_{ik}^a && \forall i \in I && (2.19f)\\
& z_{ik}^a \geqslant y_i^L w_{ik}^a, z_{ik}^a \geqslant y_i + y_i^U(w_{ik}^a - 1) && \forall i \in I, k \in \{1,\dots,\theta_i^a\} && (2.19g)\\
& w_{ik}^a \in \mathbb{B}, z_{ik}^a \geqslant 0 && \forall i \in I, k \in \{1,\dots,\theta_i^a\} && (2.19h)\\
& x \in X, t, y, r \geqslant 0. && && (2.19i)
\end{aligned}
$$

Formulation $\text{CEF}_{\log}$ requires $m + \sum_{i\in I}\theta_i^a$ rotated cone constraints, which can be significantly less than the $m + mn$ rotated cone constraints required by CEF.

Remark 6. It is also possible to develop binary-expansion reformulations for CF and CF^{P}. However, since these formulations do not include any product term of a binary and a continuous variables, the binary expansion does not allow us to reduce neither the number of their variables nor constraints. Therefore, we have excluded $\text{CF}_{\log}$ and $\text{CF}_{\log}^{\text{P}}$ from Table 1, Figure 2 and the discussion in this chapter. □

2.3.2.2 Polymatroid cuts in the binary-expansion space ($\text{CEF}_{\log}^{\text{P}}$) Formulation $\text{CEF}_{\log}$ can be further strengthened by using submodularity. Specifically, observe that by multiplying constraint (2.17b) by r_i and exploiting that $y_i r_i = 1$ in optimal solutions of (2.17), we find that the constraints $(w_i^a, r_i, t_i) \in R_i^{\log}$ can be added, where

$$R_i^{\log} = \left\{ w_i^a \in \mathbb{B}^{\theta_i^a}, (r_i, t_i) \in \mathbb{R}_+^2 \mid t_i r_i \geqslant a_{i0} + \sum_{k=1}^{\theta_i^a} 2^{k-1}(w_{ik}^a)^2 \right\}.$$

An ideal formulation of $R_i^{\log}$ can be found using polymatroids, similarly to the approach in Section 2.2.1.2, i.e.,

$$conv(R_i^{\log}) = \left\{(w_i^a, r_i, t_i) \in [0,1]^{\theta_i^a} \times \mathbb{R}_+^2 \mid t_i r_i \geqslant (\sqrt{a_{i0}} + \lambda_i' w_i^a)^2, \ \forall \lambda_i \in \Lambda_i\right\}, \tag{2.20}$$

where

$$\Lambda_i = \Big\{\lambda_i \in \mathbb{R}_+^{\theta_i^a} \mid \lambda_{i,\sigma(k)} = \sqrt{\gamma_{i,\sigma(k)}} - \sqrt{\gamma_{i,\sigma(k-1)}}, \ \text{ where } \gamma_{i,\sigma(k)} = 2^{\sigma(k)-1} + \gamma_{i,\sigma(k-1)}$$
$$\text{and } \gamma_{i,\sigma(0)} = a_{i0}, \text{ for all permutations } \sigma \in [\theta_i^a], k \in \{1,\dots,\theta_i^a\}\Big\}.$$

Observe that $\theta_i^a \ll n$ (for all $i \in I$) for large size problems with sufficiently small values for a_{ij}. Consequently, we have $(\theta_i^a)! \ll n!$, for each $i \in I$, and thus, $conv(R_i^{\log})$ can be constructed using significantly fewer polymatroid cuts than $conv(R_i)$. Adding $(w_i^a, r_i, t_i) \in conv(R_i^{\log})$ to $\text{CEF}_{\log}$ allows this binarized formulation to benefit from polymatroid cuts, that is given by

$$(\text{CEF}_{\log}^{\text{P}}): \min_{x,y,z,t,r,w^a} \left\{\sum_{i \in I} t_i \mid \ (2.19\text{b}) - (2.19\text{i}), \ (w_i^a, r_i, t_i) \in conv(R_i^{\log}), \ \forall i \in I\right\}.$$

Standout formulation. Formulation $\text{CEF}_{\log}^{\text{P}}$ is another of the best formulations in our computations. Similarly to $\text{LF}_{\log}^{\text{P}}$, formulation $\text{CEF}_{\log}^{\text{P}}$ is able to strike a good balance between the size and the strength of the convex relaxation by incorporating binary-expansion and polymatroid cuts, resulting in a similar performance as CEF in small instances, but scales much better to larger problems.

Example 1 (continued). Next, we evaluate the reformulations of (2.13) for the models proposed in Section 2.3.

(iv) In order to take the advantage of polymatroid strengthening, we add to LF, $\text{LF}_{\log}$, LEF constraints of the form (2.4), i.e., $r_1 = 2 + x_1 + x_2 + x_3 + x_4 + x_5$ and $r_2 = 1 + 2x_1 + 2x_2 + 3x_3$. Additionally, we add 144 rotated cone constraints of the form (2.6) to the aforementioned formulations and CEF. Then we obtain LF^{P}, $\text{LF}_{\log}^{\text{P}}$, LEF^{P}, and CEF^{P}, that have improved relaxation objective function values of 1.697 (vs. 0.482 of LF), 1.697 (vs. 0.405 of $\text{LF}_{\log}$), 1.702 (vs. 1.484 of LEF), and 1.702 (vs. 1.639 of CEF), respectively, and close most of the gap to the optimal objective function value 1.75.

(v) By using the binary-expansion technique, constraint (2.7b) in model (2.7) for the first and second ratios, i.e.,

$$t_1 \geqslant y_1 + (x_1 + x_2 + 2x_3 + 2x_4 + x_5)y_1, \text{ and} \tag{2.21a}$$

$$t_2 \geqslant 2y_2 + (2x_1 + 3x_2 + x_3 + x_4)y_2, \tag{2.21b}$$

can be replaced, respectively, by

$$t_1 \geqslant y_1 + (2^0 w_{11}^a + 2^1 w_{12}^a + 2^2 w_{13}^a)y_1, \text{ and} \tag{2.22a}$$

$$t_2 \geqslant 2y_2 + (2^0 w_{21}^a + 2^1 w_{22}^a + 2^2 w_{23}^a)y_2. \tag{2.22b}$$

In order to obtain CEF we need to convexify 9 bilinear terms $x_j y_i$ in the right-hand sides of (2.21a) and (2.21b) as rotated cone constraints $\bar{z}_{ij} r_i \geqslant x_j^2$. In comparison, in order to achieve CEF_{log} only 6 bilinear terms $w_{ik}^a y_i$ in the right-hand sides of (2.22a) and (2.22b) are required to be convexified as $z_{ik}^a r_i \geqslant (w_{ik}^a)^2$. Although CEF_{log} has 3 fewer rotated cone constraints than CEF, it has a worse relaxation objective function value (1.244 vs. 1.639). Next, we improve its relaxation by using polymatroid cuts in the binary-expansion space.

(vi) For permutation $\sigma = (1, 2, 3)$ inequalities $t_i r_i \geqslant (\sqrt{a_{i0}} + \lambda_i' w_i^a)^2$ in (2.20) for the first and second ratios are, respectively,

$$t_1 r_1 \geqslant \left(1 + (\sqrt{2} - 1)w_{11}^a + (\sqrt{4} - \sqrt{2})w_{12}^a + (\sqrt{8} - \sqrt{4})w_{13}^a\right)^2, \text{ and} \tag{2.23a}$$

$$t_2 r_2 \geqslant \left(2 + (\sqrt{3} - \sqrt{2})w_{12}^a + (\sqrt{5} - \sqrt{3})w_{22}^a + (\sqrt{9} - \sqrt{5})w_{32}^a\right)^2. \tag{2.23b}$$

If we add (2.23a) and (2.23b) to CEF_{log}, then its relaxation objective function value from 1.244 is improved to 1.311. If we add all $2 \cdot 3! = 12$ polymatroid inequalities to CEF_{log}, then the resulting formulation is $\text{CEF}_{\text{log}}^{\text{P}}$ with a better relaxation objective function value of 1.446. Note that the number of cuts added to obtain $\text{CEF}_{\text{log}}^{\text{P}}$ is significantly fewer than the number of cuts added in order to obtain any of the other formulations strengthened with polymatroid cuts (12 vs. 144 cuts).

Therefore, from this example, we observe that there is a trade-off between using polymatroid cuts and binarization. The former improves the relaxation objective function value at the expense of a larger problem, and the latter reduces the number of (continuous) variables and (either linear or rotated cone) constraints at the cost of a weaker relaxation. However, the incorporation of these ideas leads to moderate size formulations, i.e., $\text{CEF}^{\text{P}}_{\text{log}}$ and $\text{LF}^{\text{P}}_{\text{log}}$, that benefit from strong convex relaxations. □

2.3.3 Problems sizes

Table 2 shows the number of continuous and binary variables as well as the number of linear and rotated cone constraints for MILP and MICQP formulations discussed in Sections 2.2 and 2.3. By comparing each binarized formulation with the corresponding basic formulation, it is seen that the binary-expansion technique can potentially decrease the number of continuous variables and also the number of linear/rotated cone constraints – especially for large values of n – with a moderate increase in the number of binary variables. We also observe that adjusting the formulations to enable them to use polymatroid cuts only slightly increases the number of variables or constraints.

2.4 Computational results

We perform extensive computational experiments to evaluate the performances of the currently existing formulations in the literature presented in Section 2.2 and to compare them versus the enhancements developed in Section 2.3. We outline the structure and parameters of the computational experiments in Section 2.4.1. We discuss the obtained results in Sections 2.4.2 and 2.4.3 and Appendix A.2.

2.4.1 Computational environment and test instances

All of the computational instances are solved using CPLEX 12.7.1 [47] on a 32-core CPU (2.90GHz) with 160 GB of RAM; we allocate a single thread and 8 GB of RAM for each individual experiment, and use a time limit of one hour (3600 seconds). To avoid running-out-of-memory difficulties we use the "node-file storage-feature" of CPLEX to store some parts of the branch-and-cut tree on disk when the size of the tree exceeds the allocated memory. The polymatroid inequalities are added at the root node by using callback functions of CPLEX as described in Remarks 1 and 2.

Table 2: The reformulation sizes (number of variables and constraints), where n and m are defined as in FP, q is the number of constraints defining X, $\theta_i^a = \lfloor \log_2(\sum_{j\in J} a_{ij}) \rfloor + 1$ and $\theta_i^b = \lfloor \log_2(\sum_{j\in J} b_{ij}) \rfloor + 1$. Subscript "log" and superscript "P" are reserved for binary-expansion and polymatroid cuts, respectively.

	Variables		Constraints	
Formulation	Continuous	Binary	Linear	Rotated cone
MILP-based reformulations				
LF	$m(n+1)$	n	$m(2n+1)+q$	-
LF^{P}	$m(n+2)$	n	$m(2n+2)+q+\text{cuts}^*$	m
$\text{LF}_{\log}$	$m+\sum_{i\in I}\theta_i^b$	$n+\sum_{i\in I}\theta_i^b$	$2m+2\sum_{i\in I}\theta_i^b+q$	-
$\text{LF}_{\log}^{\text{P}}$	$2m+\sum_{i\in I}\theta_i^b$	$n+\sum_{i\in I}\theta_i^b$	$3m+2\sum_{i\in I}\theta_i^b+q+\text{cuts}$	m
LEF	$m(n+2)$	n	$m(4n+2)+q$	-
LEF^{P}	$m(n+3)$	n	$m(4n+3)+q+\text{cuts}$	m
MICQP reformulations				
CF	$m(n+3)$	n	$2m+q$	$m(n+1)^{**}$
CF^{P}	$m(n+3)$	n	$2m+q+\text{cuts}$	$m(n+2)$
CEF	$m(n+3)$	n	$m(4n+3)+q$	$m(n+1)$
CEF^{P}	$m(n+3)$	n	$m(4n+3)+q+\text{cuts}$	$m(n+2)$
$\text{CEF}_{\log}$	$3m+\sum_{i\in I}\theta_i^a$	$n+\sum_{i\in I}\theta_i^a$	$3m+2\sum_{i\in I}\theta_i^a+q$	$m+\sum_{i\in I}\theta_i^a$
$\text{CEF}_{\log}^{\text{P}}$	$3m+\sum_{i\in I}\theta_i^a$	$n+\sum_{i\in I}\theta_i^a$	$3m+2\sum_{i\in I}\theta_i^a+q+\text{cuts}$	$2m+\sum_{i\in I}\theta_i^a$

*Polymatroid cuts are added on the fly, implemented as discussed in Remark 2.

**Formulations CF and CF^{P} are based on extended formulation (2.10).

Test instances. We consider three classes of instances: "small" ($n \in \{25, 50, 100\}$) and "medium" ($n \in \{200, 500, 1000\}$) size instances with $m = \lfloor 10\% \cdot n \rfloor$, and "large" size instances ($n \in \{2000, 5000,\ 10000\}$) with $m = 100$. For each choice of n and each of the following data generation settings five instances are sampled and the results are averaged.

• *Assortment data set.* For the first setting, we consider the assortment optimization problems that naturally arise in many applications such as online advertising, retailing, and revenue management [80]. Under the mixed multinomial logit model (see, e.g., [63, 85, 89]) we are given $I = \{1, 2, \ldots, m\}$ classes of customers and $J = \{1, 2, \ldots, n\}$ available products. Then the assortment optimization problem is defined as the problem of deciding which assortment of products $S \subseteq J$ must be offered to customers in order to maximize the expected revenue. In particular, let r_{ij} and μ_{ij} denote the revenue and customer preference weight associated with selling product j to customer class i, respectively, and μ_{i0} is the no-purchase preference in class i. Then, for a given assortment S, the probability that customer class i chooses product $j \in S$ is $\mu_{ij}/(\mu_{i0} + \sum_{j \in S} \mu_{ij})$. Thus, the problem of maximizing the expected revenue for all classes of customers under the mixed multinomial logit model can be formulated as the multiple-ratio fractional binary program of the form

$$\max_{x \in X} \sum_{i \in I} \frac{\sum_{j \in J} r_{ij} \mu_{ij} x_j}{\mu_{i0} + \sum_{j \in J} \mu_{ij} x_j}. \tag{2.24}$$

In (2.24) variable x_j is 1 if and only if the decision maker offers product j. Note that (2.24) is a special case of the generally structured FPs, since in each ratio $i \in I$ the coefficient of x_j, for all $j \in J$ in the numerator, i.e., $a_{ij} = r_{ij}\mu_{ij}$, is proportional to its coefficient in the denominator, $b_{ij} = \mu_{ij}$; moreover, $a_{i0} = 0$ and $b_{i0} = \mu_{i0}$.

Problem (2.24) can be transformed into an equivalent minimization problem. Specifically, based on the related discussion in Appendix A.1, for each customer class $i \in I$ we have

$$\frac{\sum_{j \in J} r_{ij} \mu_{ij} x_j}{\mu_{i0} + \sum_{j \in J} \mu_{ij} x_j} = \frac{k_i \mu_{i0} + \sum_{j \in J} (r_{ij}\mu_{ij} + k_i \mu_{ij}) x_j}{\mu_{i0} + \sum_{j \in J} \mu_{ij} x_j} - k_i,$$

for any $k_i \in \mathbb{R}$. Let $k_i = -\overline{r}_i = -\max_{j \in J} r_{ij}$, then

$$\max_{x \in X} \sum_{i \in I} \frac{-\overline{r}_i \mu_{i0} + \sum_{j \in J} (r_{ij}\mu_{ij} - \overline{r}_i \mu_{ij}) x_j}{\mu_{i0} + \sum_{j \in J} \mu_{ij} x_j} + \overline{r}_i = -\min_{x \in X} \sum_{i \in I} \frac{\overline{r}_i \mu_{i0} + \sum_{j \in J} \mu_{ij} (\overline{r}_i - r_{ij}) x_j}{\mu_{i0} + \sum_{j \in J} \mu_{ij} x_j} + \overline{r}_i. \tag{2.25}$$

Transformation (2.25) is precisely the transformation used in [85] and satisfies the data non-negativity assumption. To satisfy the data integrality assumption, we multiply by 10 each of the terms $\mu_{i0}\overline{r}_i$, $\mu_{ij}(\overline{r}_i - r_{ij})$, μ_{i0}, and μ_{ij}, for all $i \in I$ and $j \in J$, and round them down to the nearest integer values.

For our test instances, we generate the data as in the assortment optimization problem considered in [85]. Specifically, the product prices are the same across the customer classes, i.e., $r_{ij} = r_j$ for all $i \in I$ and drawn from a $U[1,3]$ distribution. Moreover, the preferences μ_{ij} are drawn from a $U[0,1]$ distribution, and $\mu_{i0} = 5$ for all $i \in I$.

Moreover, Şen et al. [85] consider $X = \left\{x \in \mathbb{B}^n \mid \sum_{j=1}^n x_j \leqslant \kappa\right\}$. We let $\kappa \in \{10\% \cdot n,\ 20\% \cdot n,\ n\}$. The cardinality constraints: $\kappa = 10\% \cdot n$ and $\kappa = 20\% \cdot n$ correspond to a "small" and "large" retailer, respectively, where there is a physical limitation on the number of products that can be offered to customers. Additionally, $\kappa = n$ indicates the unconstrained case, i.e., $X = \mathbb{B}^n$, and it corresponds to an online retailer with the ability to sell many products [61].

Şen et al. [85] consider only the combinations $n = 200,\ m = 20$ and $n = 500,\ m = 50$. For these combinations we use the the same data (now part of the conic benchmark library, CBLIB) available at `http://cblib.zib.de`. For the other combinations of n and m tested in the paper we generate the data randomly in the aforementioned fashion.

- *Uniformly generated data set.* For the second setting, we use data generated similarly to [16, 61]. Specifically, the coefficients a_{ij} and b_{ij} are each sampled from a (discrete) $U[0,20]$ distribution, except for b_{i0} which is sampled from a $U[1,20]$. The feasible region is given by $X = \left\{x \in \mathbb{B}^n \mid \sum_{j=1}^n x_j = \kappa\right\}$ with $\kappa \in \{10\% \cdot n,\ 20\% \cdot n\}$; we also consider the unconstrained case ($X = \mathbb{B}^n$).

For constrained instances, since in both settings X contains a single cardinality constraint, the number of variables added in the binary-expansion formulations can be reduced by setting $\theta_i^a := \lfloor \log_2 \left(\sum_{j=1}^{\kappa} a_{i[j]}\right)\rfloor + 1$ and $\theta_i^b := \lfloor \log_2 \left(\sum_{j=1}^{\kappa} b_{i[j]}\right)\rfloor + 1$, for all $i \in I$, where $a_{i[j]}$ and $b_{i[j]}$ denote the j-th largest element of a_i and b_i, respectively. For all the formulations – except LF, $\text{LF}_{\log}$, and $\text{LF}_{\log}^{\text{P}}$ – we use $y_i^L = 1/(b_{i0} + \sum_{j=1}^{\kappa} b_{i[j]})$ and $y_i^U = 1/b_{i0}$ as valid lower and upper bounds for linearization, respectively. For LF, $\text{LF}_{\log}$, and $\text{LF}_{\log}^{\text{P}}$ we use $t_i^L = 0$ and $t_i^U = (a_{i0} + \sum_{j=1}^{\kappa} a_{i[j]})/b_{i0}$ as valid bounds.

Metrics. For each of the formulations we define, $z^\star$: the objective function value of an optimal integer solution (or the best-found integer solution if an optimal solution could not be found by the formulation within the time limit), $z^{\texttt{Rlx}}$: the optimal objective function value of the continuous relaxation, $z^{\texttt{Ron}}$: the objective function value obtained after processing the root node (i.e., after adding polymatroid cuts and considering other strengthening techniques used by CPLEX), and $z^{\texttt{Bbn}}$: the best lower-bound at the termination of the solver. Moreover, we define $Z^\star$ as the objective function value of the best-known integer solution over all solution methods. Note that for MILP formulations, $z^{\texttt{Rlx}} \leqslant z^{\texttt{Ron}}$ as additional constraints are added at the root node. For MICQP formulations, this is not necessarily the case: $z^{\texttt{Rlx}}$ is found via interior point methods, while $z^{\texttt{Ron}}$ is obtained after solving a linear outer approximation which may have a weaker continuous relaxation.

Then, in our experiments, we report the following metrics of interest: the continuous relaxation gap, `Rlx-gap` = $\frac{|Z^\star - z^{\texttt{Rlx}}|}{Z^\star} \times 100\%$; the root node gap, `Ron-gap` = $\frac{|Z^\star - z^{\texttt{Ron}}|}{Z^\star} \times 100\%$; the end gap, `End-gap` = $\frac{|z^\star - z^{\texttt{Bbn}}|}{z^\star} \times 100\%$; the best bound gap, `Bbn-gap`= $\frac{|Z^\star - z^{\texttt{Bbn}}|}{Z^\star} \times 100\%$; and the optimality gap, `Opt-gap`= $\frac{|Z^\star - z^\star|}{Z^\star} \times 100\%$. In addition, we report the `Time` in seconds required to solve the problems, and the number of branch-and-bound `Nodes` explored. In all cases we report the averages over five instances generated with the same parameters (n, m, κ).

2.4.2 Preliminary analysis

Here, we briefly analyze the results for the MILP and MICQP formulations outlined in Section 2.2. More detailed results are omitted from the current discussion for the sake of brevity and are reported in Appendix A.2.

In particular, the extended formulations LEF and CEF are stronger (they have better `Rlx-gap`) than the corresponding compact formulations LF and CF, respectively. The extended formulations also have better `time` and `End-gap` than the corresponding compact formulations; see Tables 19 and 20 for the results and Appendix A.2.1 for an additional discussion.

Although LF has a poor performance even for small instances, its "binarization", i.e., LF_{log}, leads to significant improvements in the running time due to the reduction in the size of the formulation, see Tables 21 and 22 and the discussion in Appendix A.2.2. These results are consistent with the previous results in the literature (see, e.g., [16, 61]) that $\mathrm{LF}_{\log}$ has a superior performance over LF and $\mathrm{LEF}_{\log}$.

Additionally, recall that among the existing formulations in the literature the polymatroid cuts have been employed only for the strengthening of CF and the resulting formulation, i.e., CF^{P} significantly outperforms CF with respect to the metrics `time`, `End-gap`, and `Ron-gap`. See [6] and our results presented in Tables 25 and 26; we also refer to Appendix A.2.3 for an additional discussion.

2.4.3 Standout vs. the state-of-the-art formulations

In this section, we further compare the performance of the state-of-the-art formulations available in the literature identified in Section 2.4.2, i.e., the extended MILP formulation LEF and the compact binary-expansion formulation $\mathrm{LF}_{\log}$ as well as the extended MICQP formulation CEF and the compact MICQP formulation with polymatroid cuts CF^{P}. In addition, we report the results of the two standout formulations derived in Section 2.3: the binary-expansion MILP and MICQP formulations strengthened with polymatroid cuts, i.e., $\mathrm{LF}^{\mathrm{P}}_{\log}$ and $\mathrm{CEF}^{\mathrm{P}}_{\log}$, respectively. In Appendix A.2, we present additional computational results and discuss in detail our extensive experiments to evaluate the individual and combined effects of the enhancements developed in this chapter.

Tables 3 and 4 show the results for the assortment and the uniformly generated instances, respectively, and for different values of n, m and κ with respect to the running time and the end gap. A detailed comparison of the standout and the state-of-the-art formulations with respect to all the metrics defined in Section 2.4.1 is provided in Tables 17 and 18 of Appendix A.2. In the tables, we use the "†" symbol to denote that CPLEX was unable to fully process the root node of the branch-and-bound tree within the time limit of one hour for a given formulation.

Observe that, overall, the uniformly generated instances used in [16], see Table 4, are much more difficult to solve than the assortment instances used in [85], see Table 3. In particular, only uniformly generated instances with $n \leqslant 50$ can be solved to optimality (by any formulation), while assortment instances with $n \leqslant 500$ can in general be handled well by MICQP formulations.

Figure 3 shows the number of continuous and binary variables as well as the number of linear and rotated cone constraints of the formulations as a function of dimension (n). Figure 4 depicts the performance profile of solution methods and can be used to evaluate the effectiveness of each formulation in *easy* instances (the instances that are solved to optimality by at least one solution method). Figure 5 portrays the end gaps across all instances as a function of the dimension and can be utilized to explore the effectiveness of each formulation in *hard*, larger, instances (the instances that are not solved to optimality by any solution method in the time limit). Figures 6 and 7 show the relaxation gaps and the root node gaps, respectively, across all instances as a function of the dimension and can be used to evaluate the strengths of the convex relaxations.

In the *easy* instances, we see from Figure 4 that CEF performs best. Formulation CEF also has the best relaxation strength among the formulations presented (Figures 6 and 7). In fact, in most of the instances that CEF solves, `Ron-gap` is nearly 0 and optimality is proven with a few branch-and-bound nodes (see Table 3 with $n \leqslant 500$).

However, when *hard* instances are also taken into account, then CEF is not necessarily the best formulation, mainly due to the fact that its large size (Figure 3) hampers its performance, and other formulations match or improve upon the end gaps of CEF even for $100 \leqslant n \leqslant 500$, see Figure 5. Indeed, in the uniformly generated instances (Table 4), CEF is not able to fully close the root node gap, and the performance in branch-and-bound is substantially impaired due to the difficulty of solving the large, nonlinear convex subproblems. Additionally, existing conic formulations CF^{P} and CEF scale the worst among the formulations presented, and CPLEX is unable to process the root node for those formulations in large settings with $n \geqslant 1000$.

On the other hand, LF_{log} has the best scaling properties among the previously proposed formulations in the literature. Notably, unlike LEF, CEF and CF^{P}, it is able to fully process the root node in all instances with $n \geqslant 1000$ and explore thousands of branch-and-bound nodes or more. Moreover, it is competitive with the other formulations in terms of end gaps for $n \leqslant 100$ and outperforms other existing formulations at $n = 100$, see Figure 5. However, it has substantially weaker convex relaxations than all the other formulations (see Figures 6 and 7), and as a consequence it struggles on the *easy* instances (Figure 4) and has worse end gaps for $200 \leqslant n \leqslant 500$ than the other previously proposed formulations.

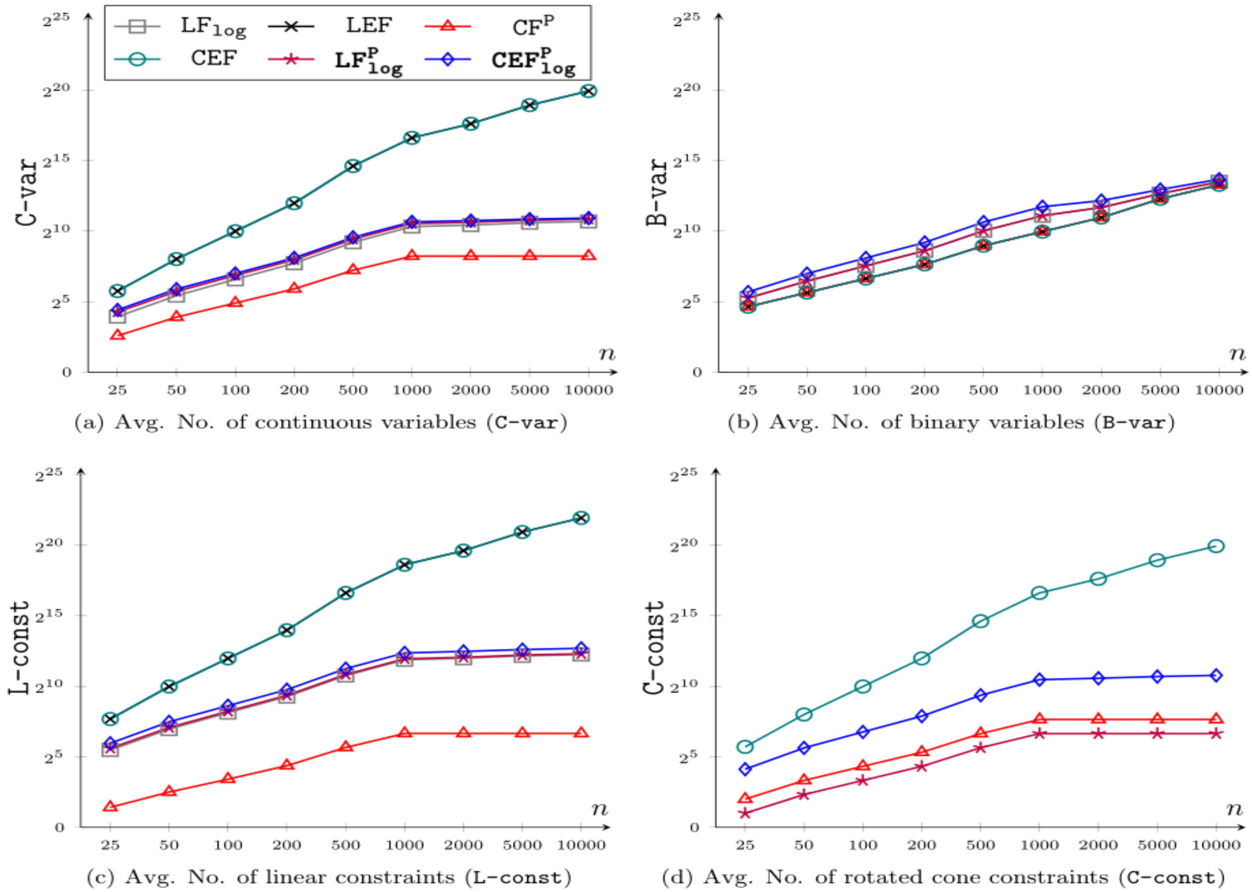
The new formulations $\text{LF}^{\text{P}}_{\log}$ and $\text{CEF}^{\text{P}}_{\log}$, which combine the binary-expansion technique, conic strengthening and polymatroid strengthening, perform well across all dimensions. Binarization leads to a significant size reduction especially in larger instances, e.g., for $n = 10,000$ the number of rotated cone constraints from 1,000,100 (corresponding to CEF) reduces to 1,750 (corresponding to $\text{CEF}^{\text{P}}_{\log}$), see Figure 3. On the other hand, polymatroid cuts improve the convex relaxation quality of the formulations. In particular, from Figure 7 we observe that $\text{LF}^{\text{P}}_{\log}$ and $\text{CEF}^{\text{P}}_{\log}$ are able to achieve a substantial root node strengthening over the simple binary-expansion formulation $\text{LF}_{\log}$, and approximately match the strength of LEF. As a consequence, in the *easy* instances (Figure 4), they also match the performance of LEF and consistently outperform $\text{LF}_{\log}$, but still lag behind the stronger conic formulations CEF and CF^{P}.

However, once *hard* instances are also taken into account, we see from Figure 5 that they achieve the best performance overall. Notably, they match the performance of the best formulations for $n \leqslant 500$, but they scale to instances with n in the thousands and consistently outclass $\text{LF}_{\log}$ (the only other formulation that scales to those instances).

2.5 Concluding remarks

Fractional 0-1 programming problems have traditionally been tackled by reformulating the problems as MILPs with a large number of variables and constraints. However, new techniques have recently been proposed to improve upon the classical MILP formulations. This chapter focuses on two such recent enhancements: a binary-expansion technique that decreases the number of variables and constraints at the expense of weak convex relaxations; and conic and submodular strengthenings, which improve the convex relaxations at the expense of even larger and harder to solve convex relaxations. Naturally, these two ideas are at odds with each other, and which enhancement is preferable largely depends on each particular instance.

In this chapter, we develop formulations that combine both enhancement ideas. The new formulations are compact and require a modest number of variables and constraints, yet retain the relaxation strength of formulations of much larger sizes. As a consequence, the new formulations are able to perform well across all instance classes. Specifically, in our computations using benchmark instances, we observe that the new formulations perform as well as the best existing methods in small and easy problems, and vastly outperform existing methods in larger and harder instances.



(a) Avg. No. of continuous variables (`C-var`)

(b) Avg. No. of binary variables (`B-var`)

(c) Avg. No. of linear constraints (`L-const`)

(d) Avg. No. of rotated cone constraints (`C-const`)

Figure 3: The average sizes (numbers of continuous and binary variables as well as numbers of linear and rotated cone constraints) of formulations as a function of dimension (n). The averages are over five test instances of both the assortment [85] and the uniformly generated [16] data sets and capacity sizes $\kappa \in \{10\% \cdot n,\ 20\% \cdot n\}$ as well as the unconstrained case.

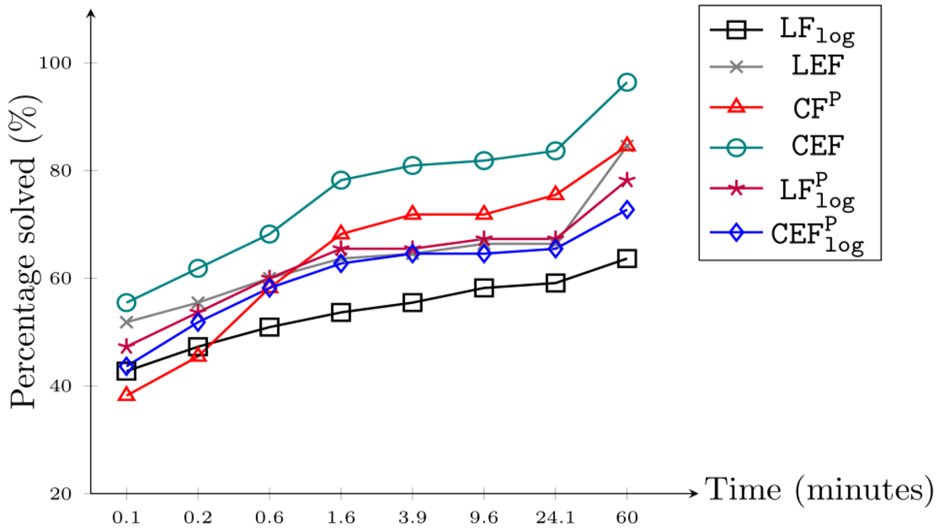


Figure 4: Performance profile for *easy* instances, that are the instances solved to optimality by at least one formulation. They include 80 instances of the assortment data (all instances with $n \leqslant 500$ and five instances with $n = 1000$), and 30 instances of the uniformly generated data (all instances with $n \leqslant 50$). We depict the percentage of such instances that could be solved as a function of the time (in log scale) for each formulation.

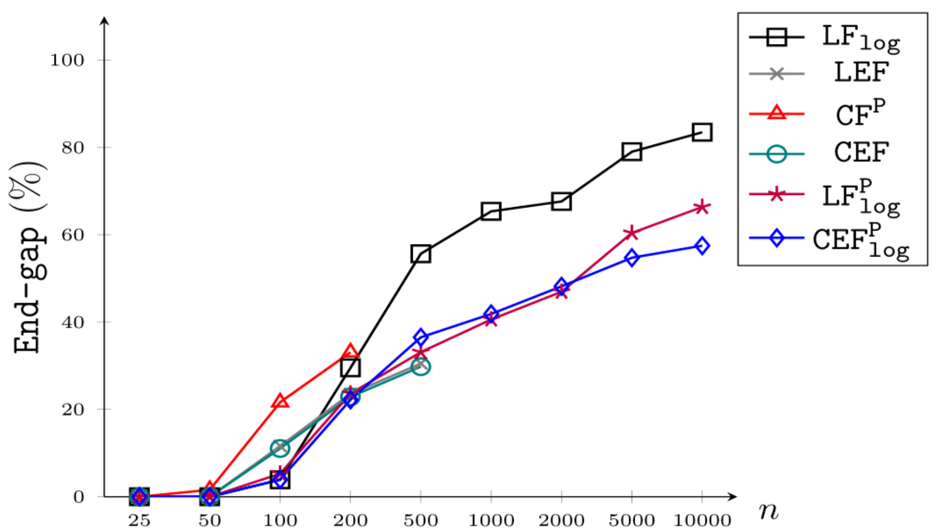


Figure 5: Average end gap (`End-gap`) for all instances as a function of dimension. No gap is reported when a given formulation is unable to solve the root node within the time limit.

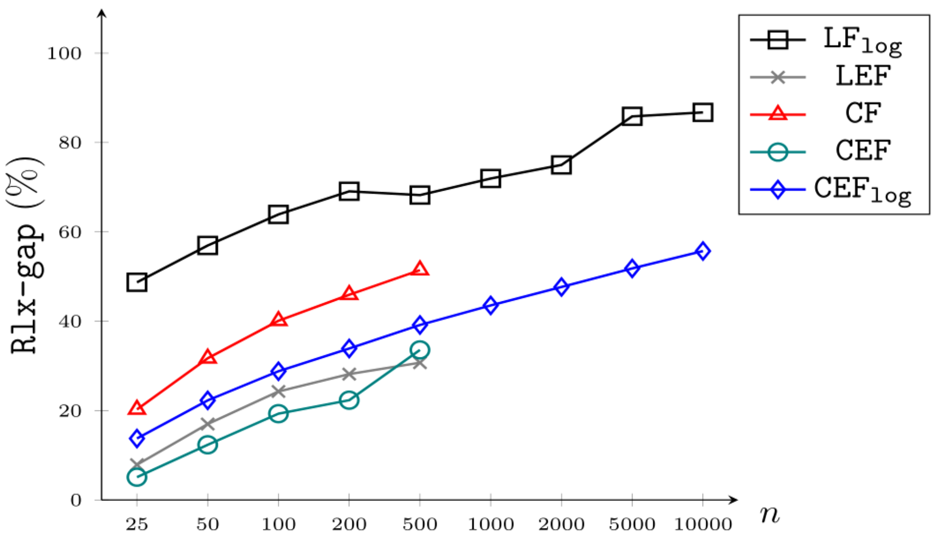


Figure 6: Average relaxation gap (Rlx-gap) for all instances as a function of dimension. Observe that Rlx-gap does not account for the effect of polymatroid cuts. No gap is reported when a given formulation is unable to solve the root node within the time limit.

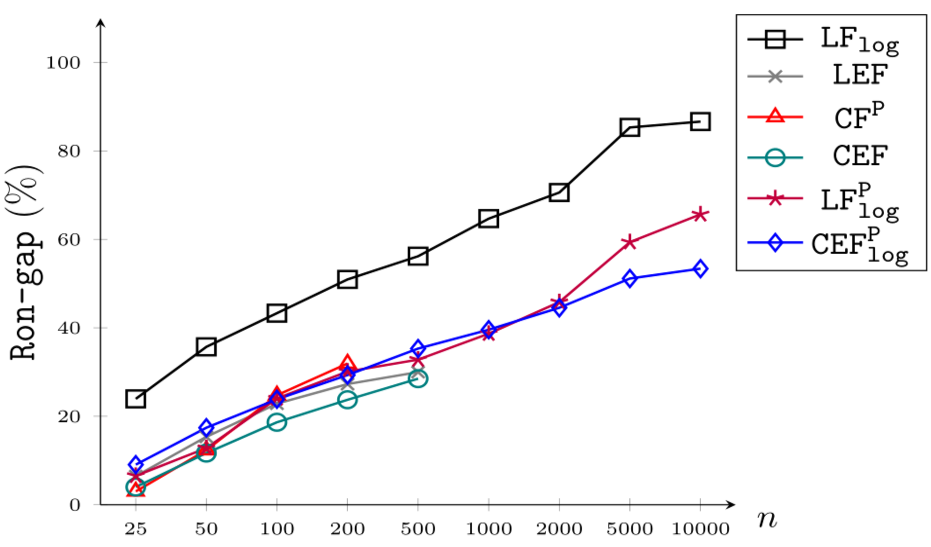


Figure 7: Average root node gap (Ron-gap) for all instances as a function of dimension. Observe that Ron-gap accounts for the strengthening from polymatroid cuts, but it is also impacted unfavorably by the use of (possibly weak) linear outer approximations. No gap is reported when a given formulation is unable to solve the root node within the time limit.

Table 3: Computational results to evaluate the best existing methods in the literature against the standout formulations for the *assortment data set* [85]. For each choice of n, m, and κ the best average time and the best average End-gap (if Time⩾ 3600 sec.) are in **bold**.

n,m	κ / Ref.	$10\% \cdot n$ Time	$10\% \cdot n$ End-gap	$20\% \cdot n$ Time	$20\% \cdot n$ End-gap	Unconstrained Time	Unconstrained End-gap
25,2*	LF_{log}	**0**	0.0%	**0**	0.0%	1	0.0%
	LEF	**0**	0.0%	**0**	0.0%	**0**	0.0%
	CF^P	**0**	0.0%	**0**	0.0%	1	0.0%
	CEF	**0**	0.0%	**0**	0.0%	**0**	0.0%
	LF^P_{log}	**0**	0.0%	**0**	0.0%	**0**	0.0%
	CEF^P_{log}	1	0.0%	**0**	0.0%	2	0.0%
50,5*	LF_{log}	1	0.0%	2	0.0%	18	0.0%
	LEF	**0**	0.0%	**1**	0.0%	**0**	0.0%
	CF^P	1	0.0%	2	0.0%	4	0.0%
	CEF	1	0.0%	**1**	0.0%	1	0.0%
	LF^P_{log}	**0**	0.0%	**1**	0.0%	6	0.0%
	CEF^P_{log}	**0**	0.0%	2	0.0%	21	0.0%
100,10*	LF_{log}	979	0.0%	3155	0.4%	3600	1.6%
	LEF	3357	1.6%	2190	0.2%	**1**	0.0%
	CF^P	10	0.0%	20	0.0%	25	0.0%
	CEF	6	0.0%	**4**	0.0%	6	0.0%
	LF^P_{log}	**1**	0.0%	6	0.0%	3600	0.8%
	CEF^P_{log}	2	0.0%	22	0.0%	3600	0.3%
200,20*	LF_{log}	3600	6.7%	3600	8.7%	3600	24.1%
	LEF	3600	8.6%	3600	1.1%	**29**	0.0%
	CF^P	**27**	0.0%	64	0.0%	1562	0.2%
	CEF	73	0.0%	**40**	0.0%	59	0.0%
	LF^P_{log}	710	0.0%	3400	0.3%	3600	6.3%
	CEF^P_{log}	2353	0.5%	3600	2.2%	3600	6.4%
500,50*	LF_{log}	3600	39.8%	3600	54.0%	3600	55.7%
	LEF	3600	8.3%	**2520**	0.2%	3501	**0.4%**
	CF^P	**1194**	0.0%	3452	0.3%	3600	7.7%
	CEF	3611	0.2%	2620	0.0%	3604	0.5%
	LF^P_{log}	3600	0.8%	3600	3.3%	3600	15.2%
	CEF^P_{log}	3600	4.7%	3600	12.2%	3601	26.1%
1000,100**	LF_{log}	3600	55.9%	3600	62.7%	3600	76.5%
	LEF	3600	13.9%	3722	**0.9%**	3600	**1.7%**
	CF^P	3600	†	3600	†	3600	†
	CEF	3605	†	3600	†	3600	†
	LF^P_{log}	3601	†	3601	20.9%	3601	26.1%
	CEF^P_{log}	3601	**10.0%**	3600	22.6%	3600	33.8%
2000,100**	LF_{log}	3600	57.8%	3600	70.5%	3600	78.3%
	LEF	3601	†	3600	†	3601	†
	CF^P	3600	†	3600	†	3600	†
	CEF	3600	†	3600	†	3600	†
	LF^P_{log}	3601	†	3600	41.4%	3601	**33.1%**
	CEF^P_{log}	3600	**16.1%**	3600	**30.7%**	3600	53.4%
5000,100**	LF_{log}	3600	78.1%	3600	80.6%	3601	83.5%
	LEF	7807	†	8155	†	7241	†
	CF^P	3600	†	3600	†	3600	†
	CEF	3600	†	3600	†	3600	†
	LF^P_{log}	3601	**29.2%**	3601	49.0%	3601	**50.7%**
	CEF^P_{log}	3600	39.3%	3600	**40.6%**	3600	58.4%
10000,100**	LF_{log}	3600	88.4%	3600	83.1%	3602	93.0%
	LEF	4225	†	4026	†	3603	†
	CF^P	3600	†	3600	†	3600	†
	CEF	3600	†	3600	†	3600	†
	LF^P_{log}	3601	55.4%	3601	53.2%	3601	**54.7%**
	CEF^P_{log}	3600	**33.4%**	3601	**45.4%**	3601	†

*_easy_ instances

**_hard_ instances

Table 4: Computational results to evaluate the best existing methods in the literature against the standout formulations for the *uniformly generated data set* [16]. For each choice of n, m, and κ, the best average time and the best average End-gap (if Time⩾ 3600 sec.) are in **bold**.

n,m	κ / Ref.	$10\% \cdot n$ Time	$10\% \cdot n$ End-gap	$20\% \cdot n$ Time	$20\% \cdot n$ End-gap	Unconstrained Time	Unconstrained End-gap
25,2*	LF_{log}	**0**	0.0%	1	0.0%	1	0.0%
	LEF	**0**	0.0%	**0**	0.0%	**0**	0.0%
	CF^P	3	0.0%	4	0.0%	4	0.0%
	CEF	**0**	0.0%	**0**	0.0%	1	0.0%
	LF^P_{log}	**0**	0.0%	1	0.0%	1	0.0%
	CEF^P_{log}	1	0.0%	1	0.0%	6	0.0%
50,5*	LF_{log}	3	0.0%	20	0.0%	52	0.0%
	LEF	**2**	0.0%	**13**	0.0%	**43**	0.0%
	CF^P	78	0.0%	3601	6.5%	2903	3.0%
	CEF	3	0.0%	18	0.0%	100	0.0%
	LF^P_{log}	9	0.0%	27	0.0%	85	0.0%
	CEF^P_{log}	6	0.0%	26	0.0%	86	0.0%
100,10**	LF_{log}	3600	**5.0%**	3600	**5.0%**	3600	11.2%
	LEF	3600	12.3%	3600	17.1%	3600	38.5%
	CF^P	3600	43.5%	3600	44.3%	3600	42.0%
	CEF	3600	10.7%	3600	15.5%	3600	40.1%
	LF^P_{log}	3600	7.5%	3600	6.1%	3600	17.2%
	CEF^P_{log}	3600	7.2%	3603	5.2%	3600	**10.9%**
200,20**	LF_{log}	3600	41.7%	3600	37.7%	3600	58.2%
	LEF	3600	**30.0%**	3600	31.1%	3600	70.6%
	CF^P	3600	65.8%	3600	61.6%	3600	70.9%
	CEF	3600	30.9%	3600	**30.0%**	3600	76.4%
	LF^P_{log}	3600	41.6%	3600	35.6%	3600	58.0%
	CEF^P_{log}	3600	35.5%	3600	34.3%	3600	**54.4%**
500,50**	LF_{log}	3600	48.7%	3600	48.7%	3600	87.0%
	LEF	3600	**42.8%**	3600	**41.1%**	3600	90.3%
	CF^P	3600	†	3600	†	3600	84.9%
	CEF	3603	**42.8%**	3604	41.8%	3603	93.4%
	LF^P_{log}	3600	48.4%	3600	48.1%	3600	**82.9%**
	CEF^P_{log}	3600	46.3%	3600	43.1%	3600	86.7%
1000,100**	LF_{log}	3600	50.3%	3600	50.1%	3600	96.6%
	LEF	3601	†	3601	†	3601	†
	CF^P	3600	†	3600	†	3600	95.6%
	CEF	3600	†	3600	†	3600	†
	LF^P_{log}	3600	50.2%	3600	50.2%	3600	**91.9%**
	CEF^P_{log}	3600	**48.0%**	3600	**44.5%**	3600	92.2%
2000,100**	LF_{log}	3600	50.7%	3600	50.6%	3600	97.8%
	LEF	3601	†	3602	†	3601	†
	CF^P	3600	†	3600	†	3600	†
	CEF	3600	†	3600	†	3600	†
	LF^P_{log}	3600	50.8%	3600	50.7%	3600	**94.8%**
	CEF^P_{log}	3600	**47.8%**	3600	**44.6%**	3600	96.6%
5000,100**	LF_{log}	3600	67.9%	3600	65.0%	3601	98.8%
	LEF	4755	†	3938	†	3603	†
	CF^P	3600	†	3600	†	3600	†
	CEF	3600	†	3600	†	3600	†
	LF^P_{log}	3600	68.8%	3600	67.9%	3601	**96.9%**
	CEF^P_{log}	3600	**46.7%**	3601	**45.2%**	3601	98.3%
10000,100**	LF_{log}	3600	68.6%	3600	68.2%	3601	99.4%
	LEF	9500	†	6022	†	5619	†
	CF^P	3600	†	3600	†	3600	†
	CEF	3600	†	3600	†	3600	†
	LF^P_{log}	3601	68.5%	3601	68.4%	3601	**97.8%**
	CEF^P_{log}	3601	**47.5%**	3600	**44.8%**	3600	†

*_easy_ instances

**_hard_ instances

3.0 Robust Fractional 0-1 Programming

3.1 Introduction

In practice, the parameters of an optimization problem are often subject to uncertainty, and existing solution methods for deterministic FPs, including the methods discussed in Chapters 1 and 2, may not be adequate for problems with unknown parameters. Our approach to uncertain fractional 0-1 programming falls within the framework of robust optimization.

Specifically, in this chapter we consider the generally structured fractional 0-1 programs in maximization form given by

$$\text{(FP)} \qquad \max_{x\in X} \sum_{i\in I} \frac{a_{i0} + \sum\limits_{j\in J} a_{ij}x_j}{b_{i0} + \sum\limits_{j\in J} b_{ij}x_j},$$

where $I = \{1, ..., m\}$, $J = \{1, ..., n\}$ and $X \subseteq \mathbb{B}^n$ for $\mathbb{B} := \{0, 1\}$. Then we assume that some or all of the coefficients a_{ij} and b_{ij} may not be known exactly, but are modeled as bounded random variables $\widetilde{a}_{ij}$ and $\widetilde{b}_{ij}$, respectively. These coefficients are presumed to lie in some uncertainty set $\mathcal{U}$; that is, $(\widetilde{a}, \widetilde{b}) \in \mathcal{U}$. Then the robust counterpart of FP with respect to the uncertainty set $\mathcal{U}$ optimizes against the worst-case scenario:

$$\text{(RFP[}\mathcal{U}\text{])} \qquad Z^{\star}_{\mathcal{U}} = \max_{x\in X} \min_{(\widetilde{a},\widetilde{b})\in\mathcal{U}} \sum_{i\in I} \frac{a_{i0} + \sum\limits_{j\in J} \widetilde{a}_{ij}x_j}{b_{i0} + \sum\limits_{j\in J} \widetilde{b}_{ij}x_j}.$$

Throughout the chapter, we assume that the data satisfy the following assumption:

Assumption 1. *For all $x \in X$, $(\widetilde{a}, \widetilde{b}) \in \mathcal{U}$ and $i \in I$, $a_{i0} + \sum_{j\in J} \widetilde{a}_{ij}x_j \geqslant 0$ and $b_{i0} + \sum_{j\in J} \widetilde{b}_{ij}x_j > 0$.*

Most fractional programming problems typically have non-negative data, since such data represent probabilities, prices, weights, utilities, etc. - see, e.g., [17] and the applications described therein. The portion of Assumption 1 related to a strictly positive denominator is a commonly made assumption for the deterministic version, see, e.g., [15, 43]. Moreover, the non-negative numerator assumption is not restrictive, since by adding a sufficiently large constant value to each ratio we can transform its numerator into the one which takes only non-negative values for any $(\widetilde{a}, \widetilde{b}) \in \mathcal{U}$ and $x \in X$. In the following, we define $(t)^{+} = \max\{0, t\}$ for any $t \in \mathbb{R}$, and let $A \times B$ denote the Cartesian product of sets A and B.

Contributions and the structure of the chapter. To the best of our knowledge, this study is the first work that addresses the robust fractional 0-1 programming in its general structure. We perform a comprehensive study of RFP[$\mathcal{U}$] that includes several types of the budgeted uncertainty sets, and also encompasses single- and multiple-ratio cases. We also briefly explore the complexity of RFP[$\mathcal{U}$] for general polyhedral $\mathcal{U}$. The structure of the chapter can be summarized as follows.

- In Section 3.2, we introduce the (disjoint and joint) generalizations of the budgeted uncertainty set for fractional 0–1 programs and discuss computational complexity of RFP.
- In Section 3.3, we propose an approach to find an optimal solution of single-ratio RFP by solving a polynomial number of linear optimization problems over X; in particular, if linear optimization over X is polynomial-time solvable, then so is RFP[$\mathcal{U}$].
- In Section 3.4, we extend classical MILP formulations for FP to tackle multiple-ratio RFP[$\mathcal{U}$], and also exploit the binary-expansion technique to improve the efficacy of the MILPs. We also provide some insights on the selection of the appropriate level of uncertainty.
- In Section 3.5, we present computations with real and synthetic data. Additionally, we examine the price of robustness and evaluate the performance of the proposed MILPs via extensive computational experiments.

3.2 Model of data uncertainty

The selection of an appropriate uncertainty set can affect the tractability of a robust optimization problem. In this section, we describe the budgeted uncertainty set, and several variations thereof, for fractional 0-1 programming as considered in this chapter, which lead to tractable (polynomial-time) methods for single-ratio RFP[$\mathcal{U}$] in Section 3.3. On the other hand, we also demonstrate that the robust counterpart of a polynomially-solvable unconstrained single-ratio FP (with strictly positive denominator) is NP-hard for a general polyhedral uncertainty set $\mathcal{U}$.

In particular, following the convention introduced by Bertsimas and Sim [12, 13], each unknown coefficient $\widetilde{a}_{ij}$ and $\widetilde{b}_{ij}$ lies in a symmetric interval centered on the nominal value, i.e., $\widetilde{a}_{ij} \in [a_{ij}-d^a_{ij}, a_{ij}+d^a_{ij}]$ and $\widetilde{b}_{ij} \in [b_{ij}-d^b_{ij}, b_{ij}+d^b_{ij}]$ with $d^a_{ij}, d^b_{ij} \geqslant 0$. The coefficients d^a_{ij} and d^b_{ij} denote the potential deviation from nominal values a_{ij} and b_{ij}, respectively, for each $i \in I, j \in J$.

Additionally, it is unlikely for all of the coefficients to simultaneously change to their worst-case values. Hence, only a predetermined number of the unknown coefficients take values different from their nominal value. Given a ratio $i \in I$ and vectors $\widetilde{a}_i, \widetilde{b}_i \in \mathbb{R}^n$, let $S_i(\widetilde{a}_i) = \{j \in J \mid \widetilde{a}_{ij} \neq a_{ij}\}$ and $S_i(\widetilde{b}_i) = \left\{j \in J \mid \widetilde{b}_{ij} \neq b_{ij}\right\}$ be the set of indices of the uncertain parameters whose values are different from the nominal in the numerator and the denominator, respectively.

Uncertainty pertaining to linear 0-1 constraints is covered in literature [12], thus we assume that the constraint coefficients are fixed. Furthermore, we assume without loss of generality that the data is integral (otherwise, the rational coefficients can be scaled to satisfy this assumption). Hence:

Assumption 2. *All data is integer, i.e.,* $a_{i0}, b_{i0}, a_{ij}, b_{ij} \in \mathbb{Z}$, *and* $d^a_{ij}, d^b_{ij} \in \mathbb{Z}_+$ *for all* $i \in I, j \in J$.

Disjoint uncertainty set. Given $\Gamma^a_i, \Gamma^b_i \in \{0, 1, \dots, n\}$ as the budget of uncertainty or the level of conservatism, for each $i \in I$ we define

$$\mathcal{U}^a_i = \left\{\widetilde{a}_i \in \mathbb{R}^n \mid \widetilde{a}_{ij} \in [a_{ij} - d^a_{ij}, a_{ij} + d^a_{ij}] \text{ for } j \in J,\ |S_i(\widetilde{a}_i)| \leqslant \Gamma^a_i\right\}, \text{ and} \tag{3.1}$$

$$\mathcal{U}^b_i = \left\{\widetilde{b}_i \in \mathbb{R}^n \mid \widetilde{b}_{ij} \in [b_{ij} - d^b_{ij}, b_{ij} + d^b_{ij}] \text{ for } j \in J,\ |S_i(\widetilde{b}_i)| \leqslant \Gamma^b_i\right\}. \tag{3.2}$$

Note that $\mathcal{U}^a_i$ and $\mathcal{U}^b_i$ correspond to the budgeted uncertainty sets studied in [12, 13], and Γ^a_i and Γ^b_i are the number of coefficients allowed to vary from their nominal value in the numerator and the denominator of the i-th ratio, respectively. Then the *disjoint uncertainty set* for fractional programming is

$$\mathcal{U}^{ab} = \left\{(\widetilde{a}, \widetilde{b}) \in \mathbb{R}^{m\times n} \times \mathbb{R}^{m\times n} \mid (\widetilde{a}_i, \widetilde{b}_i) \in \mathcal{U}^a_i \times \mathcal{U}^b_i, \text{ for all } i \in I\right\}.$$

We refer to $\mathcal{U}^{ab}$ as disjoint since uncertainty of the coefficients of each numerator and denominator is independent from the rest of the data. Also, observe that in the i-th ratio by setting $\Gamma_i^a = 0$ ($\Gamma_i^b = 0$) we can restrict the uncertainty only to the denominator (numerator) of the ratio. Therefore, set $\mathcal{U}^{ab}$ includes sub-cases in which some ratios are subject to uncertainty either only in their denominators or numerators.

Joint uncertainty sets. We now describe four joint uncertainty sets. In contrast with the disjoint uncertainty set above, there is some dependence between the uncertainties related to different numerators and denominators.

- *Shared ratio budget* - Given $\Gamma_i \in \{0, 1, \ldots, 2n\}$, for each $i \in I$ let

$$\mathcal{U}_i = \Big\{(\widetilde{a}_i, \widetilde{b}_i) \in \mathbb{R}^n \times \mathbb{R}^n \mid \widetilde{a}_{ij} \in [a_{ij} - d^a_{ij}, a_{ij} + d^a_{ij}], \widetilde{b}_{ij} \in [b_{ij} - d^b_{ij}, b_{ij} + d^b_{ij}], \\ |S_i(\widetilde{a}_i)| + |S_i(\widetilde{b}_i)| \leqslant \Gamma_i\Big\}.$$

  The *shared ratio budget uncertainty set* is

$$\mathcal{U}^{\overline{ab}} = \Big\{(\widetilde{a}, \widetilde{b}) \in \mathbb{R}^{m\times n} \times \mathbb{R}^{m\times n} \mid (\widetilde{a}_i, \widetilde{b}_i) \in \mathcal{U}_i, \text{ for all } i \in I\Big\}.$$

  Under the shared ratio budget uncertainty set, uncertainty for the i-th ratio is independent of other ratios, but the uncertainties of its numerator and denominator are connected by a common budget, Γ_i. Specifically, at most Γ_i of coefficients in the i-th ratio's numerator and denominator can change.

  The uncertainty sets $\mathcal{U}^{ab}$ and $\mathcal{U}^{\overline{ab}}$ above arise naturally when there is uncertainty concerning individual coefficients of FP. In some applications, however, the uncertainty of the original problem may have a specific structure which requires a specialized uncertainty set. We now describe three such sets.

- *Matched sets* - Consider the problem of maximizing return on investment or productivity, where "a" corresponds to the return of executing a given project (e.g., dollar amount), and "b" corresponds to the investment costs for the project (e.g., time). Additionally, suppose that undesirable events may occur (e.g., strikes, natural disasters), resulting in a *simultaneous* decrease in the returns and increase in the costs of a given project. Such uncertainty is modeled by the *matched sets uncertainty set*

$$\mathcal{U}^{\overline{ab}}_{=} = \Big\{(\widetilde{a}, \widetilde{b}) \in \mathcal{U}^{\overline{ab}} \mid S_i(\widetilde{a}_i) = S_i(\widetilde{b}_i), \text{ for all } i \in I\Big\}.$$

- *Matched effects* - Consider the assortment optimization problem under the mixed multinomial logit model (see, e.g., [18, 63]),

$$\max_{x \in X} \sum_{i \in I} \frac{\sum_{j \in J} r_{ij} \rho_{ij} x_j}{1 + \sum_{j \in J} \rho_{ij} x_j}, \tag{3.3}$$

  where r_{ij} and ρ_{ij} are the revenues and customer preferences associated with selling product j to customer class i, respectively. Note that if the revenues are known, but the preferences are uncertain, then changes with respect to the nominal values of numerator/denominator coefficients that correspond to the same variable are proportional and of the same sign. The *matched effects uncertainty set*

$$\mathcal{U}^{\overline{ab}}_{\propto} = \left\{ (\widetilde{a}, \widetilde{b}) \in \mathcal{U}^{\overline{ab}}_{=} \mid \frac{a_{ij} - \widetilde{a}_{ij}}{d^a_{ij}} = \frac{b_{ij} - \widetilde{b}_{ij}}{d^b_{ij}}, \text{ for all } i \in I, \ j \in J \right\}$$

  captures this effect.

- *Single budget* - In all of the uncertainty sets defined above, we assume each ratio has its own budget(s) of uncertainty. On the other hand, one may consider an uncertainty set in which a single budget controls the degree of conservatism over all ratios. Specifically, the *single budget uncertainty set* for numerators also arises in the assortment problem (3.3) when the preferences are known, but the revenues are unknown, and is given by

$$\mathcal{U}^{\overline{a}} = \left\{ \widetilde{a} \in \mathbb{R}^{m \times n} \mid \widetilde{a}_{ij} \in [a_{ij} - d^a_{ij}, a_{ij} + d^a_{ij}] \text{ for all } i \in I, j \in J, \ \sum_{i \in I} |S_i(\widetilde{a}_i)| \leqslant \Gamma \right\},$$

  where the budget $\Gamma \in \{0, 1, \ldots, m \cdot n\}$ is shared by all ratios. In words, only numerators are subject to uncertainty and at most Γ of the numerators coefficients are different from their nominal values.

The five uncertainty sets defined above, i.e., $\mathcal{U}^{ab}$, $\mathcal{U}^{\overline{ab}}$, $\mathcal{U}^{\overline{ab}}_{=}$, $\mathcal{U}^{\overline{ab}}_{\propto}$, and $\mathcal{U}^{\overline{a}}$, aim at modeling a broad-range of situations arising in practice; moreover, none is a special case of another. Furthermore, it can be verified that RFP[$\mathcal{U}$], in general, is neither quasi-convex nor quasi-concave.

We show in Section 3.3 that for a polynomial-time solvable FP the considered uncertainty sets lead to polynomial-time solvable robust counterparts RFP[$\mathcal{U}$]. In contrast, note that the robust counterparts corresponding to general polyhedral uncertainty are NP-hard.

RFP[$\mathcal{U}$] for general polyhedral uncertainty is NP-hard. Consider an unconstrained ($X = \mathbb{B}^n$) single-ratio problem with uncertainty limited to the numerator

$$\max_{x \in \mathbb{B}^n} \frac{a_0 + a^T x - \max\limits_{\gamma \in \mathcal{U}} \left\{ (A\gamma)^T x \right\}}{b_0 + b^T x}, \tag{3.4}$$

where $\mathcal{U} = \{\gamma : D\gamma \leqslant d, \gamma \geqslant 0\}$ is a general polyhedral uncertainty set and Assumption 1 holds. Note that, without uncertainty, the deterministic unconstrained single-ratio problem can be solved in polynomial time via a linear-time median-finding algorithm [43]. However, this property does not follow through to the robust counterpart.

Proposition 3. *Problem* (3.4) *is NP-hard.*

Proof. Let $b_0 = 1$ and $b_j = 0$ for $j \in J$, then we have a linear objective with a polyhedral uncertainty set. By Theorem 4 of [20], the resulting problem is NP-hard. □

Similarly, consider the problem with uncertainty restricted to the denominator

$$\max_{x \in \mathbb{B}^n} \frac{a_0 + a^T x}{b_0 + b^T x + \max\limits_{\gamma \in \mathcal{U}} \left\{ (A\gamma)^T x \right\}}. \tag{3.5}$$

Proposition 4. *Problem* (3.5) *is NP-hard.*

Proof. Follows directly from noting that (3.5) is equivalent to

$$\min_{x \in \mathbb{B}^n} \frac{b_0 + b^T x + \max\limits_{\gamma \in \mathcal{U}} \left\{ (A\gamma)^T x \right\}}{a_0 + a^T x},$$

and using an argument similar to the one in Proposition 3. □

In light of these results, in the remainder of this chapter we restrict $\mathcal{U}$ to any disjoint or joint uncertainty sets defined in this section, i.e., $\mathcal{U} \in \{\mathcal{U}^{ab}, \mathcal{U}^{\overline{ab}}, \mathcal{U}^{\overline{ab}}_{=}, \mathcal{U}^{\overline{ab}}_{\propto}, \mathcal{U}^{\overline{a}}\}$, and RFP[$\mathcal{U}$] as the corresponding representation of the robust problem.

3.3 Single-ratio RFP[$\mathcal{U}$]

When the uncertain coefficients of the objective function are in the form of a budgeted uncertainty set, Bertsimas and Sim [12] prove that the solution of the robust counterpart of the nominal binary-linear problem

$$\min_{x \in X} c_0 + \sum_{j \in J} c_j x_j, \tag{3.6}$$

can be found by solving n instances of (3.6). Therefore, if (3.6) is polynomially-solvable, so is its robust counterpart. Similarly, parametric algorithms such as Newton's method [31] and binary-search algorithm [2, 53, 79] can find an optimal solution for the constrained single-ratio FPs by solving a sequence of problems in the form of (3.6).

In this section, we combine and extend the ideas from robust linear programming and deterministic fractional optimization, to propose a solution method for single-ratio RFP[$\mathcal{U}$]. In particular, we show that if there exists a polynomial-time algorithm for linear optimization over X, then RFP[$\mathcal{U}$] is polynomial-time solvable when $\mathcal{U}$ is one of the uncertainty sets described in Section 3.2. We first consider the disjoint uncertainty set $\mathcal{U}^{ab}$ in Section 3.3.1, and then we tackle the joint uncertainty sets in Section 3.3.2.

3.3.1 Disjoint uncertainty set

Herein, we demonstrate how to solve single-ratio RFP[$\mathcal{U}^{ab}$] by solving at most $(n+1)^2$ nominal FPs.

Proposition 5. *Problem* RFP[$\mathcal{U}^{ab}$] *is equivalent to*

$$Z^\star_{\mathcal{U}^{ab}} = \max_{\substack{x \in X, \\ \alpha \in \{0, d^a_1, d^a_2, \dots, d^a_n\}, \\ \beta \in \{0, d^b_1, d^b_2, \dots, d^b_n\}}} \frac{a_0 - \Gamma^a \alpha + \sum_{j \in J} \left(a_j - (d^a_j - \alpha)^+\right) x_j}{b_0 + \Gamma^b \beta + \sum_{j \in J} \left(b_j + (d^b_j - \beta)^+\right) x_j}. \tag{3.7}$$

Proof. Observe that single-ratio RFP[$\mathcal{U}^{ab}$] is equivalent to $\max_{x \in X} \frac{a_0 + \min_{\widetilde{a} \in \mathcal{U}^a} \widetilde{a}^T x}{b_0 + \max_{\widetilde{b} \in \mathcal{U}^b} \widetilde{b}^T x}$, where $\mathcal{U}^a$ and $\mathcal{U}^b$ are the sets given in (3.1)–(3.2). Letting u and v be the indicator vectors of sets $S(\widetilde{a})$ and $S(\widetilde{b})$ respectively, we reformulate RFP[$\mathcal{U}^{ab}$] as

$$\max_{x\in X} \frac{a_0 + \sum_{j\in J} a_j x_j - \max_u \left\{ \sum_{j\in J} d_j^a x_j u_j \right\}}{b_0 + \sum_{j\in J} b_j x_j + \max_v \left\{ \sum_{j\in J} d_j^b x_j v_j \right\}} \tag{3.8}$$

$$\text{s.t.} \sum_{j\in J} u_j \leqslant \Gamma^a, \quad \sum_{j\in J} v_j \leqslant \Gamma^b \qquad (\alpha, \beta)$$

$$0 \leqslant u_j \leqslant 1, \quad 0 \leqslant v_j \leqslant 1 \qquad \forall j \in J. \ (p_j, q_j)$$

Note that there exist integral optimal solutions u^* and v^* to the inner optimization problems in (3.8), since the polytope defined by cardinality and bounding constraints is integral – thus, the formulation above is indeed correct. By taking the dual of (independent) inner optimization problems in the numerator and the denominator of (3.8) with respect to dual variables α, β and p, q, we obtain

$$\max_{\substack{x\in X,\\ \alpha,\beta,p,q\geqslant 0}} \frac{a_0 + \sum_{j\in J} a_j x_j - \left(\Gamma^a \alpha + \sum_{j\in J} p_j\right)}{b_0 + \sum_{j\in J} b_j x_j + \left(\Gamma^b \beta + \sum_{j\in J} q_j\right)} \tag{3.9}$$

$$\text{s.t.} \ \ p_j + \alpha \geqslant d_j^a x_j, \quad q_j + \beta \geqslant d_j^b x_j \qquad \forall j \in J.$$

Clearly, in an optimal solution of (3.9) we have $p_j^* = \left(d_j^a x_j^* - \alpha^*\right)^+ = \left(d_j^a - \alpha^*\right)^+ x_j^*$ and $q_j = \left(d_j^b x_j^* - \beta^*\right)^+ = \left(d_j^a - \alpha^*\right)^+ x_j^*$. Otherwise, we can decrease p_j or q_j and find a solution with a better objective function value.

Additionally, let $E = \left\{ j \in J \mid \left(d_j^a - \alpha^*\right)^+ x_j^* > 0 \right\}$ and observe that if $\alpha^* > 0$ and $\alpha^* \neq d_j^a$ for all $j \in J$ then

$$\Gamma^a(\alpha^* \pm \epsilon) + \sum_{j\in J} \left(d_j^a - (\alpha^* \pm \epsilon)\right)^+ x_j^* = \Gamma^a(\alpha^*) + \sum_{j\in J} \left(d_j^a - \alpha^*\right)^+ x_j^* \pm \epsilon(\Gamma^a - |E|)$$

for sufficiently small $\epsilon > 0$. In particular, depending on the sign of $\Gamma^a - |E|$, we can increase or decrease α^* and find solutions with greater or equal objective function values. Thus, we conclude that there exists an optimal solution where $\alpha^* \in \{0, d_1^a, \ldots, d_n^a\}$ and, similarly, we can conclude that there exists an optimal solution where $\beta^* \in \{0, d_1^b, \ldots, d_n^b\}$. Replacing α, β, p, q in (3.9) by their corresponding optimal values, we find formulation (3.7). □

Hence, RFP[$\mathcal{U}^{ab}$] can be tackled by solving problem (3.7) for each candidate pair $(\alpha, \beta) \in \{0, d_1^a, d_2^a, \ldots, d_n^a\} \times \{0, d_1^b, d_2^b, \ldots, d_n^b\}$ independently.

Theorem 1. *Single-ratio* RFP$[\mathcal{U}^{ab}]$ *can be solved with* $(k^a+1)(k^b+1)$ *calls to an oracle for* FP, *where* k^a *and* k^b *are the numbers of distinct values of* d_j^a *and* d_j^b, $j \in J$, *respectively.*

Theorem 1 implies that if single-ratio FP over X is solvable in strongly polynomial time, then so is its robust counterpart RFP$[\mathcal{U}^{ab}]$. Note that in the worst case $(k^a+1)(k^b+1) = (n+1)^2$, and FP is polynomial-time solvable when linear optimization over X is polynomial-time solvable.

3.3.2 Joint uncertainty sets

It can be observed that the method of Proposition 5 cannot handle single-ratio RFP under joint uncertainty sets, due to interaction between uncertainties in the numerator and the denominator of each ratio. To solve single-ratio RFP under joint uncertainty sets we first show that RFP$[\mathcal{U}^{\overline{ab}}]$, RFP$[\mathcal{U}_{=}^{\overline{ab}}]$, and RFP$[\mathcal{U}_{\propto}^{\overline{ab}}]$ can be formulated as mixed-integer nonlinear programs (MINLPs) with a similar structure (Propositions 6, 7 and 8). Then by exploring some properties of the resulting reformulations (Propositions 9 and 10) we propose a specialized algorithm for solving them (Proposition 11).

Proposition 6. *Problem* RFP$[\mathcal{U}^{\overline{ab}}]$ *is equivalent to*

$$Z^\star_{\mathcal{U}^{\overline{ab}}} = \max_{\substack{x\in X,\\ \mu,\alpha,\beta,\gamma\geqslant 0}} \ \mu \tag{3.10}$$

$$\text{s.t. } (b_0 + \sum_{j\in J} b_j x_j)\mu + \Gamma\alpha + \sum_{j\in J}\beta_j + \sum_{j\in J}\gamma_j \leqslant a_0 + \sum_{j\in J} a_j x_j$$

$$\alpha + \beta_j \geqslant d_j^a x_j, \quad \alpha + \gamma_j \geqslant d_j^b x_j \mu \qquad \forall j \in J.$$

Proof. Let u and v be the indicator variables of the sets $S(\widetilde{a})$ and $S(\widetilde{b})$, respectively. Note that RFP$[\mathcal{U}^{\overline{ab}}]$ can be written as

$$Z^\star_{\mathcal{U}^{\overline{ab}}} = \max_{x\in X} \min_{u,v\in\mathbb{R}^n} \frac{a_0 + \sum_{j\in J} a_j x_j - \sum_{j\in J} d_j^a x_j u_j}{b_0 + \sum_{j\in J} b_j x_j + \sum_{j\in J} d_j^b x_j v_j} \tag{3.11a}$$

$$\text{s.t. } \sum_{j\in J} u_j + \sum_{j\in J} v_j \leqslant \Gamma \tag{3.11b}$$

$$0 \leqslant u_j \leqslant 1, \quad 0 \leqslant v_j \leqslant 1, \qquad \forall j \in J. \tag{3.11c}$$

Observe that we relaxed the binary constraints $u_j \in \mathbb{B}$ and $v_j \in \mathbb{B}$ to convex bound constraints. Since the inner minimization problem is quasi-concave for any $x \in X$ [31], the nonlinear problem has an optimal solution that is an extreme point of the polytope induced by (3.11b)–(3.11c); in particular, there exists an optimal binary solution.

We now reformulate the inner minimization problem using the transformation proposed in [25]: letting $y = 1/\big(b_0 + \sum_{j\in J} b_j x_j + \sum_{j\in J} d_j^b x_j v_j\big)$, $z_j^u = u_j y$, and $z_j^v = v_j y$ for all $j \in J$, we can write (3.11a)–(3.11c) as

$$Z^\star_{\mathcal{U}^{\overline{ab}}} = \max_{x\in X} \min_{z^u, z^v, y} \; (a_0 + \sum_{j\in J} a_j x_j) y - \sum_{j\in J} d_j^a x_j z_j^u \tag{3.12a}$$

$$\text{s.t. } (b_0 + \sum_{j\in J} b_j x_j) y + \sum_{j\in J} d_j^b x_j z_j^v = 1 \qquad (\mu) \tag{3.12b}$$

$$\sum_{j\in J} z_j^u + \sum_{j\in J} z_j^v \leqslant \Gamma y \qquad (\alpha) \tag{3.12c}$$

$$0 \leqslant z_j^u \leqslant y \qquad \forall j \in J \quad (\beta_j) \tag{3.12d}$$

$$0 \leqslant z_j^v \leqslant y \qquad \forall j \in J. \quad (\gamma_j) \tag{3.12e}$$

It is seen that for any fixed $x \in X$, the inner minimization problem is an LP. Thus, using standard LP duality, we obtain formulation (3.10) where μ, α, β_j, and γ_j are corresponding dual variables to constraints (3.12b) to (3.12e). □

Proposition 7. *Problem* RFP[$\mathcal{U}^{\overline{ab}}_{=}$] *is equivalent to*

$$Z^\star_{\mathcal{U}^{\overline{ab}}_{=}} = \max_{\substack{x\in X,\\ \mu,\alpha,\beta\geqslant 0}} \; \mu \tag{3.13}$$

$$\text{s.t. } (b_0 + \sum_{j\in J} b_j x_j)\mu + \Gamma\alpha + \sum_{j\in J} \beta_j \leqslant a_0 + \sum_{j\in J} a_j x_j$$

$$\alpha + \beta_j \geqslant d_j^a x_j + d_j^b x_j \mu \qquad \forall j \in J.$$

Proof. Let u be the indicator variables of the sets $S(\widetilde{a}) = S(\widetilde{b})$. Note that RFP[$\mathcal{U}^{\overline{ab}}_{=}$] can be written as

$$Z^\star_{\mathcal{U}^{\overline{ab}}_{=}} = \max_{x\in X} \min_{u\in\mathbb{R}^n} \frac{a_0 + \sum_{j\in J} a_j x_j - \sum_{j\in J} d_j^a x_j u_j}{b_0 + \sum_{j\in J} b_j x_j + \sum_{j\in J} d_j^b x_j u_j}$$

$$\text{s.t. } \sum_{j\in J} u_j \leqslant \Gamma, \; 0 \leqslant u_j \leqslant 1 \qquad \forall j \in J.$$

Using the Charnes and Cooper [25] transformation as in the proof of Proposition 6, we find the equivalent formulation

$$Z^\star_{\mathcal{U}^{\overline{ab}}_{=}} = \max_{x\in X} \min_{z,y} \; (a_0 + \sum_{j\in J} a_j x_j) y - \sum_{j\in J} d^a_j x_j z_j$$

$$\text{s.t.} (b_0 + \sum_{j\in J} b_j x_j) y + \sum_{j\in J} d^b_j x_j z_j = 1 \qquad (\mu)$$

$$\sum_{j\in J} z_j \leqslant \Gamma y \qquad (\alpha)$$

$$0 \leqslant z_j \leqslant y \qquad \forall j \in J. \quad (\beta_j)$$

Using the standard LP duality for the inner minimization problem, we obtain formulation (3.13). □

Proposition 8. *Problem* RFP$[\mathcal{U}^{\overline{ab}}_{\propto}]$ *is equivalent to*

$$Z^\star_{\mathcal{U}^{\overline{ab}}_{\propto}} = \max_{\substack{x\in X,\\ \mu,\alpha,\beta,\gamma\geqslant 0}} \; \mu \tag{3.14}$$

$$\text{s.t.} \; (b_0 + \sum_{j\in J} b_j x_j)\mu + \Gamma\alpha + \sum_{j\in J}\beta_j + \sum_{j\in J}\gamma_j \leqslant a_0 + \sum_{j\in J} a_j x_j$$

$$\alpha + \beta_j \geqslant -d^a_j x_j + d^b_j x_j \mu, \quad \alpha + \gamma_j \geqslant d^a_j x_j - d^b_j x_j \mu \qquad \forall j \in J.$$

Proof. Let w be the indicator variables of the sets $S(\widetilde{a}) = S(\widetilde{b})$. To model the proportionality conditions, i.e., $\frac{a_j - \widetilde{a}_j}{d^a_j} = \frac{b_j - \widetilde{b}_j}{d^b_j} \in [-1,1]$ for all $j \in J$, we introduce additional continuous variables $\eta \in [-1,1]^n$, and write RFP$[\mathcal{U}^{\overline{ab}}_{\propto}]$ as

$$Z^\star_{\mathcal{U}^{\overline{ab}}_{\propto}} = \max_{x\in X} \min_{w,\eta} \frac{a_0 + \sum_{j\in J} a_j x_j + \sum_{j\in J} d^a_j x_j \eta_j}{b_0 + \sum_{j\in J} b_j x_j + \sum_{j\in J} d^b_j x_j \eta_j}$$

$$\text{s.t.} \; \sum_{j\in J} w_j \leqslant \Gamma$$

$$-w_j \leqslant \eta_j \leqslant w_j, \; w_j \in \{0,1\} \qquad \forall j \in J.$$

Since the inner minimization problem is quasi-concave, it follows that $\eta_j \in \{-w_j, w_j\}$ in an optimal solution. Letting $u_j = 1$ if $\eta_j = w_j > 0$ and 0 otherwise, $v_j = 1$ if $\eta_j = w_j < 0$ and 0 otherwise, we can rewrite RFP$[\mathcal{U}^{\overline{ab}}_{\propto}]$ as

$$Z^{\star}_{\mathcal{U}^{\overline{ab}}_{\propto}} = \max_{x \in X} \min_{u,v \in [0,1]^n} \frac{a_0 + \sum_{j \in J} a_j x_j + \sum_{j \in J} d^a_j x_j u_j - \sum_{j \in J} d^a_j x_j v_j}{b_0 + \sum_{j \in J} b_j x_j + \sum_{j \in J} d^b_j x_j u_j - \sum_{j \in J} d^b_j x_j v_j}$$
$$\text{s.t.} \quad \sum_{j \in J} u_j + \sum_{j \in J} v_j \leqslant \Gamma.$$

Then using the Charnes and Cooper [25] transformation and linear programming duality as in the proofs of Propositions 6 and 7, we obtain formulation (3.14). □

Example 2. Consider a trivariate ($n = 3$) single-ratio RFP[$\mathcal{U}^{\overline{ab}}_{\propto}$]: $Z^{\star}_{\mathcal{U}^{\overline{ab}}_{\propto}} = \frac{a_0 + \widetilde{a}_1 x_1 + \widetilde{a}_2 x_2 + \widetilde{a}_3 x_3}{b_0 + \widetilde{b}_1 x_1 + \widetilde{b}_2 x_2 + \widetilde{b}_3 x_3}$, wherein $a_0 = 6$, $\widetilde{a}_1 \in [-3, 13]$, $\widetilde{a}_2 \in [1, 31]$, $\widetilde{a}_3 \in [1, 5]$, and $b_0 = 3$, $\widetilde{b}_1 \in [0, 4]$, $\widetilde{b}_2 \in [0, 16]$, $\widetilde{b}_3 \in [1, 3]$ for $\Gamma = 2$. Thus, the nominal values are: $a_1 = 5, a_2 = 16, a_3 = 3, b_1 = 2, b_2 = 8, b_3 = 2$, and the deviation values are: $d^a_1 = 8, d^a_2 = 15, d^a_3 = 2, d^b_1 = 2, d^b_2 = 8, d^b_3 = 1$.

Then by Proposition 8, the equivalent reformulation of this RFP[$\mathcal{U}^{\overline{ab}}_{\propto}$] is given by

$$\begin{aligned}
Z^{\star}_{\mathcal{U}^{\overline{ab}}_{\propto}} = \max_{\substack{x \in X, \\ \mu, \alpha, \beta, \gamma \geqslant 0}} \quad & \mu \\
\text{s.t.} \quad & (3 + 2x_1 + 8x_2 + 2x_3)\mu + 2\alpha + \sum_{j \in \{1,2,3\}} \beta_j + \sum_{j \in \{1,2,3\}} \gamma_j \leqslant 6 + 5x_1 + 16x_2 + 3x_3 \\
& \alpha + \beta_1 \geqslant -8x_1 + 2x_1\mu, \qquad \alpha + \gamma_1 \geqslant 8x_1 - 2x_1\mu \\
& \alpha + \beta_2 \geqslant -15x_2 + 8x_2\mu, \qquad \alpha + \gamma_2 \geqslant 15x_2 - 8x_2\mu \\
& \alpha + \beta_3 \geqslant -2x_3 + x_3\mu, \qquad \alpha + \gamma_3 \geqslant 2x_3 - x_3\mu.
\end{aligned}$$
□

Based on Propositions 6, 7, and 8 we see that, in all cases, single-ratio RFP under the joint uncertainty sets $\mathcal{U}^{\overline{ab}}$, $\mathcal{U}^{\overline{ab}}_{=}$, and $\mathcal{U}^{\overline{ab}}_{\propto}$ can be formulated as

$$\max_{\substack{x \in X, \\ \mu, \alpha, \beta, \gamma \geqslant 0}} \quad \mu \tag{3.15a}$$
$$\text{s.t.} \; (b_0 + \sum_{j \in J} b_j x_j)\mu + \Gamma\alpha + \sum_{j \in J} \beta_j + \sum_{j \in J} \gamma_j \leqslant a_0 + \sum_{j \in J} a_j x_j \tag{3.15b}$$
$$\alpha + \beta_j \geqslant (d^1_j + d^2_j \mu) x_j, \quad \alpha + \gamma_j \geqslant (d^3_j + d^4_j \mu) x_j \qquad \forall j \in J, \tag{3.15c}$$

for some $d^1, d^2, d^3, d^4 \in \mathbb{Z}^n$, where $d^1_j \cdot d^3_j$ and $d^2_j \cdot d^4_j \leqslant 0$ for all $j \in J$. In particular, if $d^1_j = d^a_j, d^2_j = d^3_j = 0$, and $d^4_j = d^b_j$ for all $j \in J$, then problem (3.15) is equivalent to the reformulation of RFP[$\mathcal{U}^{\overline{ab}}$] given by (3.10). Similarly, letting $d^1_j = d^a_j, d^2_j = d^b_j, d^3_j = d^4_j = 0$

and $d_j^1 = -d_j^3 = -d_j^a, d_j^2 = -d_j^4 = d_j^b$ for all $j \in J$ in (3.15), lead to equivalent reformulation of $\text{RFP}[\mathcal{U}_{=}^{\overline{ab}}]$ and $\text{RFP}[\mathcal{U}_{\propto}^{\overline{ab}}]$, respectively, provided in (3.13) and (3.14).

Problem (3.15) is a mixed-integer nonlinear program. Note that for a fixed value of μ, problem (3.15) reduces to an MILP feasibility problem or equivalently checking whether the following MILP

$$\psi(\mu) = \min_{x \in X, \alpha, \beta, \gamma} \left\{ (b_0 + \sum_{j \in J} b_j x_j)\mu + \Gamma\alpha + \sum_{j \in J} \beta_j + \sum_{j \in J} \gamma_j - (a_0 + \sum_{j \in J} a_j x_j) \;\Big|\; (3.15c) \right\} \tag{3.16}$$

has a non-positive optimal objective function value (i.e., $\psi(\mu) \leqslant 0$). Proposition 9 below shows that $\psi(\mu)$ is a monotone function of μ. Thus, we can solve (3.15) by applying the binary-search algorithm on μ, where at each iteration of the algorithm we solve (3.16) for a fixed value of μ. That is if $\psi(\mu) > 0$ we must decrease μ, otherwise, we can increase μ.

Proposition 9. *For given vectors d^1, d^2, d^3, and d^4 such that $d_j^2 \cdot d_j^4 \leqslant 0$ and $|d_j^2|, |d_j^4| \leqslant d_j^b$ for all $j \in J$, if $\psi(\mu) \leqslant 0$ for a fixed $\mu \geqslant 0$, then $\psi(\mu') \leqslant 0$ for any $0 \leqslant \mu' < \mu$.*

Proof. For fixed $\mu \geqslant 0$, let $(\alpha, \beta, \gamma, x)$ denote a feasible solution of (3.16) for which the objective function value of (3.16) is non-positive. Then we show that for $\mu' = \mu - \epsilon$, $\epsilon > 0$, there exist $\beta', \gamma' \geqslant 0$ such that $(\alpha, \beta', \gamma', x)$ is a feasible solution of (3.16) with non-positive objective function value. Toward this goal, define $J^2 = \{j \in J \mid d_j^2 < 0\}$ and $J^4 = \{j \in J \mid d_j^4 < 0\}$; note that $J^2 \cap J^4 = \varnothing$ since $d_j^2 \cdot d_j^4 \leqslant 0$ for all $j \in J$. Then let $\beta'_j = \beta_j$ for $j \in J \smallsetminus J^2$ and $\beta'_j = \beta_j - \epsilon d_j^2 x_j \geqslant 0$ for $j \in J^2$. Similarly, let $\gamma'_j = \gamma_j$ for $j \in J \smallsetminus J^4$ and $\gamma'_j = \gamma_j - \epsilon d_j^4 x_j \geqslant 0$ for $j \in J^4$. Hence, $(\mu', \alpha, \beta', \gamma', x)$ satisfies the constraints of (3.15c).

Next, we show that for $(\mu', \alpha, \beta', \gamma', x)$ the objective function value of (3.16) is non-positive.

$$\begin{aligned}
&(b_0 + \sum_{j \in J} b_j x_j)\mu' + \Gamma\alpha + \sum_{j \in J} \beta'_j + \sum_{j \in J} \gamma'_j - (a_0 + \sum_{j \in J} a_j x_j) \\
&= (b_0 + \sum_{j \in J} b_j x_j)(\mu - \epsilon) + \Gamma\alpha + \sum_{j \in J \smallsetminus J^2} \beta_j + \sum_{j \in J^2} (\beta_j - \epsilon d_j^2 x_j) \\
&\qquad + \sum_{j \in J \smallsetminus J^4} \gamma_j + \sum_{j \in J^4} (\gamma_j - \epsilon d_j^4 x_j) - (a_0 + \sum_{j \in J} a_j x_j) \\
&= (b_0 + \sum_{j \in J} b_j x_j)\mu + \Gamma\alpha + \sum_{j \in J} \beta_j + \sum_{j \in J} \gamma_j - (a_0 + \sum_{j \in J} a_j x_j) \\
&\qquad - \epsilon(b_0 + \sum_{j \in J} b_j x_j + \sum_{j \in J^2} d_j^2 x_j + \sum_{j \in J^4} d_j^4 x_j) \leqslant 0.
\end{aligned}$$

The last inequality holds because the objective function value of (3.16) is non-positive for $(\mu,\alpha,\beta,\gamma,x)$; moreover, since $J^2 \cap J^4 = \varnothing$ and $|d_j^2|,|d_j^4| \leqslant d_j^b$, for all $j \in J$, by Assumption 1 we have $(b_0 + \sum_{j\in J} b_j x_j + \sum_{j\in J^2} d_j^2 x_j + \sum_{j\in J^4} d_j^4 x_j) > 0$. □

In order to solve (3.16) efficiently at each iteration of the binary-search algorithm, we further simplify it by using an argument similar to the one used for proving Proposition 5.

Proposition 10. *Problem* (3.16) *can be reformulated as*

$$\psi(\mu) = \min_{x\in X,\alpha\in\mathcal{F}} \Big\{ (b_0 + \sum_{j\in J} b_j x_j)\mu + \Gamma\alpha + \sum_{j\in J}\left(d_j^1 + d_j^2\mu - \alpha\right)^+ x_j + \sum_{j\in J}\left(d_j^3 + d_j^4\mu - \alpha\right)^+ x_j - (a_0 + \sum_{j\in J} a_j x_j)\Big\}, \tag{3.17}$$

where

$$\mathcal{F} = \Big\{0, (d_1^1 + d_1^2\mu)^+, (d_2^1 + d_2^2\mu)^+, \ldots, (d_n^1 + d_n^2\mu)^+, (d_1^3 + d_1^4\mu)^+, (d_2^3 + d_2^4\mu)^+, \ldots, (d_n^3 + d_n^4\mu)^+\Big\}.$$

Proof. In an optimal solution of (3.16), we have that, for all $j \in J$, $\beta_j^\star = \left((d_j^1 + d_j^2\mu)x_j^\star - \alpha^\star\right)^+ = (d_j^1 + d_j^2\mu - \alpha^\star)^+ x_j^\star$ and $\gamma_j^\star = \left((d_j^3 + d_j^4\mu)x_j^\star - \alpha^\star\right)^+ = \left(d_j^3 + d_j^4\mu - \alpha^\star\right)^+ x_j^\star$. Thus, (3.16) reduces to

$$\psi(\mu) = \min_{\substack{x\in X,\\ \alpha\geqslant 0}} (b_0 + \sum_{j\in J} b_j x_j)\mu + \Gamma\alpha + \sum_{j\in J}(d_j^1 + d_j^2\mu - \alpha)^+ x_j + \sum_{j\in J}(d_j^3 + d_j^4\mu - \alpha)^+ x_j - (a_0 + \sum_{j\in J} a_j x_j).$$

Additionally, similar to the proof of Proposition 5 observe that if $\alpha^\star > 0$, $\alpha^\star \neq d_j^1 + d_j^2\mu$ and $\alpha^\star \neq d_j^3 + d_j^4\mu$ for all $j \in J$, then it can be verified that either $\alpha^\star + \epsilon$ or $\alpha^\star - \epsilon$ is also feasible for sufficiently small $\epsilon > 0$. Thus, we may assume without loss of generality that $\alpha^\star \in \{0\} \cup \{(d_j^1 + d_j^2\mu)^+\}_{j\in J} \cup \{(d_j^3 + d_j^4\mu)^+\}_{j\in J}$, which completes the proof. □

Example 3. According to Proposition 10, the corresponding formulation (3.17) for RFP$[\mathcal{U}_\propto^{\overline{ab}}]$ in Example 2 is

$$\begin{aligned}
&\psi(\mu) = \\
&\min_{x\in X,\alpha\in\mathcal{F}} \Big\{ (3 + 2x_1 + 8x_2 + 2x_3)\mu + 2\alpha \\
&\quad + (-8 + 2\mu - \alpha)^+ x_1 + (-15 + 8\mu - \alpha)^+ x_2 + (-2 + \mu - \alpha)^+ x_3 \\
&\quad + (8 - 2\mu - \alpha)^+ x_1 + (15 - 8\mu - \alpha)^+ x_2 + (2 - \mu - \alpha)^+ x_3 - (6 + 5x_1 + 16x_2 + 3x_3)\Big\},
\end{aligned}$$

where $\mathcal{F} = \Big\{0, (-8 + 2\mu)^+, (-15 + 8\mu)^+, (-2 + \mu)^+, (8 - 2\mu)^+, (15 - 8\mu)^+, (2 - \mu)^+\Big\}$. □

In the following, we focus our efforts on obtaining the optimal objective function value of (3.17). To this end, define $T = \{1, 2, \dots, |\mathcal{F}|\}$, $|T| \leqslant 2n + 1$, and for each $t \in T$ define binary-linear problem

$$\psi_t(\mu) = \min_{x \in X} g_t(x, \mu) \tag{3.18}$$

where

$$g_t(x,\mu) = (b_0 + \sum_{j \in J} b_j x_j)\mu + \Gamma\left(\bar{c}_t + \bar{d}_t\mu\right)^+ + \sum_{j \in J}\left(d_j^1 + d_j^2\mu - \left(\bar{c}_t + \bar{d}_t\mu\right)^+\right)^+ x_j$$
$$+ \sum_{j \in J}\left(d_j^3 + d_j^4\mu - \left(\bar{c}_t + \bar{d}_t\mu\right)^+\right)^+ x_j - a_0 - \sum_{j \in J} a_j x_j,$$

and $(\bar{c}_t, \bar{d}_t) \in \{(0,0)\} \cup \{(d_j^1, d_j^2)\}_{j \in J} \cup \{(d_j^3, d_j^4)\}_{j \in J}$.

Evidently, $\psi(\mu) = \min_{t \in T} \psi_t(\mu)$. Thus, for μ fixed, checking whether $\psi(\mu) \leqslant 0$ can be done by verifying whether there exists $t \in T$ such that $\psi_t(\mu) \leqslant 0$. Thereby, in the following result we conclude that problem (3.15) can be solved efficiently using the binary-search method.

Proposition 11. *Problem* (3.15) *can be solved with* $O\big(n \log(U/\epsilon)\big)$ *calls to an oracle for* (3.18)*, where* $U = |a_0| + \sum_{j \in J} |a_j|$ *and* $\epsilon > 0$ *is a precision parameter.*

Proof. The binary search requires $O\big(\log(\frac{U}{\epsilon})\big)$ iterations and each iteration requires solving at most $|\mathcal{F}| = |T| = 2n + 1$ problems of the form (3.18). Moreover, let $\tau(n)$ denote the complexity of solving binary-linear problem (3.18). Then the binary-search algorithm to solve problem (3.15) has the worst-case complexity $O\big(n \log(U/\epsilon)\tau(n)\big)$. □

As a direct consequence of Propositions 9 to 11, we get the main result of this subsection, i.e.,

Theorem 2. *Single-ratio case of* RFP[$\mathcal{U}^{\overline{ab}}$], RFP[$\mathcal{U}^{\overline{ab}}_{=}$] *and* RFP[$\mathcal{U}^{\overline{ab}}_{\propto}$] *can be solved in* $O\big(n \log(U/\epsilon)\tau(n)\big)$*, where* $\tau(n)$ *is the complexity of solving problem* (3.18)*. In particular, if linear optimization over* X *is polynomial-time solvable, then so is single-ratio* RFP *under the joint uncertainty sets.*

Notably, when $X = \mathbb{B}^n$ the complexity of solving problem (3.18) is $O(n)$, i.e., $\tau(n) = n$, resulting in the overall complexity $O\big(n^2 \log(U/\epsilon)\big)$ to solve RFP[$\mathcal{U}^{\overline{ab}}$], RFP[$\mathcal{U}^{\overline{ab}}_{=}$] and RFP[$\mathcal{U}^{\overline{ab}}_{\propto}$]. Additionally, if $X = \{x \in \mathbb{B}^n \mid \sum_{j \in J} x_j \leqslant k\}$ or $X = \{x \in \mathbb{B}^n \mid \sum_{j \in J} x_j = k\}$ we have $\tau(n) = n \log(n)$, resulting in the overall complexity $O\big(n^2 \log(n) \log(U/\epsilon)\big)$. Therefore,

Corollary 1. *The unconstrained and cardinality constrained single-ratio* RFP[$\mathcal{U}$] *under joint uncertainty sets* $\mathcal{U}^{\overline{ab}}$, $\mathcal{U}^{\overline{ab}}_{=}$ *and* $\mathcal{U}^{\overline{ab}}_{\propto}$ *can be solved in polynomial time.*

It is worth mentioning that the cardinality-constrained ($X = \{x \in \mathbb{B}^n \mid \sum_{j \in J} x_j \leqslant k\}$) single-ratio assortment problem (3.3) when customer preferences (ρ_j) are subject to rectangular uncertainty $\mathcal{U} = \prod_{j=0}^{n}[l_j, u_j] \subset \mathbb{R}^{n+1}_{++}$, where ℓ_j and u_j are lower and upper bounds on ρ_j, can be solved in $O(n^2)$, see [81]. This problem is a special case of RFP[$\mathcal{U}^{\overline{ab}}_{\propto}$] when $\Gamma = n$, and $\widetilde{a}_j, \widetilde{b}_j > 0$. However, the aforementioned result cannot be extended, e.g., when revenues (r_j) are uncertain or, more importantly, for generally structured single-ratio RFP[$\mathcal{U}$] (such as other choice models) under other types of the budgeted uncertainty sets or (weaker) Assumption 1. We conclude the discussion on single-ratio RFP[$\mathcal{U}$] with the following remarks.

Remark 7. The solutions methods outlined in this section are particularly efficient for unconstrained problems. Additionally, they are useful when there exist specialized algorithms to solve the corresponding constrained linear binary problem, e.g., those that exploit the constraint structure of the underlying combinatorial optimization problem. If these algorithms are polynomial time (for example, such as those for the linear assignment, the shortest path and the minimum spanning tree problems, see [2]), then the single-ratio RFP[$\mathcal{U}$] is also polynomial-time solvable. □

Remark 8. In the case of single-ratio RFPs under the disjoint uncertainty set, the approach of Theorem 1 is superior to the binary search approach developed in Section 3.3.2 since the former is strongly polynomial, $O(n^2)$, while the latter involves the binary search algorithm with the number of iterations $O\big(\log(\frac{U}{\epsilon})\big)$. □

3.4 Multiple-ratio RFP[$\mathcal{U}$]

In this section, we present MILP formulations for multiple-ratio RFP[$\mathcal{U}$]. First, for the disjoint uncertainty set, we reformulate RFP[$\mathcal{U}$] as robust linear problems. Then with these reformulations in hand, we adapt the methods of [13] to transform them into MILPs, see Section 3.4.1. For the joint uncertainty sets (except $\mathcal{U}^{\overline{a}}$) we use the results from Section 3.3.2, see

Section 3.4.2.1; for $\mathcal{U}^{\overline{a}}$ we use a same approach provided in Section 3.4.1, see Section 3.4.2.2. Then, in Section 3.4.3 we discuss the sizes (numbers of variables and constraints) of the obtained MILP reformulations. Finally, in Section 3.4.4 we show that the optimal value of the robust formulations provided in this chapter with high probability are not overestimator of the true value of the fractional problems with symmetrical and bounded random coefficients.

3.4.1 Disjoint uncertainty set

For the present discussion, we consider the uncertainty set $\mathcal{U}^{ab}$, and present three MILP reformulations of RFP[$\mathcal{U}^{ab}$]. For the first two formulations presented in Section 3.4.1.1 and Section 3.4.1.2 we exploit the ideas from fractional programming literature, see [54, 99]. The third formulation, presented in Section 3.4.1.3 corresponds to a binary expansion reformulation proposed by [16].

3.4.1.1 Reformulation 1 $\left(\mathbf{MILP}_1[\mathcal{U}^{ab}]\right)$. Note that RFP[$\mathcal{U}^{ab}$] can be written as

$$\max_{x\in X}\ \min_{(\widetilde{a},\widetilde{b})\in\mathcal{U}^{ab}} \sum_{i\in I}\left(a_{i0} + \sum_{j\in J}\widetilde{a}_{ij}x_j\right)\left(\frac{1}{b_{i0} + \sum_{j\in J}\widetilde{b}_{ij}x_j}\right).$$

Using the substitutions $\omega_i \leqslant \frac{1}{b_{i0}+\sum_{j\in J}\widetilde{b}_{ij}x_j}$, for all $\widetilde{b}_i \in \mathcal{U}_i^b$ and $i \in I$, and exploiting the fact that $\mathcal{U}^{ab}$ is disjoint, we find the equivalent formulation

$$\begin{aligned}
\max_{\substack{x\in X,\\ \omega\geqslant 0}}\ \min_{\widetilde{a}\in\mathcal{U}^a}\ & \sum_{i\in I}(a_{i0} + \sum_{j\in J}\widetilde{a}_{ij}x_j)\omega_i \\
\text{s.t. } & (b_{i0} + \sum_{j\in J}\widetilde{b}_{ij}x_j)\omega_i \leqslant 1 \qquad \forall \widetilde{b}_i \in \mathcal{U}_i^b,\ \forall i\in I,
\end{aligned}$$

where $\mathcal{U}^a := \left\{\widetilde{a}\in\mathbb{R}^{m\times n} \mid \widetilde{a}_i \in \mathcal{U}_i^a \text{ for all } i \in I\right\}$. Similarly, defining new variables μ_i such that $\mu_i \leqslant (a_{i0} + \sum_{j\in J}\widetilde{a}_{ij}x_j)\omega_i$ for all $\widetilde{a}_i \in \mathcal{U}_i^a$ and $i \in I$ yields the robust optimization problem

$$(\mathrm{RFP}_1[\mathcal{U}^{ab}]) \qquad \begin{aligned}
\max_{\substack{x\in X,\\ \mu,\omega\geqslant 0}}\ & \sum_{i\in I}\mu_i \\
\text{s.t. } & \mu_i \leqslant (a_{i0} + \sum_{j\in J}\widetilde{a}_{ij}x_j)\omega_i \qquad \forall \widetilde{a}_i \in \mathcal{U}_i^a,\ \forall i\in I \\
& (b_{i0} + \sum_{j\in J}\widetilde{b}_{ij}x_j)\omega_i \leqslant 1 \qquad \forall \widetilde{b}_i \in \mathcal{U}_i^b,\ \forall i\in I.
\end{aligned}$$

Note that the directions of the inequalities ($\leqslant$) rely on the sense of the objective function and Assumption 1. Since $x \in X \subseteq \mathbb{B}^n$, we linearize the bilinear terms $x_j\omega_i$ using standard techniques (e.g., [1, 100]) as follows

$$\Delta_{ij} := \left\{(x_j, \omega_i, z_{ij}) \in \mathbb{B} \times \mathbb{R}^2_+ \mid \omega_i^L x_j \leqslant z_{ij} \leqslant \omega_i^U x_j, \omega_i + \omega_i^U(x_j - 1) \leqslant z_{ij} \leqslant \omega_i + \omega_i^L(x_j - 1)\right\},$$

where ω_i^U and ω_i^L are an upper bound and a lower bound on ω_i, respectively, and note that $(x_j, w_i, z_{ij}) \in \Delta_{ij} \Leftrightarrow z_{ij} = w_i x_j$. Hence, $\text{RFP}_1[\mathcal{U}^{ab}]$ is equivalent to the robust linear problem

$$\begin{aligned}
\max_{\substack{x \in X \\ \omega,\mu,z \geqslant 0}} \; & \sum_{i \in I} \mu_i && (3.19)\\
\text{s.t. } & \mu_i \leqslant a_{i0}\omega_i + \sum_{j \in J} \widetilde{a}_{ij} z_{ij} && \forall \widetilde{a}_i \in \mathcal{U}_i^a,\ \forall i \in I\\
& b_{i0}\omega_i + \sum_{j \in J} \widetilde{b}_{ij} z_{ij} \leqslant 1 && \forall \widetilde{b}_i \in \mathcal{U}_i^b,\ \forall i \in I\\
& (x_j, \omega_i, z_{ij}) \in \Delta_{ij} && \forall i \in I, j \in J.
\end{aligned}$$

Following the approach of [13], the robust linear problem (3.19) can be transformed into an MILP reformulation of $\text{RFP}[\mathcal{U}^{ab}]$ as follows.

$$\begin{aligned}
(\text{MILP}_1[\mathcal{U}^{ab}]) \;\; \max \; & \sum_{i \in I} \mu_i \\
\text{s.t. } & \mu_i - \sum_{j \in J} a_{ij} z_{ij} + \sum_{j \in J} \beta_{ij} + \Gamma_i^a \alpha_i \leqslant a_{i0}\omega_i && \forall i \in I\\
& b_{i0}\omega_i + \sum_{j \in J} b_{ij} z_{ij} + \sum_{j \in J} \gamma_{ij} + \Gamma_i^b \lambda_i \leqslant 1 && \forall i \in I\\
& \alpha_i + \beta_{ij} \geqslant d_{ij}^a z_{ij} && \forall i \in I, \forall j \in J\\
& \lambda_i + \gamma_{ij} \geqslant d_{ij}^b z_{ij} && \forall i \in I, \forall j \in J\\
& x \in X,\ (x_j, \omega_i, z_{ij}) \in \Delta_{ij},\ \beta_{ij}, \gamma_{ij}, \alpha_i, \lambda_i, \mu_i \geqslant 0 && \forall i \in I, \forall j \in J.
\end{aligned}$$

3.4.1.2 Reformulation 2 $\left(\mathbf{MILP}_2[\mathcal{U}^{ab}]\right)$.

As an alternative to the approach of Section 3.4.1.1, one could instead replace each ratio with an auxiliary variable. Let $\mu_i \leqslant \frac{a_{i0}+\sum_{j\in J}\widetilde{a}_{ij}x_j}{b_{i0}+\sum_{j\in J}\widetilde{b}_{ij}x_j}$ for all $i \in I$, $(\widetilde{a}_i, \widetilde{b}_i) \in \mathcal{U}_i^a \times \mathcal{U}_i^b$. Then we can write RFP$[\mathcal{U}^{ab}]$ as

$$(\mathrm{RFP}_2[\mathcal{U}^{ab}]) \qquad \max_{\substack{x\in X,\\ \mu\geqslant 0}} \sum_{i\in I}\mu_i$$

$$\text{s.t. } (b_{i0} + \sum_{j\in J}\widetilde{b}_{ij}x_j)\mu_i \leqslant a_{i0} + \sum_{j\in J}\widetilde{a}_{ij}x_j \qquad \forall \widetilde{a}_i \in \mathcal{U}_i^a,\ \forall \widetilde{b}_i \in \mathcal{U}_i^b, \forall i \in I.$$

Finally, after linearization of $x_j\mu_i$ using a variant of the set Δ_{ij} and applying the transformation of a robust linear problem to an MILP similar to the one used in Section 3.4.1.1, we find the MILP reformulation of RFP$_2[\mathcal{U}^{ab}]$.

$$(\mathrm{MILP}_2[\mathcal{U}^{ab}]) \qquad \max \sum_{i\in I}\mu_i$$

$$\begin{aligned}
\text{s.t. } & b_{i0}\mu_i - \sum_{j\in J}a_{ij}x_j + \sum_{j\in J}b_{ij}z_{ij} + \\
& \Gamma_i^a\alpha_i + \Gamma_i^b\lambda_i + \sum_{j\in J}\beta_{ij} + \sum_{j\in J}\gamma_{ij} \leqslant a_{i0} && \forall i \in I \\
& \alpha_i + \beta_{ij} \geqslant d_{ij}^a x_j && \forall i \in I, \forall j \in J \\
& \lambda_i + \gamma_{ij} \geqslant d_{ij}^b z_{ij} && \forall i \in I, \forall j \in J \\
& x \in X, (x_j, \mu_i, z_{ij}) \in \Delta_{ij}, \beta_{ij}, \gamma_{ij}, \alpha_i, \lambda_i, \mu_i \geqslant 0 && \forall i \in I, \forall j \in J.
\end{aligned}$$

3.4.1.3 Binary-expansion reformulation $\left(\mathbf{MILP}_2^{\log}[\mathcal{U}^{ab}]\right)$.

The third considered formulation uses a base-2 expansion [16] to reduce the number of bilinear terms that require linearization. In the context of RFP, we employ this idea to reformulate RFP$_2[\mathcal{U}^{ab}]$.

Observe that for any $x \in X$ and worst-case realization $\widetilde{b}_i \in \mathcal{U}_i^b$, the term $\sum_{j\in J}\widetilde{b}_{ij}x_j$ is integer since the data are integral (Assumption 2). To ascertain the (logarithmic) number of additional variables needed, let $\max^r(H_i)$ return the r-th largest element in the set $H_i = \{d_{ij}^b \mid j \in J\}$. Then for all $i \in I$, we define π_i as follows

$$\pi_i := \left\lfloor \log_2\left(\sum_{j\in J}|b_{ij}| + \sum_{r\leqslant \Gamma_i^b}\max{}^r(H_i)\right)\right\rfloor + 1. \tag{3.20}$$

We then define the binarization variables $w_{ik} \in \mathbb{B}$ for all $k \in K_i := \{1, 2, \ldots, \pi_i\},\ i \in I$. We also define $\bar{B}_i := \sum_{j \in J, b_{ij} < 0} |b_{ij}|$. Observe that $\sum_{j \in J} \widetilde{b}_{ij} x_j + \bar{B}_i \geqslant 0$ for any $x \in X$ and $\widetilde{b}_i \in \mathcal{U}_i^b$. Replacing the terms $\sum_{j \in J} \widetilde{b}_{ij} x_j$ with $-\bar{B}_i + \sum_{k=1}^{\pi_i} 2^{k-1} w_{ik}$ for all $i \in I$ in $\text{RFP}_2[\mathcal{U}^{ab}]$, yields

$$
\begin{aligned}
(\text{RFP}_2^{\log}[\mathcal{U}^{ab}]) \quad & \max \sum_{i \in I} \mu_i \\
\text{s.t. } & (b_{i0} - \bar{B}_i + \sum_{k \in K_i} 2^{k-1} w_{ik}) \mu_i \leqslant a_{i0} + \sum_{j \in J} \widetilde{a}_{ij} x_j && \forall \widetilde{a}_i \in \mathcal{U}_i^a,\ \forall i \in I \\
& \sum_{j \in J} \widetilde{b}_{ij} x_j + \bar{B}_i \leqslant \sum_{k \in K_i} 2^{k-1} w_{ik} && \forall \widetilde{b}_i \in \mathcal{U}_i^b,\ \forall i \in I \\
& x \in X, w_{ik} \in \mathbb{B}, \mu_i \geqslant 0 && \forall k \in K_i, \forall i \in I.
\end{aligned}
$$

Let $z_{ik} = w_{ik} \mu_i$. By using a variant of the set Δ_{ij} in model $\text{RFP}_2^{\log}[\mathcal{U}^{ab}]$ and applying the transformation of a robust linear problem to an MILP similar to the one used in Section 3.4.1.1, $\text{RFP}_2^{\log}[\mathcal{U}^{ab}]$ can be reformulated as the following MILP.

$$
\begin{aligned}
(\text{MILP}_2^{\log}[\mathcal{U}^{ab}]) \quad & \max \sum_{i \in I} \mu_i \\
\text{s.t. } & (b_{i0} - \bar{B}_i) \mu_i + \sum_{k \in K_i} 2^{k-1} z_{ik} - \sum_{j \in J} a_{ij} x_j + \sum_{j \in J} \beta_{ij} + \Gamma_i^a \alpha_i \leqslant a_{i0} && \forall i \in I \\
& - \sum_{k \in K_i} 2^{k-1} w_{ik} + \sum_{j \in J} b_{ij} x_j + \bar{B}_i + \sum_{j \in J} \gamma_{ij} + \Gamma_i^b \lambda_i \leqslant 0 && \forall i \in I \\
& \alpha_i + \beta_{ij} \geqslant d_{ij}^a x_j && \forall i \in I, \forall j \in J \\
& \lambda_i + \gamma_{ij} \geqslant d_{ij} x_j && \forall i \in I, \forall j \in J \\
& x \in X,\ \beta_{ij}, \gamma_{ij}, \alpha_i, \lambda_i, \mu_i \geqslant 0 && \forall i \in I, \forall j \in J \\
& w_{ik} \in \mathbb{B},\ (w_{ik}, \mu_i, z_{ik}) \in \Delta_{ij} && \forall i \in I, \forall k \in K_i.
\end{aligned}
$$

Remark 9. It is also possible to develop a binary-expansion reformulation of $\text{RFP}_1[\mathcal{U}^{ab}]$. However, based on our experiments such a formulation performs poorly in computations; also, refer to [16] for an analogous comparison regarding deterministic FP. Hence, we omit this formulation for brevity. □

3.4.2 Joint uncertainty sets

We now present MILP formulations of RFP[$\mathcal{U}$] under the joint uncertainty sets $\mathcal{U} \in \{\mathcal{U}^{\overline{ab}}, \mathcal{U}^{\overline{ab}}_{=}, \mathcal{U}^{\overline{ab}}_{\propto}, \mathcal{U}^{\overline{a}}\}$. Toward this goal, we use the results of Section 3.3.2 to develop MILPs for multiple-ratio RFP[$\mathcal{U}^{\overline{ab}}$], RFP[$\mathcal{U}^{\overline{ab}}_{=}$], and RFP[$\mathcal{U}^{\overline{ab}}_{\propto}$]; see Section 3.4.2.1. For RFP[$\mathcal{U}^{\overline{a}}$] we use a similar approach to the one used in Section 3.4.1.1, see Section 3.4.2.2. Note that, for the joint uncertainty sets we cannot take the advantage of the binary-expansion technique, either due to dependencies in the uncertainty sets, or because it does not reduce the number of bilinear terms for the joint cases.

3.4.2.1 Reformulation for RFP[$\mathcal{U}$] when $\mathcal{U} \in \{\mathcal{U}^{\overline{ab}}, \mathcal{U}^{\overline{ab}}_{=}, \mathcal{U}^{\overline{ab}}_{\propto}\}$. By Propositions 6, 7, and 8 it is verified that multiple-ratio RFP[$\mathcal{U}$] under joint uncertainties $\mathcal{U}^{\overline{ab}}$, $\mathcal{U}^{\overline{ab}}_{=}$, and $\mathcal{U}^{\overline{ab}}_{\propto}$ can be represented as the following problem.

$$\begin{aligned} \max_{\substack{x\in X,\\ \mu,\alpha,\beta,\gamma\geqslant 0}} \quad & \sum_{i\in I} \mu_i && (3.21)\\ \text{s.t. } & (b_{i0} + \sum_{j\in J} b_{ij}x_j)\mu_i + \Gamma_i\alpha_i + \sum_{j\in J}\beta_{ij} + \sum_{j\in J}\gamma_{ij} \leqslant a_{i0} + \sum_{j\in J} a_{ij}x_j && \forall i \in I\\ & \alpha_i + \beta_{ij} \geqslant (d^1_{ij} + d^2_{ij}\mu_i)x_j, \quad \alpha_i + \gamma_{ij} \geqslant (d^3_{ij} + d^4_{ij}\mu_i)x_j && \forall i \in I, \forall j \in J, \end{aligned}$$

for some $d^1, d^2, d^3, d^4 \in \mathbb{Z}^{m\times n}$. By linearizing the bilinear terms $x_j\mu_i$, problem (3.21) can be reformulated as an equivalent MILP.

$$\begin{aligned} \max_{\substack{x\in X,\\ \mu,\alpha,\beta,\gamma\geqslant 0}} \quad & \sum_{i\in I} \mu_i && (3.22)\\ \text{s.t. } & b_{i0}\mu_i - \sum_{j\in J} a_{ij}x_j + \sum_{j\in J} b_{ij}z_{ij} + \Gamma_i\alpha_i + \sum_{j\in J}\beta_{ij} + \sum_{j\in J}\gamma_{ij} \leqslant a_{i0} && \forall i \in I\\ & \alpha_i + \beta_{ij} \geqslant d^1_{ij}x_j + d^2_{ij}z_{ij}, \quad \alpha_i + \gamma_{ij} \geqslant d^3_{ij}x_j + d^4_{ij}z_{ij} && \forall i \in I, \forall j \in J\\ & x \in X, (x_j, \mu_i, z_{ij}) \in \Delta_{ij}, \beta_{ij}, \gamma_{ij}, \alpha_i, \mu_i \geqslant 0 && \forall i \in I, \forall j \in J. \end{aligned}$$

Specifically, if we let $d^1_j = d^a_j, d^2_j = d^3_j = 0$, and $d^4_j = d^b_j$ for all $j \in J$, then problem (3.22) is an equivalent MILP reformulation of RFP[$\mathcal{U}^{\overline{ab}}$] denoted by MILP[$\mathcal{U}^{\overline{ab}}$]. Similarly, letting $d^1_j = d^a_j, d^2_j = d^b_j, d^3_j = d^4_j = 0$ and $d^1_j = -d^3_j = -d^a_j, d^2_j = -d^4_j = d^b_j$ for all $j \in J$ in (3.22), lead to equivalent MILP reformulations of RFP[$\mathcal{U}^{\overline{ab}}_{=}$] and RFP[$\mathcal{U}^{\overline{ab}}_{\propto}$] indicated by MILP[$\mathcal{U}^{\overline{ab}}_{=}$] and

MILP[$\mathcal{U}^{\overline{ab}}_{\propto}$], respectively. Finally, note that in MILP[$\mathcal{U}^{\overline{ab}}_{=}$] since $d_j^3 = d_j^4 = 0$ variable γ_{ij} and constraint $\alpha_i + \gamma_{ij} \geqslant d_{ij}^3 x_j + d_{ij}^4 z_{ij}$ can be removed for all $i \in I, j \in J$; see Table 5 for the size of formulations.

3.4.2.2 Reformulation for RFP[$\mathcal{U}^{\overline{a}}$]. Let ω as in Section 3.4.1.1, define a new variable $\mu \leqslant \sum_{i\in I}(a_{i0} + \sum_{j\in J} \widetilde{a}_{ij} x_j)\omega_i$ for all $\widetilde{a} \in \mathcal{U}^{\overline{a}}$, and write RFP[$\mathcal{U}^{\overline{a}}$] as

$$\begin{aligned} \max_{x\in X, \omega, \mu \geqslant 0} \ & \mu \\ \text{s.t. } & \mu \leqslant \sum_{i\in I} a_{i0}\omega_i + \sum_{i\in I}\sum_{j\in J} \widetilde{a}_{ij} x_j \omega_i && \forall \widetilde{a} \in \mathcal{U}^{\overline{a}} \\ & b_{i0}\omega_i + \sum_{j\in J} b_{ij} x_j \omega_i \leqslant 1 && \forall i \in I. \end{aligned}$$

Letting $z_{ij} = x_j\omega_i$ and u be the indicator variables of set $S_i(\widetilde{a})$, we obtain

$$\begin{aligned} \max_{\substack{x\in X,\\ \mu,\omega \geqslant 0}} \ & \mu \\ \text{s.t. } & \mu - \sum_{i\in I} a_{i0}\omega_i - \sum_{i\in I}\sum_{j\in J} a_{ij} z_{ij} + \max_{u\in V}\left\{ \sum_{j\in J} d_{ij}^a z_{ij} u_{ij} \right\} \leqslant 0 && \forall i \in I \\ & b_{i0}\omega_i + \sum_{j\in J} b_{ij} z_{ij} \leqslant 1 && \forall i \in I \\ & (x_j, \omega_i, z_{ij}) \in \Delta_{ij} && \forall i \in I,\ j \in J, \end{aligned}$$

where V is the polytope defined by the constraints

$$\begin{aligned} & \sum_{i\in I}\sum_{j\in J} u_{ij} \leqslant \Gamma && && (\alpha) \\ & 0 \leqslant u_{ij} \leqslant 1 && \forall i \in I, j \in J. && (\beta_{ij}) \end{aligned}$$

Using LP-duality for the inner maximization problem, we obtain the MILP formulation:

$$\begin{aligned} (\text{MILP}[\mathcal{U}^{\overline{a}}]) \quad \max \ & \mu \\ \text{s.t. } & \mu - \sum_{i\in I} a_{i0}\omega_i - \sum_{i\in I}\sum_{j\in J} a_{ij} z_{ij} + \Gamma\alpha + \sum_{i\in I}\sum_{j\in J} \beta_{ij} \leqslant 0 \\ & b_{i0}\omega_i + \sum_{j\in J} b_{ij} z_{ij} \leqslant 1 && \forall i \in I \\ & \alpha + \beta_{ij} \geqslant d_{ij}^a z_{ij} && \forall i \in I, \forall j \in J \\ & x \in X,\ (x_j, \omega_i, z_{ij}) \in \Delta_{ij},\ \beta_{ij}, \alpha, \mu, \omega_i \geqslant 0 && \forall i \in I, \forall j \in J. \end{aligned}$$

3.4.3 Problems sizes and MILP enhancement $\left(\mathrm{MILP}_{2'}^{\log}[\mathcal{U}^{ab}]\right)$

Table 5 shows the number of variables and constraints for all MILP reformulations provided in Section 3.4.1 and Section 3.4.2. This table also includes data for the well-known MILPs for FP, denoted by FP_1 [54, 92, 100] and FP_2 [92], as well as their respective binary-expansion versions [16], denoted by FP_3 and FP_4.

Later in Section 3.5.2.3 we observe that, among the MILPs developed for the disjoint uncertainty, $\mathrm{MILP}_1[\mathcal{U}^{ab}]$ typically has the best LP relaxation and $\mathrm{MILP}_2^{\log}[\mathcal{U}^{ab}]$ often has the best performance due to a reduced number of variables and constraints - see Table 5. Hence, we enhance $\mathrm{MILP}_2^{\log}[\mathcal{U}^{ab}]$ by adding the valid inequality $\sum_{i\in I}\mu_i \leqslant z_{LP}^{MILP_1[\mathcal{U}^{ab}]}$ to $\mathrm{MILP}_2^{\log}[\mathcal{U}^{ab}]$ where $z_{LP}^{MILP_1[\mathcal{U}^{ab}]}$ is the objective function value of the LP relaxation of $\mathrm{MILP}_1[\mathcal{U}^{ab}]$, and we call the new formulation $\mathrm{MILP}_{2'}^{\log}[\mathcal{U}^{ab}]$. In the deterministic fractional programming, a similar observation is made regarding FP_1 and FP_4 [16]. The new formulation is called $\mathrm{FP}_{4'}$ and we compare its performance versus the performances of the developed MILPs for the disjoint uncertainty in the next section.

3.4.4 Insights on the price of robustness

In robust linear optimization when uncertain coefficients are symmetric, bounded and independent random variables, Bertsimas and Sim [13] provide a probabilistic guarantee for each constraint violation. Next, we exploit their approach to establish somewhat similar results for RFPs under dis/joint uncertainty sets.

Let $x^\star$ and $\mu_i^\star$ denote a robust optimal solution and the robust value of the i-th ratio in $\mathrm{RFP}[\mathcal{U}]$, respectively. By using the binomial distribution

$$B(r,P) = \frac{1}{2^r}\left\{(1-\nu+\lfloor\nu\rfloor)\binom{r}{\lfloor\nu\rfloor} + \sum_{j=\lfloor\nu\rfloor+1}^{r}\binom{r}{j}\right\},$$

for $\nu = (P+r)/2$, and $r, P \in \mathbb{Z}_+$, we show the probability that $\mu_i^\star$ overestimates the true value of the i-th ratio for random variables $\widetilde{a}$ and $\widetilde{b}$ is bounded above.

Table 5: Sizes of the MILPs for nominal problems FP_1 to FP_4, and the robust problems, where n and m are defined as in FP, c is the number of constraints defining X, and π_i is defined as in (3.20). Moreover, $\theta_i^a := \lfloor \log_2(\sum_{j\in J} |a_{ij}|) \rfloor + 1$ and $\theta_i^b := \lfloor \log_2(\sum_{j\in J} |b_{ij}|) \rfloor + 1$.

MILP reformulation	No. of continuous variables	No. of binary variables	No. of linear constraints
Nominal reformulations			
FP_1	$m(n+1)$	n	$m(4n+1)+c$
FP_2	$m(n+1)$	n	$m(4n+1)+c$
FP_3	$m+\sum_{i\in I}(\theta_i^a+\theta_i^b)$	$n+\sum_{i\in I}(\theta_i^a+\theta_i^b)$	$3m+4\sum_{i\in I}(\theta_i^a+\theta_i^b)+c$
FP_4	$m+\sum_{i\in I}\theta_i^b$	$n+\sum_{i\in I}\theta_i^b$	$2m+4\sum_{i\in I}\theta_i^b+c$
Robust reformulations (Disjoint)			
$\mathrm{MILP}_1[\mathcal{U}^{ab}]$	$m(3n+4)$	n	$m(6n+2)+c$
$\mathrm{MILP}_2[\mathcal{U}^{ab}]$	$m(3n+3)$	n	$m(6n+1)+c$
$\mathrm{MILP}_2^{\log}[\mathcal{U}^{ab}]$	$m(2n+3)+\sum_{i\in I}\pi_i$	$n+\sum_{i\in I}\pi_i$	$m(2n+2)+4\sum_{i\in I}\pi_i+c$
Robust reformulations (Joint)			
$\mathrm{MILP}_2[\mathcal{U}^{\overline{ab}}]$ & $\mathrm{MILP}_2[\mathcal{U}_{\propto}^{\overline{ab}}]$	$m(3n+2)$	n	$m(6n+1)+c$
$\mathrm{MILP}_2[\mathcal{U}_{=}^{\overline{ab}}]$	$m(2n+2)$	n	$m(5n+1)+c$
$\mathrm{MILP}[\mathcal{U}^{\overline{a}}]$	$m(2n+1)+2$	n	$m(5n+1)+c+1$

Proposition 12. *Let $\widetilde{a}$ and $\widetilde{b}$ be symmetric, bounded, and independent random variables, i.e., $\widetilde{a}_{ij} = a_{ij} + \eta_{ij} d_{ij}^a$ and $\widetilde{b}_{ij} = b_{ij} + \eta_{i,j+n} d_{ij}^b$, where $\eta_{ij}, \eta_{i,j+n} \in [-1,1]$, for all $i \in I, j \in J$, are independently distributed random variables. For each $i \in I$, in* $\mathrm{RFP}[\mathcal{U}]$

$$(i)\ \text{if } \mathcal{U} = \mathcal{U}^{ab},\ \text{then } Pr\left(\mu_i^\star > \frac{a_{i0} + \sum\limits_{j\in J} \widetilde{a}_{ij} x_j^\star}{b_{i0} + \sum\limits_{j\in J} \widetilde{b}_{ij} x_j^\star}\right) \leqslant B\big(2n, \Gamma_i^a + \Gamma_i^b\big), \qquad \Gamma_i^a, \Gamma_i^b \in \{0,\dots,n\};$$

$$(ii)\ \text{if } \mathcal{U} = \mathcal{U}^{\overline{ab}},\ \text{then } Pr\left(\mu_i^\star > \frac{a_{i0} + \sum\limits_{j\in J} \widetilde{a}_{ij} x_j^\star}{b_{i0} + \sum\limits_{j\in J} \widetilde{b}_{ij} x_j^\star}\right) \leqslant B\big(2n, \Gamma_i\big), \qquad \Gamma_i \in \{0,\dots,2n\};$$

additionally,

$$(iii)\ \text{if } \mathcal{U} = \mathcal{U}^{\overline{a}},\ \text{then } Pr\left(\sum_{i\in I} \mu_i^\star > \sum_{i\in I} \frac{a_{i0} + \sum\limits_{j\in J} \widetilde{a}_{ij} x_j^\star}{b_{i0} + \sum\limits_{j\in J} b_{ij} x_j^\star}\right) \leqslant B\big(m\cdot n, \Gamma\big), \qquad \Gamma \in \{0,\dots,m\cdot n\}.$$

Proof. We prove part (i); parts (ii) and (iii) can be proved in a similar manner. Note that the fractional binary problems subject to uncertain coefficients can be represented as

$$\max_{\substack{x\in X,\\ \mu\geqslant 0}} \sum_{i\in I}\mu_i \tag{3.23a}$$

$$\text{s.t.} \sum_{j\in J}\widetilde{b}_{ij}x_j\mu_i - \sum_{j\in J}\widetilde{a}_{ij}x_j \leqslant a_{i0} - b_{i0}\mu_i \qquad \forall i\in I, \tag{3.23b}$$

when $b_{i0} + \sum_{j\in J}\widetilde{b}_{ij}x_j^\star > 0$. For given $(x^\star, \mu^\star)$, random variables $\widetilde{a}$ and $\widetilde{b}$, and for each $i \in I$, we aim to compute an upper-bound for the probability that i-th constraint in (3.23b) is violated, i.e.,

$$Pr\Big(\sum_{j\in J}\widetilde{b}_{ij}\mu_i^\star x_j^\star - \sum_{j\in J}\widetilde{a}_{ij}x_j^\star > a_{i0} - \mu_i^\star b_{i0}\Big) = Pr\left(\mu_i^\star > \frac{a_{i0} + \sum_{j\in J}\widetilde{a}_{ij}x_j^\star}{b_{i0} + \sum_{j\in J}\widetilde{b}_{ij}x_j^\star}\right).$$

Then, for each $i \in I$,

$$\begin{aligned}
&Pr\left(\sum_{j\in J}\widetilde{b}_{ij}\mu_i^\star x_j^\star - \sum_{j\in J}\widetilde{a}_{ij}x_j^\star > a_{i0} - \mu_i^\star b_{i0}\right)\\
&= Pr\left(\sum_{j\in J} b_{ij}\mu_i^\star x_j^\star + \sum_{j\in J}\eta_{ij}d_{ij}^b\mu_i^\star x_j^\star - \sum_{j\in J}a_{ij}x_j^\star - \sum_{j\in J}\eta_{i,j+n}d_{ij}^a x_j^\star > a_{i0} - \mu_i^\star b_{i0}\right) && (3.24)\\
&= Pr\left(\sum_{j\in J}\eta_{ij}d_{ij}^b\mu_i^\star x_j^\star + \sum_{j\in J}\eta_{i,j+n}d_{ij}^a x_j^\star > a_{i0} - \mu_i^\star b_{i0} - \sum_{j\in J}b_{ij}\mu_i^\star x_j^\star + \sum_{j\in J}a_{ij}x_j^\star\right) && (3.25)\\
&\leqslant Pr\left(\sum_{j\in J}\eta_{ij}d_{ij}^b\mu_i^\star x_j^\star + \sum_{j\in J}\eta_{i,j+n}d_{ij}^a x_j^\star > \sum_{j\in S_{i,b}^\star} d_{ij}^b\mu_i^\star x_j^\star + \sum_{j\in S_{i,a}^\star} d_{ij}^a x_j^\star\right) && (3.26)\\
&= Pr\Bigg(\sum_{j\in J\setminus S_{i,b}^\star}\eta_{ij}d_{ij}^b\mu_i^\star x_j^\star + \sum_{j\in J\setminus S_{i,a}^\star}\eta_{i,j+n}d_{ij}^a x_j^\star >\\
&\qquad \sum_{j\in S_{i,b}^\star} d_{ij}^b\mu_i^\star x_j^\star(1-\eta_{ij}) + \sum_{j\in S_{i,a}^\star} d_{ij}^a x_j^\star(1-\eta_{i,j+n})\Bigg)\\
&\leqslant Pr\left(\sum_{j\in J\setminus S_{i,b}^\star}\eta_{ij}d_{ij}^b\mu_i^\star x_j^\star + \sum_{j\in J\setminus S_{i,a}^\star}\eta_{i,j+n}d_{ij}^a x_j^\star > c_i\sum_{j\in S_{i,b}^\star}(1-\eta_{ij}) + c_i\sum_{j\in S_{i,a}^\star}(1-\eta_{i,j+n})\right) && (3.27)\\
&= Pr\left(\sum_{j\in S_{i,b}^\star}\eta_{ij} + \sum_{j\in S_{i,a}^\star}\eta_{i,j+n} + \sum_{j\in J\setminus S_{i,b}^\star}\eta_{ij}\frac{d_{ij}^b\mu_i^\star x_j^\star}{c_i} + \sum_{j\in J\setminus S_{i,a}^\star}\eta_{i,j+n}\frac{d_{ij}^a x_j^\star}{c_i} > \Gamma_i^a + \Gamma_i^b\right) && (3.28)
\end{aligned}$$

Probability (3.24) is correct for independently and symmetrically distributed random variables $\eta_{ij} \in [-1,1]$ for all $j \in \{1,\ldots,2n\}$. Probability (3.25) is correct since $\eta_{i,j+n} \in [-1,1]$. Let $S_{i,a}^\star$ and $S_{i,b}^\star$ be the sets of indices of parameters that take the robust value in the numerator and the denominator of the i-th ratio, respectively, in a robust optimal solution. Then note that

$$\sum_{j\in J} b_{ij}\mu_i^\star x_j^\star + \sum_{j\in S_{i,b}^\star} d_{ij}^b \mu_i^\star x_j^\star - \sum_{j\in J} a_{ij}x_j^\star + \sum_{j\in S_{i,a}^\star} d_{ij}^a x_j^\star \leqslant a_{i0} - \mu_i^\star b_{i0}$$

is a valid inequality for problem (3.23) under uncertainty set $\mathcal{U}^{ab}$. Thus, probability (3.26) is correct. Additionally, probability (3.27) is correct for $c_i = \min\left\{\{d_{ij}^b\mu_i^\star x_j^\star\}_{j\in S_{i,b}^\star}, \{d_{ij}^a x_j^\star\}_{j\in S_{i,a}^\star}\right\}$. Next, for $j \in \{1, 2, \ldots, 2n\}$ define

$$\gamma_{ij} = \begin{cases} 1, & \text{if } j \in S_{i,b}^\star \text{ or } j - n \in S_{i,a}^\star \\ \frac{d_{ij}^b \mu_i^\star x_j^\star}{c_i}, & \text{if } j \in J \backslash S_{i,b}^\star \\ \frac{d_{ij}^a x_j^\star}{c_i}, & \text{if } j - n \in J\backslash S_{i,a}^\star, \end{cases}$$

(note that $\gamma_{ij} \leqslant 1$ for all $j \in \{1, \ldots, 2n\}$, otherwise $S_{i,a}^\star$ or $S_{i,b}^\star$ are not the robust optimal set of indices). Hence, probability (3.28) is equivalent to

$$Pr\Bigg(\sum_{j\in\{1,\ldots,2n\}} \gamma_{ij}\eta_{ij} > \Gamma_i^a + \Gamma_i^b\Bigg) \leqslant Pr\Bigg(\sum_{j\in\{1,\ldots,2n\}} \gamma_{ij}\eta_{ij} \geqslant \Gamma_i^a + \Gamma_i^b\Bigg) \leqslant B(2n, \Gamma_i^a + \Gamma_i^b).$$

The last inequality follows from Theorem 3 part (a) in [13] for independent and symmetrically distributed random variables $\eta_j \in [-1, 1]$ and $\gamma_{ij} \leqslant 1$, for $j \in J$. □

Proposition 13. *Let $\widetilde{a}$ and $\widetilde{b}$ be symmetric and bounded random variables, i.e., $\widetilde{a}_{ij} = a_{ij} + \eta_{ij}d_{ij}^a$ and $\widetilde{b}_{ij} = b_{ij} + \eta_{ij}d_{ij}^b$, where η_{ij}, for all $i \in I, j \in J$, are independently distributed random variables. For each $i \in I$, in* RFP[$\mathcal{U}$]

$$\text{if } \mathcal{U} = \mathcal{U}_\propto^{\overline{ab}}, \text{ then } Pr\left(\mu_i^\star > \frac{a_{i0} + \sum_{j\in J}\widetilde{a}_{ij}x_j^\star}{b_{i0} + \sum_{j\in J}\widetilde{b}_{ij}x_j^\star}\right) \leqslant B\big(n, \Gamma_i\big), \qquad \Gamma_i \in \{0, \ldots, n\}.$$

Proof. Following the proof of Proposition 12, for each $i \in I$,

$$\begin{aligned}
&Pr\left(\mu_i^\star > \frac{a_{i0} + \sum_{j\in J}\widetilde{a}_{ij}x_j^\star}{b_{i0} + \sum_{j\in J}\widetilde{b}_{ij}x_j^\star}\right)\\
&= Pr\Bigg(\sum_{j\in J}\widetilde{b}_{ij}\mu_i^\star x_j^\star - \sum_{j\in J}\widetilde{a}_{ij}x_j^\star > a_{i0} - \mu_i^\star b_{i0}\Bigg)\\
&= Pr\Bigg(\sum_{j\in J} b_{ij}\mu_i^\star x_j^\star + \sum_{j\in J}\eta_{ij}d_{ij}^b\mu_i^\star x_j^\star - \sum_{j\in J} a_{ij}x_j^\star - \sum_{j\in J}\eta_{ij}d_{ij}^a x_j^\star > a_{i0} - \mu_i^\star b_{i0}\Bigg)
\end{aligned}$$

$$= Pr\Big(\sum_{j\in J}\eta_{ij}(d^b_{ij}\mu^\star_i x^\star_j - d^a_{ij}x^\star_j) > a_{i0} - \mu^\star_i b_{i0} - \sum_{j\in J} b_{ij}\mu^\star_i x^\star_j + \sum_{j\in J} a_{ij}x^\star_j\Big)$$

$$= Pr\Big(\sum_{j\in J}\eta_{ij}|d^b_{ij}\mu^\star_i x^\star_j - d^a_{ij}x^\star_j| > a_{i0} - \mu^\star_i b_{i0} - \sum_{j\in J} b_{ij}\mu^\star_i x^\star_j + \sum_{j\in J} a_{ij}x^\star_j\Big) \tag{3.29}$$

$$\leqslant Pr\Big(\sum_{j\in J}\eta_{ij}|d^b_{ij}\mu^\star_i x^\star_j - d^a_{ij}x^\star_j| > \sum_{j\in S^\star_i}|d^b_{ij}\mu^\star_i x^\star_j - d^a_{ij}x^\star_j|\Big) \tag{3.30}$$

$$= Pr\Big(\sum_{j\in J\setminus S^\star_i}\eta_{ij}|d^b_{ij}\mu^\star_i x^\star_j - d^a_{ij}x^\star_j| > \sum_{j\in S^\star_i}|d^b_{ij}\mu^\star_i x^\star_j - d^a_{ij}x^\star_j|(1-\eta_{ij})\Big)$$

$$\leqslant Pr\Big(\sum_{j\in J\setminus S^\star_i}\eta_{ij}|d^b_{ij}\mu^\star_i x^\star_j - d^a_{ij}x^\star_j| > \sum_{j\in S^\star_i} c_i(1-\eta_{ij})\Big) \tag{3.31}$$

$$= Pr\Big(\sum_{j\in S^\star_i}\eta_{ij} + \sum_{j\in J\setminus S^\star_i}\eta_{ij}\frac{|d^b_{ij}\mu^\star_i x^\star_j - d^a_{ij}x^\star_j|}{c_i} > \Gamma_i\Big) \tag{3.32}$$

Probability (3.29) is correct for $\eta_{ij} \in [-1,1]$. Let $S^\star_i$ be the set of indices of parameters that take the robust value in a robust optimal solution of the i-th ratio. Then note that

$$\sum_{j\in J} b_{ij}\mu^\star_i x^\star_j - \sum_{j\in J} a_{ij}x^\star_j + \sum_{j\in S^\star_i}|d^b_{ij}\mu^\star_i x^\star_j - d^a_{ij}x^\star_j| \leqslant a_{i0} - \mu^\star_i b_{i0}$$

is a valid inequality for for problem (3.23) under uncertainty set $\mathcal{U}^{ab}_\propto$. Thus, probability (3.30) is correct. Additionally, probability (3.31) is correct for $c_i = \min\Big\{|d^b_{ij}\mu^\star_i x^\star_j - d^a_{ij}x^\star_j|\Big\}_{j\in S^\star_i}$. Next, for $j \in \{1,2,\dots,n\}$ define

$$\gamma_{ij} = \begin{cases} 1, & \text{if } j \in S^\star_i \\ \frac{|d^b_{ij}\mu^\star_i x^\star_j - d^a_{ij}x^\star_j|}{c_i}, & \text{if } j \in J\setminus S^\star_i, \end{cases}$$

(note that $\gamma_{ij} \leqslant 1$ for all $j \in J$, otherwise $S^\star_i$ is not the robust optimal set of indices). Hence, probability (3.32) is equivalent to

$$Pr\Big(\sum_{j\in J}\gamma_{ij}\eta_{ij} > \Gamma_i\Big) \leqslant Pr\Big(\sum_{j\in J}\gamma_{ij}\eta_{ij} \geqslant \Gamma_i\Big) \leqslant B\big(n,\Gamma_i\big).$$

The last inequality follows from Theorem 3 part (a) in [13] for independent and symmetrically distributed random variables $\eta_j \in [-1,1]$ and $\gamma_{ij} \leqslant 1$, for $j \in J$. □

Evidently, as the decision-maker is more conservative and selects larger level of uncertainty (Γ), the probability that $\mu^\star_i$ is larger than the value of the i-th ratio for $x^\star$ and random variables $\widetilde{a}$ and $\widetilde{b}$ is smaller. Note that we do not derive a similar upper-bound when $\mathcal{U} = \mathcal{U}^{\overline{ab}}_=$ since we cannot satisfy the key assumption that random variables η are independently distributed.

3.5 Computational results

The computational experiments in this section encompass a case study of a particular assortment problem (see Section 3.5.1), as well as experiments on instances with synthetic data to evaluate the performance of our MILP reformulations (see Section 3.5.2). In both of the following subsections, we describe the relevant test instances, compare the robust and nominal solutions, and discuss relevant aspects of the solutions. Our experiments were performed using CPLEX 12.7.1 [47] on an 8-core CPU (3.7 GHz) with 32 GB of RAM.

3.5.1 Case study: assortment optimization for frozen pizza

Assortment optimization problems arise in many applications such as retailing, revenue management problems, and online advertising. Assortment optimization with uncertainty considerations is a growing area of research; in addition to [81], discussed in Sections 1.1 and 3.3.2, the studies in [11] and [29] have proposed robust optimization approaches for different classes of assortment optimization problems.

Our case study, outlined next, optimizes an assortment problem for a real retailer of frozen pizza studied in [51]; the data is available at `http://cblib.zib.de`. The objective of the assortment problem is to maximize revenue for a company, given a large number of potential product offerings, associated revenues for those offerings, and estimations of customer preferences between those offerings. Additionally, the customers are divided into several different classes, thus the mixed-multinomial logit choice model is a natural fit for the problem.

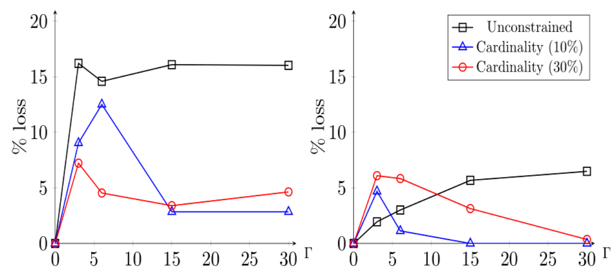
3.5.1.1 Test instances The test instances comprise customer preference data on frozen pizzas from [51]. In particular, there are 130 potential product offerings divided into 5 tiers of revenue ($1.49, $1.75, $1.79, $1.89, and $2.75), and there are 3 classes of customers. Thus, the problem is an instance of (3.3) with $m = 3$ ratios and $n = 130$ variables. The same data was used for each test, with variations in the type of uncertainty set, as well as the level of uncertainty Γ. We fixed $d^a_{ij} = 0.5a_{ij}$, and $d^b_{ij} = 0.5b_{ij}$ (where relevant) for all uncertainty sets.

For the case study we consider four robust problems; specifically, we consider unconstrained ($X = \mathbb{B}^n$) and cardinality-constrained ($X = \{x \in \mathbb{B}^n \mid \sum_{j \in J} x_j \leqslant k\}$) versions of RFP[$\mathcal{U}_{\propto}^{\overline{ab}}$] and RFP[$\mathcal{U}^{\overline{a}}$] which are a natural fit for this application. Uncertainty in customer preferences (ρ_{ij}) and revenues (r_{ij}) can be captured by the matched effects, $\mathcal{U}_{\propto}^{\overline{ab}}$, and the single budget, $\mathcal{U}^{\overline{a}}$, uncertainty sets respectively; see Section 3.2. With respect to the feasible region, we test both the unconstrained case - for an online retailer with the ability to market many options - as well as two sizes of cardinality constraint: $k = 13$ and $k = 39$, corresponding to 10% and 30% of the 130 variables, respectively. The latter problem classes correspond to a small and large retailer, respectively, where there is a physical limitation on the number of products which can be offered to customers.

3.5.1.2 The price of robustness The value in the robust approach is demonstrated by checking the performance of the nominal (optimal) solution in the uncertain environment, and vice versa. These results are shown in Figures 8 and 9. Figure 8 shows the relative decrease in the robust objective function value when the optimal nominal solution is used in the uncertain setting instead of the optimal robust solution (at the given uncertainty level) as "% loss". Figure 9 depicts the opposite case - the loss of using the robust optimal solution when the unknown coefficient take their nominal values. Thus, higher "% loss" in these two figures implies worse results.

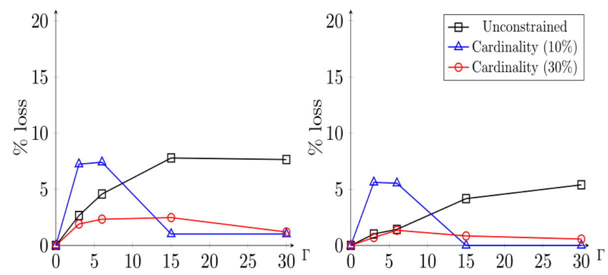
The results for the unconstrained case show that the nominal optimal solution performs worse in the robust setting than the robust solution does in the deterministic environment. Additionally, we observe that, as the level of uncertainty increases for both uncertainty sets, the percentage loss ("% loss") of using both nominal and robust solutions in the opposite setting increases.

The cardinality results exhibit a somewhat different pattern of behavior, although we continue to see that the robust solution performs better in the nominal setting than vice versa. For the cardinality feasible regions, in both uncertainty sets, the nominal and robust solutions are different for small to moderate values of Γ_i, but for the larger values of Γ_i the nominal and robust solutions become similar again. The reason for this behavior is that, as Γ_i grows, all (or almost all) of the variable coefficients in the optimal robust solution are reduced by uncertainty; that is, Γ_i is close to or larger than the size of the cardinality k. Since each uncertain coefficient is reduced by 50% (see above), the most favorable products without uncertainty reduction remain the most favorable products when everything (within the limited cardinality size k) is reduced 50% by uncertainty.



(a) $\mathcal{U}_{\propto}^{\overline{ab}}$ (b) $\mathcal{U}^{\overline{a}}$

Figure 8: Decrease in the robust optimal objective function value by plugging a nominal optimal solution into the robust problem for frozen pizza. Specifically, let $Z^{\star}_{\mathcal{U}}$ denote the optimal objective function value of RFP[$\mathcal{U}$]. Additionally, let $\hat{Z}_{\mathcal{U}} = \min_{(\widetilde{a},\widetilde{b})\in\mathcal{U}} \sum_{i\in I} \frac{a_{i0}+\widetilde{a}_i^T x^{\star}}{b_{i0}+\widetilde{b}_i^T x^{\star}}$ where $x^{\star}$ is a nominal optimal solution. Then % loss for each Γ is $\frac{Z^{\star}_{\mathcal{U}}-\hat{Z}_{\mathcal{U}}}{Z^{\star}_{\mathcal{U}}} \times 100\%$.



(a) $\mathcal{U}_{\propto}^{\overline{ab}}$ (b) $\mathcal{U}^{\overline{a}}$

Figure 9: Decrease in the nominal optimal objective function value by plugging a robust optimal solution into the nominal problem for frozen pizza. Specifically, let $Z^{\star}$ denote the optimal objective function value of FP. Additionally, let $\hat{Z} = \sum_{i\in I} \frac{a_{i0}+a_i^T x^{\star}_{\mathcal{U}}}{b_{i0}+b_i^T x^{\star}_{\mathcal{U}}}$ where $x^{\star}_{\mathcal{U}}$ is an optimal solution of RFP[$\mathcal{U}$]. Then % loss for each Γ is $\frac{Z^{\star}-\hat{Z}}{Z^{\star}} \times 100\%$.

3.5.1.3 Solution Analysis A salient feature of the unconstrained robust solutions in our case study is that, under both uncertainty sets $\mathcal{U}^{\overline{a}}$ and $\mathcal{U}_{\propto}^{\overline{ab}}$, the robust optimal solution contains more variables with $x_j = 1$ as Γ_i increases, see Figure 10. For example, under $\mathcal{U}^{\overline{a}}$, each increase in Γ_i results in roughly 10 more variables included in the optimal solution. With $\Gamma_i = 0$, the optimal solution contains more variables from the highest 2 revenue classes, and as uncertainty increases, more choices from lower revenue classes become part of the solution. This can be explained by observing that, with increasing uncertainty, the Γ_i most favorable products are the ones with their coefficients changed by uncertainty. Hence, the reduction in preference and/or revenue brings these products more in line with the lesser revenue products, which then become part of the optimal solution.

However, somewhat counter-intuitively, given a cardinality size of 13, the optimal solutions (both nominal and robust) consist of variables mostly from the second-highest revenue tier, \$1.89. When the cardinality size is expanded to 39, more variables from both the first and second highest revenue tiers become part of the optimal solutions. An examination of the data shows that the highest revenue tier items are generally (significantly) less-preferred (they have smaller values of preference ρ) than the more reasonably priced second tier items, hence the second tier items show themselves to be superior generators of revenue.

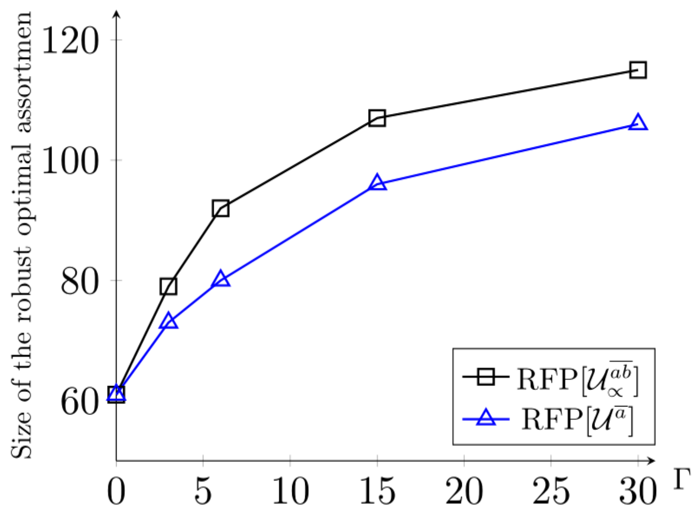


Figure 10: Size of the unconstrained robust optimal assortment versus the the level of uncertainty (Γ).

The outlined observations for (either constrained or unconstrained) multi-class deterministic and robust assortment optimizations can be compared to the previous results in the literature for unconstrained single-class deterministic and robust assortment optimizations. For example, assuming (without loss of generality) that the revenues are ordered such that $r_1 \geqslant r_2 \geqslant \ldots r_n$, Talluri and Van Ryzin [90] show that the unconstrained single-class nominal assortment optimization problems under multi-nominal logit choice model are "revenue-ordered assortments", i.e., there exists a set of optimal solutions of the form $\{1, 2, \ldots, j\}$, for some index j. Rusmevichientong and Topaloglu [81] derive a similar result for the robust case, where uncertainty is limited to customer preferences.

3.5.2 Synthetic instances

We now conduct extensive computational experiments on randomly generated instances to gain insights into the performance of the disjoint and joint MILP reformulations provided in Section 3.4. Additionally, we evaluate the nominal solution in a robust setting, and vice versa, to determine the "price of robustness." In Section 3.5.2.1, we outline the structure and parameters of the computational experiments. The price of robustness is studied in Section 3.5.2.2. We describe the results for the disjoint and joint uncertainty sets in Section 3.5.2.3 and Section 3.5.2.4, respectively.

3.5.2.1 Test instances We chose combinations of $m \in \{1,3,5\}$ and $n \in \{50,100,150\}$. The uncertainty parameters Γ_i^a, Γ_i^b were chosen based on m, n, and the relevant uncertainty set $\mathcal{U}$, and these choices are given in the appropriate section below. For each choice of m, n, Γ and a particular constraint type (detailed below), five instances were sampled and the results averaged. The instances were each given a time limit of 1 hour (3600 seconds).

The LP relaxation quality, denoted by R in the following tables, is computed by $\frac{Z_{LP}^*}{Z^*}$, where Z_{LP}^* is the optimal solution of the LP continuous relaxation, and Z^* is the optimal integer solution (if Z^* cannot be found within the time limit by any solution approach, then the best-known integer solution is used in place of Z^*). Moreover, the optimality gap is denoted by G and is computed by $\frac{UB-LB}{LB}$, where UB and LB are the upper- and the lower-bound on the optimal objective function value, respectively.

Coefficients sampling. The coefficients a_{ij} and b_{ij} were each sampled from a (discrete) $U[0,20]$ distribution, except for b_{i0} which was sampled from a $U[1,20]$. Subsequently, each d_{ij}^a and d_{ij}^b were sampled from $U[0,\lfloor \frac{1}{2}a_{ij} \rfloor]$ or $U[0,\lfloor \frac{1}{2}b_{ij} \rfloor]$, respectively. Note that these parameter choices satisfy Assumptions 1 and 2.

Constraints. Three different constraint types were used: unconstrained (denoted by U in the following tables), cardinality-constrained (C), and knapsack-constrained (K). The cardinality constraint is of the *equality* type so that $\sum_{j \in J} x_j = k$, where $k = \frac{2}{5}n$. The knapsack constraint was of the *inequality* type, structured so that $\sum_{j \in J} k_j x_j \leqslant k$, where k_j was sampled from a $U[1,10]$ distribution, and $k = 2n$.

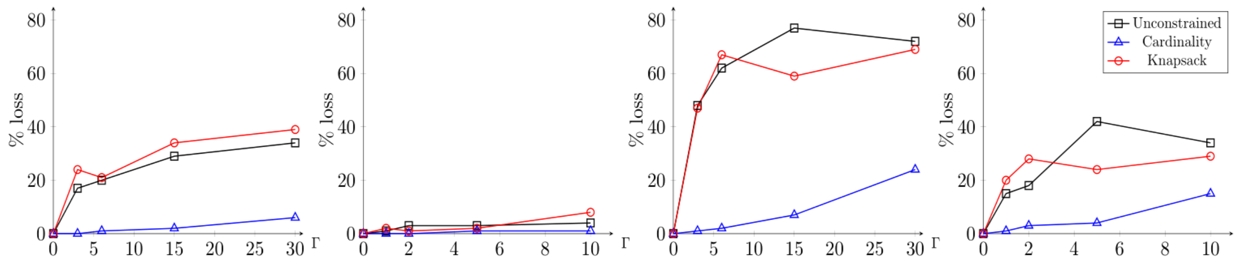
Linearization Bounds. For $\text{MILP}_1[\mathcal{U}^{ab}]$, note that $\omega_i^L = 0$ and $\omega_i^U = 1$ are valid bounds. Similar (not necessarily tight) lower and upper bound computations were performed for the other linearization procedures.

3.5.2.2 The price of robustness Herein, we demonstrate the value of the robust approach; that is, we show that ignoring the possibility of uncertain data can be more costly than being conservative. In Figures 11 and 12, the "small" d_{ij}^a and d_{ij}^b were sampled using the procedure described in Section 3.5.2.1. The "large" d_{ij}^a and d_{ij}^b in these two figures were sampled by instead letting d_{ij}^a and d_{ij}^b be distributed as $U[\lfloor \frac{1}{2} a_{ij} \rfloor, a_{ij}]$ and $U[\lfloor \frac{1}{2} b_{ij} \rfloor, b_{ij}]$, respectively (that is, a higher level of uncertainty). Each sub-figure is comparable to the one directly above/below it.

Figure 11 exhibits the benefit from applying the robust approach. It shows that under the worst-case scenario in the robust setting the objective function value attained by an optimal nominal solution can be rather poor and thus, illustrates how much the decision-maker can gain by taking into account the data uncertainty. More precisely, Figure 11 depicts the average decrease in the robust objective function value for $m \in \{1, 3, 5\}$, by inserting optimal x from the associated nominal problem into the robust problem. We observe that in case of large d, especially for the unconstrained and knapsack-constrained cases, inserting the nominal solution into the robust problem can cause a loss of up to 80%. This observation holds, albeit with scaled-down percentages, for the smaller d values as well.

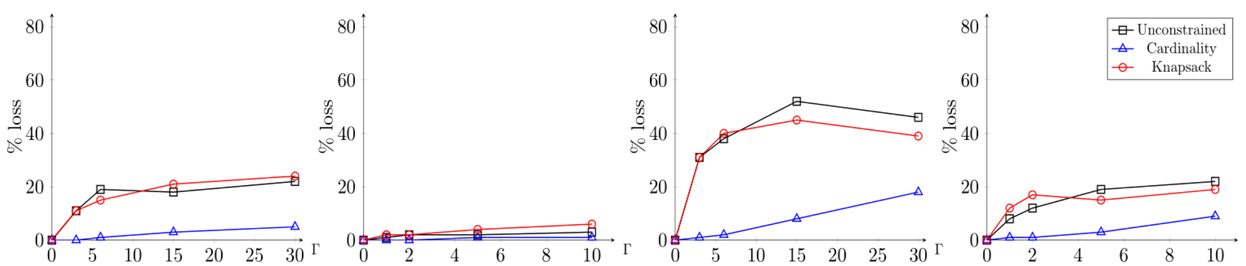
Therefore, we conclude that the decision-maker has more to lose by failing to account for uncertainty than she does by being over-conservative. Simply speaking, if the decision-maker is overly conservative (chooses the Γ_i, for all $i \in I$, too large), then the loss on the objective function is outweighed by the amount she would lose by incorrectly ignoring the uncertainty (i.e., assuming Γ_i=0 for all $i \in I$). These results are similar to those of robust linear problems - see, e.g., [12].

Figure 12 illustrates the opposite situation. That is, it shows how much the decision-maker can gain by having precise information about the problem data parameters. Specifically, Figure 12 depicts the average decrease in the nominal objective function value for $m \in \{1, 3, 5\}$, by inserting robust optimal solution x into the nominal problem. This insertion causes a loss of up to 50% in the objective function value of the nominal problem for large d in case of unconstrained and knapsack-constrained problems.



(a) "small" d, $\mathcal{U}^{ab}$ (b) "small" d, $\mathcal{U}^{\overline{a}}$ (c) "large" d, $\mathcal{U}^{ab}$ (d) "large" d, $\mathcal{U}^{\overline{a}}$

Figure 11: Average decrease in the robust optimal objective function value by plugging a nominal optimal solution into the robust problem for synthetic data and $n = 150$. Specifically, let $Z^\star_{\mathcal{U}}$ denote the optimal objective function value of RFP[$\mathcal{U}$]. Additionally, let $\hat{Z}_{\mathcal{U}} = \min_{(\widetilde{a},\widetilde{b})\in\mathcal{U}} \sum_{i\in I} \frac{a_{i0}+\widetilde{a}_i^T x^\star}{b_{i0}+\widetilde{b}_i^T x^\star}$ where $x^\star$ is a nominal optimal solution. Then % loss for each Γ is the average of $\frac{Z^\star_{\mathcal{U}}-\hat{Z}_{\mathcal{U}}}{Z^\star_{\mathcal{U}}} \cdot 100$ over five test instances and three ratio sizes $m \in \{1, 3, 5\}$.



(a) "small" d, $\mathcal{U}^{ab}$ (b) "small" d, $\mathcal{U}^{\overline{a}}$ (c) "large" d, $\mathcal{U}^{ab}$ (d) "large" d, $\mathcal{U}^{\overline{a}}$

Figure 12: Average decrease in the nominal optimal objective function value by plugging a robust optimal solution into the nominal problem for for synthetic data and $n = 150$. Specifically, let $Z^\star$ denote the optimal objective function value of FP. Additionally, let $\hat{Z} = \sum_{i\in I} \frac{a_{i0}+a_i^T x^\star_{\mathcal{U}}}{b_{i0}+b_i^T x^\star_{\mathcal{U}}}$ where $x^\star_{\mathcal{U}}$ is an optimal solution of RFP[$\mathcal{U}$]. Then % loss for each Γ is the average of $\frac{Z^\star-\hat{Z}}{Z^\star} \cdot 100$ over five test instances and three ratio sizes $m \in \{1, 3, 5\}$.

3.5.2.3 Disjoint reformulations The results for the disjoint uncertainty set $\mathcal{U}^{ab}$ and $n \in \{50, 100, 150\}$ are presented in Tables 6-8, for single-ratio ($m = 1$) and multiple-ratio ($m \in \{3, 5\}$) problems. The uncertainty parameters were chosen so that $\Gamma_i^a = \Gamma_i^b$ for all $i \in I$, as stated in the tables. Observe that, in general, single-ratio problem is easy to solve for any of the constraint types. In particular, the binary reformulation $\text{MILP}_{2'}^{\log}[\mathcal{U}^{ab}]$ (recall Section 3.4.3) can handle the single-ratio setting, in that its average solution times for $m = 1$ in Tables 6-8 are the same as those for the nominal problem $\text{FP}_{4'}$.

As one would expect, increasing either m or n increases the difficulty of the fractional problem under disjoint uncertainty. In the nominal case (see, e.g., [92]), FP_1 generally outperforms the FP_2 across all constraint types for the multiple-ratio problem, and we find that this result carries over into the robust case. Specifically, for $m = 3$ and $m = 5$ in Tables 7 and 8, $\text{MILP}_1[\mathcal{U}^{ab}]$ solves more than half of unconstrained and knapsack instances to optimality, while $\text{MILP}_2[\mathcal{U}^{ab}]$ solves almost none.

However, the *binarized* $\text{MILP}_2^{\log}[\mathcal{U}^{ab}]$ outperforms both $\text{MILP}_1[\mathcal{U}^{ab}]$ and $\text{MILP}_2[\mathcal{U}^{ab}]$. In Table 8, note that when $m = 5$, $\text{MILP}_{2'}^{\log}[\mathcal{U}^{ab}]$ solves all except one of the unconstrained and knapsack instances to optimality, while $\text{MILP}_2^{\log}[\mathcal{U}^{ab}]$ all solves the cardinality-constrained instances to optimality.

For the multiple-ratio problem, the cardinality-constrained problems seem to be the most computationally difficult (when the best solution approach is chosen for each constraint type), although this observation holds for the nominal case as well - see, for example the $m = 5$ case under constraint C in Table 8. On the other hand, the unconstrained problem is sometimes more difficult than the knapsack-constrained problem (as when $\Gamma_i = 1, m = 5$ in Table 6), though not universally so (e.g., $\Gamma_i = 2, m = 5$ in Table 6). Finally, we note that there appears to be no particular pattern or relationship between the level of uncertainty Γ_i^a, Γ_i^b and the computational difficulty for any of the parameter settings.

To summarize these results, we observe that $\text{MILP}_1[\mathcal{U}^{ab}]$ tends to have the best continuous relaxation bound. This observation is consistent with the earlier observations in the literature that the corresponding nominal reformulation FP_1 typically has the best relaxation quality; see, [16, 62]. Nonetheless, this does not always (or even often) lead to superior solution times mainly due to the large size of the reformulation. In particular, for a small number of variables (Table 6), it appears that $\mathrm{MILP}_2^{\log}[\mathcal{U}^{ab}]$ is the best choice for disjoint cardinality-constrained problems, while $\mathrm{MILP}_1[\mathcal{U}^{ab}]$ is usually better for unconstrained or knapsack-constrained models. However, as the number of variables increases (Table 8), the logarithmic reformulation $\mathrm{MILP}_{2'}^{\log}[\mathcal{U}^{ab}]$ is a better choice for unconstrained and knapsack-constrained problems, although it appears that the binarized reformulations have weaker relaxation qualities than the corresponding original MILPs.

3.5.2.4 Joint reformulations Results for joint uncertainty sets $\mathcal{U}^{\overline{ab}}$, $\mathcal{U}^{\overline{ab}}_{=}$ and $\mathcal{U}^{\overline{ab}}_{\propto}$ are given in Tables 9-11 for $n \in \{50, 100, 150\}$. These tables also include the respective results of the most efficient reformulation for the disjoint uncertainty, i.e., $\mathrm{MILP}_{2'}^{\log}[\mathcal{U}^{ab}]$ provided in Tables 6-8, to compare the difficulty of solving $\mathrm{RFP}[\mathcal{U}]$ under disjoint versus joint uncertainty sets.

The uncertainty parameters were chosen based upon those chosen for the disjoint case. With Γ_i^a, Γ_i^b as the relevant disjoint uncertainty parameters, we have: for $\mathcal{U}^{\overline{ab}}$ that $\Gamma_i = 2\,\Gamma_i^a$, for $\mathcal{U}^{\overline{ab}}_{=}$ and $\mathcal{U}^{\overline{ab}}_{\propto}$ that $\Gamma_i = \Gamma_i^a$, and for $\mathcal{U}^{\overline{a}}$ that $\Gamma = m\,\Gamma_i^a$ for problems with similar m, n.

Observe that $\mathrm{MILP}_2[\mathcal{U}]$ performs similarly (with respect to solution times/optimality gap) on both the disjoint and joint uncertainty sets, by comparing the $\mathrm{MILP}_2[\mathcal{U}^{ab}]$ of Table 6 with the relevant columns of Table 9, and conducting similar comparisons for columns of the 100 and 150 variable tables. However, for the disjoint uncertainty case we were able to use a binary reformulation ($\mathrm{MILP}_{2'}^{\log}[\mathcal{U}^{ab}]$) to obtain superior solution times. Thus, the joint problems are generally more computationally difficult than the disjoint due to the absence of such a binary reformulation for them, which can be seen by comparing the first column of Tables 9-11 with the other columns.

Though the multiple-ratio problem utilized the entire hour of solution time allowed for most joint uncertainty sets, the single-ratio problem was solved quickly in most cases. Additionally, for the multiple-ratio problem, $\mathcal{U}^{\overline{a}}$ remains tractable for unconstrained and knapsack-constrained problems. In these two special cases, $\mathrm{MILP}[\mathcal{U}^{\overline{a}}]$ typically solved the joint problem to optimality in a similar time as $\mathrm{MILP}_2^{\log}[\mathcal{U}^{ab}]$ solved the disjoint instance. Finally, we observe that the cardinality constraint is universally difficult (as in the disjoint case) for all multiple-ratio instances with the joint uncertainty sets.

3.6 Concluding remarks

This chapter addresses single- and multiple-ratio RFPs defined as the robust counterparts of the fractional 0-1 programming problems (FPs) under various disjoint and joint uncertainty sets. We demonstrate that single-ratio RFP, contrary to its deterministic counterpart, is NP-hard for a general polyhedral uncertainty set. However, if the uncertainties are in the form of the budgeted uncertainty sets, then we develop polynomial-time solution methods for single-ratio RFP provided that the nominal problem is polynomial-time solvable.

In particular, for the disjoint uncertainty set we propose an approach to solve single-ratio RFP by calling at most $(n+1)^2$ instances of FP. Moreover, in the case of joint uncertainty sets we show that single-ratio RFP can be solved by solving a polynomial number of instances of a linear binary problem. Therefore, if the latter admits a specialized polynomial-time solution algorithm, then single-ratio RFP under dis/joint uncertainty sets is polynomial-time solvable, as well.

In case of multiple-ratio RFPs, we exploit the structure of the budgeted dis/joint uncertainty sets in order to propose various MILPs to solve them. Particularly, based on our extensive computational experiments it is noted that RFPs are more challenging to solve under the joint sets than the disjoint one, as the former cannot take advantage of the binary-expansion technique. Indeed, it appears that as the size of the problem increases, the binarized formulations are often a better choice for the robust problem under the disjoint uncertainty set.

We also explore the value of the robust optimal solution for instances with both the real and synthetic data and find that ignoring the data uncertainty can lead to poor decisions. These results coupled with the insights on the selection of budget(s) of uncertainties can provide guidance to consider the suitable solution method and level of uncertainty in practice.

Table 6: Results for disjoint reformulations. Average time (T) in seconds with the number (#) of instances solved within default optimality gap 0.01, and the average remaining optimality gap (G) along with the average relaxation quality (R) across instances for $n = 50$. In each row, among the solution methods that solve the most number of instances to optimality, the best average time and the best average gap (if #< 5) are in **bold**.

$n = 50$	Cons.	$FP_{4'}$				$MILP_1[\mathcal{U}^{ab}]$				$MILP_2[\mathcal{U}^{ab}]$				$MILP_2^{\log}[\mathcal{U}^{ab}]$				$MILP_{2'}^{\log}[\mathcal{U}^{ab}]$			
$m = 1$	type	T	#	G	R	T	#	G	R	T	#	G	R	T	#	G	R	T	#	G	R
$\Gamma_i^a = \Gamma_i^b = 0$	U	0.1	5	0.00	1.0	**0.0**	5	0.00	1.0	0.3	5	0.00	10.8	0.1	5	0.00	12.0	0.1	5	0.00	1.0
	C	0.2	5	0.00	1.0	1.6	5	0.00	1.9	0.3	5	0.00	16.8	**0.1**	5	0.00	17.1	**0.1**	5	0.00	1.9
	K	0.1	5	0.00	1.0	**0.0**	5	0.00	1.0	0.3	5	0.00	9.3	0.1	5	0.00	9.9	0.0	5	0.00	1.0
$\Gamma_i^a = \Gamma_i^b = 1$	U	0.2	5	0.00	1.2	**0.1**	5	0.00	1.0	0.9	5	0.00	16.0	0.2	5	0.00	17.2	0.2	5	0.00	1.0
	C	0.2	5	0.00	1.2	5.1	5	0.00	1.7	0.7	5	0.00	26.0	0.2	5	0.00	27.6	**0.1**	5	0.00	1.7
	K	0.0	5	0.00	1.4	**0.1**	5	0.00	1.0	0.4	5	0.00	23.0	**0.1**	5	0.00	27.2	**0.1**	5	0.00	1.0
$\Gamma_i^a = \Gamma_i^b = 2$	U	0.1	5	0.00	1.5	**0.1**	5	0.00	1.0	0.8	5	0.00	19.4	0.2	5	0.00	21.7	**0.1**	5	0.00	1.0
	C	0.2	5	0.00	1.2	0.8	5	0.00	1.4	0.7	5	0.00	16.4	0.2	5	0.00	16.8	**0.1**	5	0.00	1.4
	K	0.1	5	0.00	1.4	**0.1**	5	0.00	1.0	0.4	5	0.00	13.8	0.2	5	0.00	14.8	**0.1**	5	0.00	1.0
$\Gamma_i^a = \Gamma_i^b = 5$	U	0.1	5	0.00	1.6	**0.1**	5	0.00	1.0	0.7	5	0.00	21.2	0.2	5	0.00	24.7	**0.1**	5	0.00	1.0
	C	0.2	5	0.00	1.5	2.1	5	0.00	1.4	0.6	5	0.00	19.6	0.2	5	0.00	20.0	**0.1**	5	0.00	1.4
	K	0.1	5	0.00	1.9	**0.1**	5	0.00	1.0	0.4	5	0.00	14.8	**0.1**	5	0.00	15.5	**0.1**	5	0.00	1.0
$\Gamma_i^a = \Gamma_i^b = 10$	U	0.1	5	0.00	1.6	**0.1**	5	0.00	1.0	0.5	5	0.00	23.3	0.2	5	0.00	28.0	**0.1**	5	0.00	1.0
	C	0.2	5	0.00	1.7	4.9	5	0.00	2.2	0.7	5	0.00	32.6	0.2	5	0.00	34.6	**0.1**	5	0.00	2.2
	K	0.0	5	0.00	1.5	**0.0**	5	0.00	1.0	0.5	5	0.00	10.4	0.2	5	0.00	10.6	**0.0**	5	0.00	1.0
Average		0.1	5.0	0.00	1.4	1.0	5.0	0.00	1.3	0.5	5.0	0.00	18.2	0.2	5.0	0.00	19.8	**0.1**	5.0	0.00	1.3
$m = 3$		T	#	G	R	T	#	G	R	T	#	G	R	T	#	G	R	T	#	G	R
$\Gamma_i^a = \Gamma_i^b = 0$	U	0.6	5	0.00	1.5	**0.3**	5	0.00	1.5	2,223.4	3	0.23	18.7	0.5	5	0.00	23.4	0.4	5	0.00	1.5
	C	1.0	5	0.00	1.2	1,798.0	4	0.02	3.1	3,600.0	0	1.07	26.4	1.0	5	0.00	27.8	**0.9**	5	0.00	3.1
	K	0.4	5	0.00	1.5	**0.2**	5	0.00	1.5	1,324.4	4	0.07	16.6	**0.2**	5	0.00	19.9	0.3	5	0.00	1.5
$\Gamma_i^a = \Gamma_i^b = 1$	U	0.4	5	0.00	2.2	**1.1**	5	0.00	1.8	3,600.0	0	0.52	28.5	2.0	5	0.00	34.4	1.2	5	0.00	1.8
	C	0.4	5	0.00	1.2	143.6	5	0.00	2.6	3,600.0	0	0.77	41.4	0.7	5	0.00	48.7	**0.9**	5	0.00	2.6
	K	0.4	5	0.00	1.9	2.0	5	0.00	1.6	2,171.2	2	0.41	19.6	2.0	5	0.00	22.7	**1.3**	5	0.00	1.6
$\Gamma_i^a = \Gamma_i^b = 2$	U	0.3	5	0.00	2.3	**0.6**	5	0.00	1.6	2,972.0	1	0.56	34.7	2.4	5	0.00	44.5	1.4	5	0.00	1.6
	C	0.8	5	0.00	1.3	529.6	5	0.00	2.2	3,600.0	0	0.84	19.4	**1.8**	5	0.00	19.6	2.0	5	0.00	2.2
	K	0.3	5	0.00	2.1	2.7	5	0.00	1.5	1,170.7	4	0.15	19.6	2.9	5	0.00	21.2	**2.3**	5	0.00	1.5
$\Gamma_i^a = \Gamma_i^b = 5$	U	0.4	5	0.00	2.2	7.7	5	0.00	1.5	2,218.2	2	0.43	27.3	2.2	5	0.00	33.5	**1.1**	5	0.00	1.5
	C	0.7	5	0.00	1.6	848.0	4	0.01	2.6	3,600.0	0	0.91	44.1	**3.1**	5	0.00	48.3	12.6	5	0.00	2.6
	K	0.6	5	0.00	2.6	13.7	5	0.00	1.6	2,980.0	2	0.24	26.7	3.9	5	0.00	31.4	**2.3**	5	0.00	1.6
$\Gamma_i^a = \Gamma_i^b = 10$	U	0.4	5	0.00	2.7	**0.5**	5	0.00	1.7	2,920.0	1	0.47	32.8	2.8	5	0.00	40.8	1.3	5	0.00	1.7
	C	0.8	5	0.00	1.8	632.0	5	0.00	2.4	3,600.0	0	1.00	28.2	**6.7**	5	0.00	28.7	29.6	5	0.00	2.4
	K	0.6	5	0.00	2.3	**0.7**	5	0.00	1.5	2,340.3	3	0.18	16.0	2.2	5	0.00	17.2	0.8	5	0.00	1.5
Average		0.5	5.0	0.00	1.9	265.4	4.9	0.00	1.9	2,794.7	1.5	0.52	26.7	**2.3**	5.0	0.00	30.8	3.9	5.0	0.00	1.9
$m = 5$		T	#	G	R	T	#	G	R	T	#	G	R	T	#	G	R	T	#	G	R
$\Gamma_i^a = \Gamma_i^b = 0$	U	3.3	5	0.00	1.9	**1.0**	5	0.00	1.9	2,883.6	1	0.67	24.0	7.7	5	0.00	30.5	8.0	5	0.00	1.9
	C	57.8	5	0.00	1.2	3,600.0	0	0.16	3.9	3,600.0	0	1.56	46.5	**12.5**	5	0.00	51.8	20.5	5	0.00	3.9
	K	4.8	5	0.00	1.8	**1.2**	5	0.00	1.8	2,884.2	1	0.55	18.7	10.7	5	0.00	20.9	10.9	5	0.00	1.8
$\Gamma_i^a = \Gamma_i^b = 1$	U	8.4	5	0.00	2.4	**76.3**	5	0.00	1.9	2,360.0	2	0.77	34.7	816.3	4	0.00	47.2	307.0	5	0.00	1.9
	C	26.1	5	0.00	1.4	3,080.0	1	0.09	2.6	3,600.0	0	1.11	26.5	**14.4**	5	0.00	27.2	24.0	5	0.00	2.6
	K	9.2	5	0.00	2.5	342.2	5	0.00	1.9	2,948.0	1	0.55	22.3	216.8	5	0.00	25.4	**132.2**	5	0.00	1.9
$\Gamma_i^a = \Gamma_i^b = 2$	U	7.8	5	0.00	2.4	**74.4**	5	0.00	1.8	2,922.0	1	0.86	29.2	645.0	5	0.00	37.6	111.6	5	0.00	1.8
	C	18.7	5	0.00	1.4	3,600.0	0	0.08	3.1	3,600.0	0	1.20	33.6	**22.4**	5	0.00	36.8	67.3	5	0.00	3.1
	K	16.7	5	0.00	2.7	906.8	4	0.01	1.9	3,600.0	0	0.98	26.0	1,629.0	3	0.01	27.9	**297.0**	5	0.00	1.9
$\Gamma_i^a = \Gamma_i^b = 5$	U	4.7	5	0.00	2.9	**9.3**	5	0.00	1.8	3,600.0	0	0.91	30.4	273.6	5	0.00	37.8	52.2	5	0.00	1.8
	C	25.5	5	0.00	1.6	3,600.0	0	0.08	2.6	3,600.0	0	1.22	34.3	**74.8**	5	0.00	37.3	513.0	5	0.00	2.6
	K	2.9	5	0.00	2.1	**0.7**	5	0.00	1.5	1,521.6	3	0.21	23.6	42.9	5	0.00	28.0	25.4	5	0.00	1.5
$\Gamma_i^a = \Gamma_i^b = 10$	U	4.2	5	0.00	2.6	**22.4**	5	0.00	1.7	2,244.0	2	0.46	28.5	264.6	5	0.00	36.6	59.6	5	0.00	1.7
	C	17.7	5	0.00	1.8	3,600.0	0	0.11	3.5	3,600.0	0	1.36	40.3	**94.4**	5	0.00	43.5	751.6	5	0.00	3.5
	K	3.3	5	0.00	2.8	**0.6**	5	0.00	1.8	3,000.0	2	0.30	22.6	176.2	5	0.00	24.1	51.0	5	0.00	1.8
Average		14.1	5.0	0.00	2.1	1,261.0	3.3	0.03	2.2	3,064.2	0.9	0.85	29.4	286.8	4.8	0.00	34.2	**162.1**	5.0	0.00	2.2

Table 7: Results for disjoint reformulations. Average time (T) in seconds with the number (#) of instances solved within default optimality gap 0.01, and the average remaining optimality gap (G) along with the average relaxation quality (R) across instances for $n = 100$. In each row, among the solution methods that solve the most number of instances to optimality, the best average time and the best average gap (if #< 5) are in **bold**.

$n = 100$	Cons.	$FP_{4'}$				$MILP_1[\mathcal{U}^{ab}]$				$MILP_2[\mathcal{U}^{ab}]$				$MILP_2^{\log}[\mathcal{U}^{ab}]$				$MILP_{2'}^{\log}[\mathcal{U}^{ab}]$			
$m = 1$	type	T	#	G	R	T	#	G	R	T	#	G	R	T	#	G	R	T	#	G	R
$\Gamma_i^a = \Gamma_i^b = 0$	U	0.3	5	0.00	1.0	**0.1**	5	0.00	1.0	0.5	5	0.00	20.9	0.2	5	0.00	24.0	**0.1**	5	0.00	1.0
	C	0.2	5	0.00	1.0	3,048.0	1	0.08	2.8	1.0	5	0.00	64.9	0.2	5	0.00	74.4	**0.1**	5	0.00	2.8
	K	0.0	5	0.00	1.0	**0.0**	5	0.00	1.0	0.6	5	0.00	14.4	0.2	5	0.00	15.0	**0.0**	5	0.00	1.0
$\Gamma_i^a = \Gamma_i^b = 2$	U	0.1	5	0.00	1.2	**0.1**	5	0.00	1.0	1.4	5	0.00	25.4	0.3	5	0.00	31.8	0.2	5	0.00	1.0
	C	0.2	5	0.00	1.1	2,942.0	1	0.18	2.7	13.1	5	0.00	64.8	0.3	5	0.00	69.5	**0.2**	5	0.00	2.7
	K	0.1	5	0.00	1.2	**0.1**	5	0.00	1.0	1.2	5	0.00	16.1	**0.1**	5	0.00	16.6	**0.1**	5	0.00	1.0
$\Gamma_i^a = \Gamma_i^b = 4$	U	0.1	5	0.00	1.5	**0.1**	5	0.00	1.0	1.2	5	0.00	34.0	0.3	5	0.00	42.5	**0.1**	5	0.00	1.0
	C	0.2	5	0.00	1.2	2,891.2	1	0.13	2.3	70.9	5	0.00	80.1	0.4	5	0.00	87.5	**0.1**	5	0.00	2.3
	K	0.1	5	0.00	1.5	**0.1**	5	0.00	1.1	1.2	5	0.00	23.5	0.2	5	0.00	25.3	0.2	5	0.00	1.1
$\Gamma_i^a = \Gamma_i^b = 10$	U	0.1	5	0.00	1.6	**0.1**	5	0.00	1.0	1.0	5	0.00	39.8	0.3	5	0.00	46.4	0.2	5	0.00	1.0
	C	0.2	5	0.00	1.3	3,600.0	0	0.14	2.4	125.6	5	0.00	96.5	**0.3**	5	0.00	111.1	**0.3**	5	0.00	2.4
	K	0.0	5	0.00	1.4	**0.1**	5	0.00	1.0	1.2	5	0.00	19.2	0.2	5	0.00	19.9	**0.1**	5	0.00	1.0
$\Gamma_i^a = \Gamma_i^b = 20$	U	0.1	5	0.00	1.4	**0.1**	5	0.00	1.0	1.2	5	0.00	31.3	0.3	5	0.00	40.4	**0.1**	5	0.00	1.0
	C	0.2	5	0.00	1.5	3,600.0	0	0.13	2.9	14.4	5	0.00	74.8	0.4	5	0.00	80.8	**0.3**	5	0.00	2.9
	K	0.1	5	0.00	1.6	**0.0**	5	0.00	1.0	1.1	5	0.00	35.2	0.2	5	0.00	39.0	0.1	5	0.00	1.0
Average		0.1	5.0	0.00	1.3	1,072.1	3.5	0.04	1.5	15.7	5.0	0.00	42.7	0.3	5.0	0.00	48.3	**0.1**	5.0	0.00	1.5
$m = 3$		T	#	G	R	T	#	G	R	T	#	G	R	T	#	G	R	T	#	G	R
$\Gamma_i^a = \Gamma_i^b = 0$	U	1.0	5	0.00	1.9	6.4	5	0.00	1.9	3,600.0	0	2.48	31.5	1.5	5	0.00	35.9	**0.9**	5	0.00	1.9
	C	2.2	5	0.00	1.2	3,600.0	0	0.49	3.3	3,600.0	0	4.22	35.0	1.8	5	0.00	35.7	**1.5**	5	0.00	3.3
	K	0.4	5	0.00	1.7	2.0	5	0.00	1.7	3,160.0	1	1.30	31.5	1.0	5	0.00	38.4	**0.6**	5	0.00	1.7
$\Gamma_i^a = \Gamma_i^b = 2$	U	0.4	5	0.00	2.1	4.6	5	0.00	1.7	3,600.0	0	2.28	47.1	4.1	5	0.00	65.4	**1.3**	5	0.00	1.7
	C	3.4	5	0.00	1.2	3,600.0	0	0.46	3.5	3,600.0	0	4.84	66.3	**4.2**	5	0.00	69.4	5.0	5	0.00	3.5
	K	0.4	5	0.00	2.0	17.5	5	0.00	1.6	3,600.0	0	1.99	32.6	3.3	5	0.00	39.2	**2.3**	5	0.00	1.6
$\Gamma_i^a = \Gamma_i^b = 4$	U	0.7	5	0.00	2.5	981.4	5	0.00	1.8	3,600.0	0	3.62	48.5	9.0	5	0.00	61.3	**4.8**	5	0.00	1.8
	C	1.9	5	0.00	1.3	3,600.0	0	0.36	2.7	3,600.0	0	4.40	60.1	5.3	5	0.00	67.1	**5.1**	5	0.00	2.7
	K	1.0	5	0.00	2.7	1,656.0	5	0.00	2.0	3,600.0	0	3.46	31.6	17.3	5	0.00	32.8	**7.8**	5	0.00	2.0
$\Gamma_i^a = \Gamma_i^b = 10$	U	0.5	5	0.00	2.4	47.4	5	0.00	1.6	3,600.0	0	3.35	54.2	6.7	5	0.00	73.9	**2.4**	5	0.00	1.6
	C	5.2	5	0.00	1.4	3,600.0	0	0.33	2.9	3,600.0	0	4.92	86.6	**11.6**	5	0.00	100.2	407.9	5	0.00	2.9
	K	0.3	5	0.00	2.1	**0.5**	5	0.00	1.4	3,600.0	0	1.42	41.1	2.4	5	0.00	50.4	0.8	5	0.00	1.4
$\Gamma_i^a = \Gamma_i^b = 20$	U	0.9	5	0.00	2.8	11.2	5	0.00	1.7	3,600.0	0	4.42	55.7	7.3	5	0.00	68.3	**3.3**	5	0.00	1.7
	C	1.4	5	0.00	1.6	3,600.0	0	0.32	2.8	3,600.0	0	5.44	63.6	**30.4**	5	0.00	65.8	1,273.2	5	0.00	2.8
	K	0.5	5	0.00	2.3	724.9	4	0.00	1.8	3,600.0	0	2.84	28.9	5.9	5	0.00	31.0	**2.2**	5	0.00	1.8
Average		1.4	5.0	0.00	1.9	1,430.1	3.3	0.13	2.2	3,570.7	0.1	3.40	47.6	**7.4**	5.0	0.00	55.7	114.6	5.0	0.00	2.2
$m = 5$		T	#	G	R	T	#	G	R	T	#	G	R	T	#	G	R	T	#	G	R
$\Gamma_i^a = \Gamma_i^b = 0$	U	10.4	5	0.00	2.1	282.6	5	0.00	2.1	3,600.0	0	2.98	41.5	58.4	5	0.00	56.4	**24.0**	5	0.00	2.1
	C	1,676.0	5	0.00	1.3	3,600.0	0	0.67	4.3	3,600.0	0	5.92	70.5	**93.4**	5	0.00	79.9	393.8	5	0.00	4.3
	K	8.3	5	0.00	2.0	169.6	5	0.00	2.0	3,600.0	0	1.21	23.7	**21.6**	5	0.00	26.8	21.8	5	0.00	2.0
$\Gamma_i^a = \Gamma_i^b = 2$	U	7.1	5	0.00	2.6	1,567.5	3	0.07	2.1	3,600.0	0	4.84	50.5	1,312.4	4	0.01	64.8	**206.4**	5	0.00	2.1
	C	1,560.6	5	0.00	1.4	3,600.0	0	0.62	3.4	3,600.0	0	5.94	53.7	**313.6**	5	0.00	55.6	1,299.6	5	0.00	3.4
	K	9.7	5	0.00	2.7	2,198.4	2	0.07	2.2	3,600.0	0	2.28	25.0	400.0	5	0.00	26.3	**167.2**	5	0.00	2.2
$\Gamma_i^a = \Gamma_i^b = 4$	U	4.8	5	0.00	3.3	1,692.8	3	0.07	2.2	3,600.0	0	5.40	66.9	1,139.8	5	0.00	89.8	**480.0**	5	0.00	2.2
	C	1,146.2	5	0.00	1.4	3,600.0	0	0.57	3.3	3,600.0	0	6.16	79.1	**98.0**	5	0.00	86.0	431.4	5	0.00	3.3
	K	5.9	5	0.00	3.1	2,161.2	2	0.12	2.1	3,600.0	0	1.98	33.7	1,023.8	4	0.03	37.2	**378.2**	5	0.00	2.1
$\Gamma_i^a = \Gamma_i^b = 10$	U	13.4	5	0.00	3.1	2,166.1	2	0.20	2.1	3,600.0	0	5.78	70.3	1,870.0	3	0.06	99.3	**1,524.8**	3	**0.02**	2.1
	C	1,764.0	4	0.00	1.5	3,600.0	0	0.50	3.1	3,600.0	0	6.86	70.5	**804.0**	5	0.00	77.1	3,600.0	4	0.01	3.1
	K	11.3	5	0.00	2.7	**725.6**	4	0.07	1.7	3,600.0	0	1.61	30.3	897.4	4	**0.01**	32.6	964.0	4	**0.01**	1.7
$\Gamma_i^a = \Gamma_i^b = 20$	U	14.3	5	0.00	3.2	904.0	4	0.01	1.9	3,600.0	0	5.12	56.2	1,304.0	4	0.02	66.6	**182.0**	5	0.00	1.9
	C	839.0	5	0.00	1.7	3,600.0	0	0.50	3.5	3,600.0	0	7.44	64.7	**1,614.0**	5	0.00	66.2	3,600.0	1	0.01	3.5
	K	16.2	5	0.00	3.3	751.6	4	0.04	2.0	3,600.0	0	2.06	25.1	762.0	5	0.00	25.8	**238.2**	5	0.00	2.0
Average		472.5	4.9	0.00	2.4	2,041.3	2.3	0.23	2.5	3,600.0	0.0	4.37	50.8	**780.8**	4.6	**0.01**	59.4	900.8	4.5	0.00	2.5

Table 8: Results for disjoint reformulations. Average time (T) in seconds with the number (#) of instances solved within default optimality gap 0.01, and the average remaining optimality gap (G) along with the average relaxation quality (R) across instances for $n = 150$. In each row, among the solution methods that solve the most number of instances to optimality, the best average time and the best average gap (if #< 5) are in **bold**.

$n = 150$	Cons.	$FP_{4'}$				$MILP_1[\mathcal{U}^{ab}]$				$MILP_2[\mathcal{U}^{ab}]$				$MILP_2^{\log}[\mathcal{U}^{ab}]$				$MILP_{2'}^{\log}[\mathcal{U}^{ab}]$			
$m = 1$	type	T	#	G	R	T	#	G	R	T	#	G	R	T	#	G	R	T	#	G	R
$\Gamma_i^a = \Gamma_i^b = 0$	U	0.1	5	0.00	1.0	**0.1**	5	0.00	1.0	0.7	5	0.00	29.6	0.2	5	0.00	37.5	**0.1**	5	0.00	1.0
	C	0.2	5	0.00	1.0	3,600.0	0	0.31	2.4	32.5	5	0.00	45.6	0.3	5	0.00	46.7	**0.1**	5	0.00	2.4
	K	0.1	5	0.00	1.0	**0.1**	5	0.00	1.0	1.0	5	0.00	20.7	0.2	5	0.00	22.5	**0.1**	5	0.00	1.0
$\Gamma_i^a = \Gamma_i^b = 3$	U	0.1	5	0.00	1.3	**0.2**	5	0.00	1.0	2.3	5	0.00	34.7	**0.2**	5	0.00	42.3	**0.2**	5	0.00	1.0
	C	0.2	5	0.00	1.1	3,600.0	0	0.30	2.0	80.0	5	0.00	31.3	0.3	5	0.00	31.8	**0.2**	5	0.00	2.0
	K	0.1	5	0.00	1.3	**0.1**	5	0.00	1.0	1.3	5	0.00	26.0	0.3	5	0.00	27.4	**0.1**	5	0.00	1.0
$\Gamma_i^a = \Gamma_i^b = 6$	U	0.2	5	0.00	1.4	**0.1**	5	0.00	1.1	6.0	5	0.00	38.8	0.3	5	0.00	47.2	0.2	5	0.00	1.1
	C	0.2	5	0.00	1.1	3,600.0	0	0.31	2.4	1,112.9	4	0.48	88.8	0.3	5	0.00	109.5	**0.2**	5	0.00	2.4
	K	0.1	5	0.00	1.4	**0.2**	5	0.00	1.1	1.5	5	0.00	18.6	0.3	5	0.00	18.9	**0.2**	5	0.00	1.1
$\Gamma_i^a = \Gamma_i^b = 15$	U	0.2	5	0.00	1.6	**0.2**	5	0.00	1.0	2.5	5	0.00	47.3	0.3	5	0.00	57.3	**0.2**	5	0.00	1.0
	C	0.2	5	0.00	1.2	3,600.0	0	0.25	1.8	473.3	5	0.00	46.6	**0.3**	5	0.00	47.3	**0.3**	5	0.00	1.8
	K	0.1	5	0.00	1.9	**0.1**	5	0.00	1.0	1.9	5	0.00	43.8	0.4	5	0.00	46.9	0.2	5	0.00	1.0
$\Gamma_i^a = \Gamma_i^b = 30$	U	0.2	5	0.00	1.9	**0.1**	5	0.00	1.0	1.9	5	0.00	39.8	0.5	5	0.00	43.0	0.2	5	0.00	1.0
	C	0.2	5	0.00	1.4	3,600.0	0	0.28	2.1	931.9	4	0.12	72.2	**0.4**	5	0.00	74.5	**0.4**	5	0.00	2.1
	K	0.1	5	0.00	1.6	**0.1**	5	0.00	1.0	1.6	5	0.00	26.3	0.3	5	0.00	27.4	0.2	5	0.00	1.0
Average		0.2	5.0	0.00	1.4	1,200.1	3.3	0.10	1.4	176.7	4.9	0.04	40.7	0.3	5.0	0.00	45.3	**0.2**	5.0	0.00	1.4
$m = 3$		T	#	G	R	T	#	G	R	T	#	G	R	T	#	G	R	T	#	G	R
$\Gamma_i^a = \Gamma_i^b = 0$	U	0.7	5	0.00	1.8	721.0	4	0.01	1.8	3,600.0	0	3.62	46.0	0.8	5	0.00	62.3	**0.7**	5	0.00	1.8
	C	4.9	5	0.00	1.2	3,600.0	0	0.99	5.1	3,600.0	0	9.66	118.6	4.2	5	0.00	141.2	**3.2**	5	0.00	5.1
	K	0.5	5	0.00	1.7	3.1	5	0.00	1.7	3,600.0	0	2.86	38.9	0.9	5	0.00	46.4	**0.8**	5	0.00	1.7
$\Gamma_i^a = \Gamma_i^b = 3$	U	0.5	5	0.00	2.3	450.2	5	0.00	1.8	3,600.0	0	5.54	69.9	7.0	5	0.00	92.7	**4.1**	5	0.00	1.8
	C	3.1	5	0.00	1.2	3,600.0	0	0.81	3.9	3,600.0	0	9.36	109.8	**7.4**	5	0.00	122.7	30.8	5	0.00	3.9
	K	0.5	5	0.00	2.4	929.3	4	0.02	1.9	3,600.0	0	4.94	48.0	7.4	5	0.00	52.2	**5.2**	5	0.00	1.9
$\Gamma_i^a = \Gamma_i^b = 6$	U	0.7	5	0.00	2.8	1,660.5	3	0.04	2.0	3,600.0	0	6.62	70.6	11.5	5	0.00	89.2	**5.4**	5	0.00	2.0
	C	8.8	5	0.00	1.3	3,600.0	0	0.68	3.1	3,600.0	0	8.40	56.2	**6.1**	5	0.00	57.4	29.6	5	0.00	3.1
	K	0.9	5	0.00	2.4	1,472.3	3	0.04	1.8	3,600.0	0	4.28	38.5	12.5	5	0.00	43.0	**6.9**	5	0.00	1.8
$\Gamma_i^a = \Gamma_i^b = 15$	U	0.5	5	0.00	2.9	2,164.4	2	0.04	1.8	3,600.0	0	7.84	91.2	20.0	5	0.00	122.4	**13.7**	5	0.00	1.8
	C	6.3	5	0.00	1.4	3,600.0	0	0.59	2.9	3,600.0	0	9.58	62.3	**13.0**	5	0.00	64.0	49.9	5	0.00	2.9
	K	0.8	5	0.00	2.8	2,160.5	2	0.10	1.9	3,600.0	0	5.98	45.1	30.0	5	0.00	47.7	**16.5**	5	0.00	1.9
$\Gamma_i^a = \Gamma_i^b = 30$	U	0.9	5	0.00	2.7	721.5	4	0.05	1.7	3,600.0	0	6.72	58.7	119.8	5	0.00	69.0	**30.0**	5	0.00	1.7
	C	3.7	5	0.00	1.5	3,600.0	0	0.58	3.1	3,600.0	0	11.48	65.1	**22.8**	5	0.00	66.2	1,448.6	5	0.00	3.1
	K	0.8	5	0.00	2.7	730.0	4	0.06	1.6	3,600.0	0	4.68	47.2	**55.4**	5	0.00	53.8	204.1	5	0.00	1.6
Average		2.2	5.0	0.00	2.1	1,934.2	2.4	0.27	2.4	3,600.0	0.0	6.77	64.4	**21.3**	5.0	0.00	75.3	123.3	5.0	0.00	2.4
$m = 5$		T	#	G	R	T	#	G	R	T	#	G	R	T	#	G	R	T	#	G	R
$\Gamma_i^a = \Gamma_i^b = 0$	U	16.9	5	0.00	2.4	2,004.4	3	0.06	2.4	3,600.0	0	7.18	57.3	46.0	5	0.00	73.3	**32.6**	5	0.00	2.4
	C	2,210.0	4	0.00	1.3	3,600.0	0	1.18	4.5	3,600.0	0	11.58	63.6	**234.0**	5	0.00	65.5	666.6	5	0.00	4.5
	K	20.2	5	0.00	2.3	2,164.8	2	0.09	2.3	3,600.0	0	5.06	40.6	39.1	5	0.00	46.3	**34.7**	5	0.00	2.3
$\Gamma_i^a = \Gamma_i^b = 3$	U	30.6	5	0.00	3.2	2,888.8	1	0.23	2.5	3,600.0	0	10.22	91.4	3,012.0	1	0.11	113.7	**960.0**	5	0.00	2.5
	C	2,250.0	5	0.00	1.3	3,600.0	0	1.03	4.0	3,600.0	0	12.40	105.5	**370.0**	5	0.00	114.4	2,302.0	5	0.00	4.0
	K	15.3	5	0.00	3.0	2,884.2	1	0.25	2.4	3,600.0	0	7.28	58.2	1,726.0	4	0.02	67.5	**734.0**	5	0.00	2.4
$\Gamma_i^a = \Gamma_i^b = 6$	U	24.2	5	0.00	3.0	2,160.7	2	0.25	2.2	3,600.0	0	8.84	84.6	1,574.6	3	0.05	110.6	**1,578.6**	4	**0.00**	2.2
	C	1,067.8	5	0.00	1.4	3,600.0	0	1.04	4.3	3,600.0	0	13.80	159.8	**1,047.2**	5	0.00	185.9	1,608.8	5	0.00	4.3
	K	7.8	5	0.00	2.6	1,442.0	3	0.16	2.0	3,600.0	0	5.48	55.5	1,505.0	3	0.03	64.1	**417.0**	5	0.00	2.0
$\Gamma_i^a = \Gamma_i^b = 15$	U	17.9	5	0.00	3.0	2,161.8	2	0.16	1.9	3,600.0	0	9.22	82.3	1,478.0	4	0.03	105.7	**1,018.0**	4	**0.02**	1.9
	C	1,356.6	5	0.00	1.5	3,600.0	0	0.81	3.5	3,600.0	0	13.80	102.7	**1,568.0**	5	0.00	111.1	3,600.0	4	0.01	3.5
	K	55.0	5	0.00	3.4	2,166.2	2	0.31	2.0	3,600.0	0	9.84	76.5	2,278.0	2	0.14	85.9	**1,806.0**	3	0.03	2.0
$\Gamma_i^a = \Gamma_i^b = 30$	U	9.8	5	0.00	3.1	741.4	4	0.05	1.9	3,600.0	0	8.46	81.9	1,790.0	3	0.26	100.9	**306.0**	5	0.00	1.9
	C	3,160.0	5	0.00	1.6	3,600.0	0	0.71	3.5	3,600.0	0	14.60	107.0	**3,440.0**	5	0.00	111.5	3,600.0	1	0.01	3.5
	K	20.3	5	0.00	3.3	**782.4**	4	0.13	1.9	3,600.0	0	7.74	75.8	1,308.0	4	0.05	94.9	1,056.0	4	**0.02**	1.9
Average		684.1	4.9	0.00	2.4	2,493.1	1.6	0.43	2.8	3,600.0	0.0	9.70	82.8	1,427.7	3.9	0.05	96.8	**1,314.7**	4.3	**0.01**	2.8

Table 9: Comparison of results for the best disjoint reformulation $\text{MILP}^{\log}_{2'}[\mathcal{U}^{ab}]$ versus joint reformulations ($\text{MILP}_2[\mathcal{U}^{\overline{ab}}]$, $\text{MILP}_2[\mathcal{U}^{\overline{ab}}_{=}]$, $\text{MILP}_2[\mathcal{U}^{\overline{ab}}_{\propto}]$ and $\text{MILP}[\mathcal{U}^{\overline{a}}]$). Average time (T) in seconds with the number (#) of instances solved within default optimality gap 0.01, and the average remaining optimality gap (G) along with the average relaxation quality (R) across instances for $n = 50$. We have: for $\mathcal{U}^{\overline{ab}}$ that $\Gamma_i = 2\,\Gamma^a_i$, for $\mathcal{U}^{\overline{ab}}_{=}$ and $\mathcal{U}^{\overline{ab}}_{\propto}$ that $\Gamma_i = \Gamma^a_i$, and for $\mathcal{U}^{\overline{a}}$ that $\Gamma = m\,\Gamma^a_i$.

$n = 50$	Cons.	$\text{MILP}^{\log}_{2'}[\mathcal{U}^{ab}]$				$\text{MILP}_2[\mathcal{U}^{\overline{ab}}]$				$\text{MILP}_2[\mathcal{U}^{\overline{ab}}_{=}]$				$\text{MILP}_2[\mathcal{U}^{\overline{ab}}_{\propto}]$				$\text{MILP}[\mathcal{U}^{\overline{a}}]$			
$m = 1$	type	T	#	G	R	T	#	G	R	T	#	G	R	T	#	G	R	T	#	G	R
	U	0.1	5	0.00	1.0	0.2	5	0.00	10.8	0.1	5	0.00	10.8	0.1	5	0.00	10.8	0.0	5	0.00	1.0
$\Gamma^a_i = 0$	C	0.1	5	0.00	1.9	0.3	5	0.00	16.8	0.3	5	0.00	16.8	0.3	5	0.00	16.8	1.6	5	0.00	1.9
	K	0.0	5	0.00	1.0	0.3	5	0.00	9.3	0.2	5	0.00	9.3	0.3	5	0.00	9.3	0.0	5	0.00	1.0
	U	0.2	5	0.00	1.0	0.7	5	0.00	16.2	0.6	5	0.00	15.7	0.7	5	0.00	14.9	0.1	5	0.00	1.0
$\Gamma^a_i = 1$	C	0.1	5	0.00	1.7	0.6	5	0.00	25.9	0.7	5	0.00	25.7	0.8	5	0.00	24.8	2.3	5	0.00	2.0
	K	0.1	5	0.00	1.0	0.3	5	0.00	22.9	0.4	5	0.00	22.3	0.3	5	0.00	21.1	0.1	5	0.00	1.0
	U	0.1	5	0.00	1.0	0.7	5	0.00	19.2	0.6	5	0.00	19.0	0.7	5	0.00	16.2	0.1	5	0.00	1.0
$\Gamma^a_i = 2$	C	0.1	5	0.00	1.4	0.7	5	0.00	16.2	0.6	5	0.00	15.7	0.7	5	0.00	15.1	0.5	5	0.00	1.6
	K	0.1	5	0.00	1.0	0.4	5	0.00	13.7	0.4	5	0.00	13.6	0.4	5	0.00	12.6	0.0	5	0.00	1.0
	U	0.1	5	0.00	1.0	0.7	5	0.00	20.4	0.7	5	0.00	21.2	0.7	5	0.00	18.5	0.1	5	0.00	1.0
$\Gamma^a_i = 5$	C	0.1	5	0.00	1.4	0.8	5	0.00	19.0	0.7	5	0.00	18.6	0.5	5	0.00	16.7	1.7	5	0.00	1.5
	K	0.1	5	0.00	1.0	0.3	5	0.00	14.2	0.4	5	0.00	14.6	0.4	5	0.00	11.7	0.1	5	0.00	1.0
	U	0.1	5	0.00	1.0	0.5	5	0.00	22.0	0.5	5	0.00	23.3	0.6	5	0.00	20.8	0.1	5	0.00	1.0
$\Gamma^a_i = 10$	C	0.1	5	0.00	2.2	0.7	5	0.00	31.6	0.7	5	0.00	30.3	0.9	5	0.00	26.8	2.3	5	0.00	2.2
	K	0.0	5	0.00	1.0	0.4	5	0.00	10.0	0.4	5	0.00	10.4	0.4	5	0.00	8.7	0.0	5	0.00	1.0
Average		0.1	5.0	0.00	1.3	0.5	5.0	0.00	17.9	0.5	5.0	0.00	17.8	0.5	5.0	0.00	16.3	0.6	5.0	0.00	1.3
$m = 3$		T	#	G	R	T	#	G	R	T	#	G	R	T	#	G	R	T	#	G	R
	U	0.4	5	0.00	1.5	2,229.4	3	0.23	18.7	2,249.4	3	0.23	18.7	2,225.4	3	0.23	18.7	0.4	5	0.00	1.5
$\Gamma^a_i = 0$	C	0.9	5	0.00	3.1	3,600.0	0	1.07	26.4	3,600.0	0	1.07	26.4	3,600.0	0	1.07	26.4	1,898.0	4	0.02	3.1
	K	0.3	5	0.00	1.5	1,324.4	4	0.07	16.6	1,324.4	4	0.07	16.6	1,324.3	4	0.07	16.6	0.3	5	0.00	1.5
	U	1.2	5	0.00	1.8	3,600.0	0	0.53	28.6	3,600.0	0	0.77	27.7	3,600.0	0	0.43	26.0	0.4	5	0.00	1.7
$\Gamma^a_i = 1$	C	0.9	5	0.00	2.6	3,600.0	0	0.77	41.8	3,600.0	0	0.88	41.1	3,600.0	0	0.78	40.1	833.2	4	0.01	2.8
	K	1.3	5	0.00	1.6	2,170.8	2	0.38	19.7	2,198.6	2	0.39	19.3	2,169.6	2	0.34	17.8	0.9	5	0.00	1.6
	U	1.4	5	0.00	1.6	2,982.0	1	0.56	34.4	3,600.0	0	0.62	34.4	3,064.0	1	0.47	31.5	0.3	5	0.00	1.6
$\Gamma^a_i = 2$	C	2.0	5	0.00	2.2	3,600.0	0	0.85	18.9	3,600.0	0	0.97	18.8	3,600.0	0	0.85	18.1	1,420.0	5	0.00	2.4
	K	2.3	5	0.00	1.5	1,202.7	4	0.14	19.4	2,194.6	3	0.24	18.9	1,204.8	4	0.11	17.5	0.4	5	0.00	1.4
	U	1.1	5	0.00	1.5	2,214.3	2	0.40	26.1	2,420.3	2	0.52	27.3	2,340.3	2	0.40	24.7	0.7	5	0.00	1.6
$\Gamma^a_i = 5$	C	12.6	5	0.00	2.6	3,600.0	0	0.89	42.3	3,600.0	0	0.86	41.8	3,600.0	0	0.85	38.3	2,190.4	2	0.03	3.0
	K	2.3	5	0.00	1.6	2,800.0	2	0.21	25.5	3,580.0	1	0.48	26.7	2,960.0	1	0.23	21.7	5.9	5	0.00	1.6
	U	1.3	5	0.00	1.7	2,900.0	1	0.41	30.6	3,260.0	1	0.56	32.8	2,948.0	1	0.42	28.4	0.4	5	0.00	1.7
$\Gamma^a_i = 10$	C	29.6	5	0.00	2.4	3,600.0	0	0.97	26.9	3,600.0	0	1.17	27.1	3,600.0	0	0.81	22.7	1,242.0	5	0.00	2.6
	K	0.8	5	0.00	1.5	2,040.2	3	0.16	15.2	2,880.3	1	0.37	16.0	2,360.2	3	0.16	14.3	0.5	5	0.00	1.6
Average		3.9	5.0	0.00	1.9	2,764.3	1.5	0.51	26.1	3,020.5	1.1	0.61	26.2	2,813.1	1.4	0.48	24.2	506.2	4.7	0.00	2.0
$m = 5$		T	#	G	R	T	#	G	R	T	#	G	R	T	#	G	R	T	#	G	R
	U	8.0	5	0.00	1.9	2,883.6	1	0.67	24.0	2,883.6	1	0.67	24.0	2,883.6	1	0.67	24.0	1.0	5	0.00	1.9
$\Gamma^a_i = 0$	C	20.5	5	0.00	3.9	3,600.0	0	1.56	46.7	3,600.0	0	1.56	46.7	3,600.0	0	1.56	46.7	3,600.0	0	0.16	3.9
	K	10.9	5	0.00	1.8	2,884.2	1	0.55	18.7	2,884.2	1	0.55	18.7	2,884.2	1	0.55	18.7	1.2	5	0.00	1.8
	U	307.0	5	0.00	1.9	2,390.0	2	0.75	34.7	2,374.0	2	0.74	34.4	2,390.0	2	0.69	32.2	10.6	5	0.00	2.0
$\Gamma^a_i = 1$	C	24.0	5	0.00	2.6	3,600.0	0	1.07	26.3	3,600.0	0	1.10	26.1	3,600.0	0	1.12	25.5	3,180.0	1	0.09	2.9
	K	132.2	5	0.00	1.9	2,916.0	1	0.51	22.1	2,926.0	1	0.56	22.0	2,968.0	1	0.54	21.7	78.6	5	0.00	1.9
	U	111.6	5	0.00	1.8	2,892.4	1	0.85	28.8	2,894.0	1	0.88	29.0	2,902.0	1	0.81	27.5	8.4	5	0.00	1.8
$\Gamma^a_i = 2$	C	67.3	5	0.00	3.1	3,600.0	0	1.18	32.9	3,600.0	0	1.14	32.7	3,600.0	0	1.18	31.0	3,600.0	0	0.10	3.4
	K	297.0	5	0.00	1.9	3,600.0	0	0.97	25.3	3,600.0	0	0.94	25.4	3,600.0	0	0.96	24.1	114.2	5	0.00	2.0
	U	52.2	5	0.00	1.8	3,600.0	0	0.89	29.1	3,600.0	0	0.98	30.4	3,600.0	0	0.82	26.4	2.5	5	0.00	2.0
$\Gamma^a_i = 5$	C	513.0	5	0.00	2.6	3,600.0	0	1.20	33.0	3,600.0	0	1.16	32.7	3,600.0	0	1.15	29.7	3,600.0	0	0.07	2.9
	K	25.4	5	0.00	1.5	1,503.8	3	0.20	22.4	1,559.6	3	0.24	23.6	1,770.8	3	0.27	23.2	1.1	5	0.00	1.8
	U	59.6	5	0.00	1.7	2,207.2	2	0.38	26.7	2,244.0	2	0.48	28.5	2,394.0	2	0.40	25.8	20.4	5	0.00	1.8
$\Gamma^a_i = 10$	C	751.6	5	0.00	3.5	3,600.0	0	1.28	38.6	3,600.0	0	1.22	38.4	3,600.0	0	1.14	32.8	3,600.0	0	0.10	3.5
	K	51.0	5	0.00	1.8	2,880.0	2	0.27	21.1	3,020.0	2	0.31	22.6	3,600.0	0	0.36	20.3	0.8	5	0.00	1.9
Average		162.1	5.0	0.00	2.2	3,050.5	0.9	0.82	28.7	3,065.7	0.9	0.84	29.0	3,132.8	0.7	0.81	27.3	1,187.9	3.4	0.03	2.4

Table 10: Comparison of results for the best disjoint reformulation $\text{MILP}^{\log}_{2'}[\mathcal{U}^{ab}]$ versus joint reformulations ($\text{MILP}_2[\mathcal{U}^{\overline{ab}}]$, $\text{MILP}_2[\mathcal{U}^{\overline{ab}}_{=}]$, $\text{MILP}_2[\mathcal{U}^{\overline{ab}}_{\propto}]$ and $\text{MILP}[\mathcal{U}^{\overline{a}}]$). Average time (T) in seconds with the number (#) of instances solved within default optimality gap 0.01, and the average remaining optimality gap (G) along with the average relaxation quality (R) across instances for $n = 100$. We have: for $\mathcal{U}^{\overline{ab}}$ that $\Gamma_i = 2\,\Gamma^a_i$, for $\mathcal{U}^{\overline{ab}}_{=}$ and $\mathcal{U}^{\overline{ab}}_{\propto}$ that $\Gamma_i = \Gamma^a_i$, and for $\mathcal{U}^{\overline{a}}$ that $\Gamma = m\,\Gamma^a_i$.

$n = 100$	Cons.	$\text{MILP}^{\log}_{2'}[\mathcal{U}^{ab}]$				$\text{MILP}_2[\mathcal{U}^{\overline{ab}}]$				$\text{MILP}_2[\mathcal{U}^{\overline{ab}}_{=}]$				$\text{MILP}_2[\mathcal{U}^{\overline{ab}}_{\propto}]$				$\text{MILP}[\mathcal{U}^{\overline{a}}]$			
$m = 1$	type	T	#	G	R	T	#	G	R	T	#	G	R	T	#	G	R	T	#	G	R
	U	0.1	5	0.00	1.0	0.4	5	0.00	20.9	0.5	5	0.00	20.9	0.4	5	0.00	20.9	0.1	5	0.00	1.0
$\Gamma^a_i = 0$	C	0.1	5	0.00	2.8	1.0	5	0.00	64.9	1.1	5	0.00	64.9	1.1	5	0.00	64.9	3,046.0	1	0.10	2.8
	K	0.0	5	0.00	1.0	0.5	5	0.00	14.4	0.6	5	0.00	14.4	0.6	5	0.00	14.4	0.0	5	0.00	1.0
	U	0.2	5	0.00	1.0	1.3	5	0.00	25.7	1.0	5	0.00	25.3	1.1	5	0.00	24.0	0.1	5	0.00	1.0
$\Gamma^a_i = 2$	C	0.2	5	0.00	2.7	27.8	5	0.00	65.0	8.3	5	0.00	63.8	21.7	5	0.00	63.4	3,600.0	1	0.16	3.0
	K	0.1	5	0.00	1.0	1.1	5	0.00	16.9	1.0	5	0.00	15.9	1.0	5	0.00	15.1	0.1	5	0.00	1.0
	U	0.1	5	0.00	1.0	1.1	5	0.00	34.3	1.3	5	0.00	33.6	1.1	4	0.28	30.9	0.1	5	0.00	1.0
$\Gamma^a_i = 4$	C	0.1	5	0.00	2.3	247.1	5	0.00	80.5	108.3	5	0.00	79.0	56.0	5	0.00	76.8	2,340.0	2	0.14	2.8
	K	0.2	5	0.00	1.1	1.2	5	0.00	23.4	1.0	5	0.00	23.2	1.0	5	0.00	21.8	0.1	5	0.00	1.0
	U	0.2	5	0.00	1.0	1.0	5	0.00	38.9	1.0	5	0.00	39.6	0.9	5	0.00	34.3	0.1	5	0.00	1.0
$\Gamma^a_i = 10$	C	0.3	5	0.00	2.4	128.3	5	0.00	95.7	42.2	5	0.00	93.1	81.9	5	0.00	86.7	3,600.0	0	0.21	3.1
	K	0.1	5	0.00	1.0	1.2	5	0.00	18.6	1.1	5	0.00	18.9	1.1	5	0.00	17.2	0.0	5	0.00	1.0
	U	0.1	5	0.00	1.0	1.3	5	0.00	30.1	1.3	5	0.00	31.3	1.2	5	0.00	29.6	0.1	5	0.00	1.0
$\Gamma^a_i = 20$	C	0.3	5	0.00	2.9	11.8	5	0.00	73.2	12.3	5	0.00	69.4	74.0	5	0.00	64.4	3,440.0	1	0.14	3.0
	K	0.1	5	0.00	1.0	1.0	5	0.00	33.5	0.9	5	0.00	35.2	1.1	5	0.00	30.6	0.0	5	0.00	1.0
Average		0.1	5.0	0.00	1.5	28.4	5.0	0.00	42.4	12.1	5.0	0.00	41.9	16.3	4.9	0.02	39.7	1,068.5	3.7	0.05	1.6
$m = 3$		T	#	G	R	T	#	G	R	T	#	G	R	T	#	G	R	T	#	G	R
	U	0.9	5	0.00	1.9	3,600.0	0	2.48	31.5	3,600.0	0	2.50	31.5	3,600.0	0	2.48	31.5	6.4	5	0.00	1.9
$\Gamma^a_i = 0$	C	1.5	5	0.00	3.3	3,600.0	0	4.22	35.4	3,600.0	0	4.22	35.4	3,600.0	0	4.22	35.4	3,600.0	0	0.49	3.3
	K	0.6	5	0.00	1.7	3,160.0	1	1.30	31.5	3,160.0	1	1.30	31.5	3,160.0	1	1.30	31.5	1.7	5	0.00	1.7
	U	1.3	5	0.00	1.7	3,600.0	0	2.27	47.6	3,600.0	0	2.24	46.8	3,600.0	0	2.22	44.4	1.1	5	0.00	1.7
$\Gamma^a_i = 2$	C	5.0	5	0.00	3.5	3,600.0	0	4.76	66.2	3,600.0	0	4.84	65.8	3,600.0	0	4.94	64.9	3,600.0	0	0.54	4.1
	K	2.3	5	0.00	1.6	3,600.0	0	1.96	33.0	3,600.0	0	1.90	32.1	3,600.0	0	1.89	31.8	1.8	5	0.00	1.6
	U	4.8	5	0.00	1.8	3,600.0	0	3.54	48.3	3,600.0	0	3.60	48.0	3,600.0	0	3.14	44.5	8.9	5	0.00	1.9
$\Gamma^a_i = 4$	C	5.1	5	0.00	2.7	3,600.0	0	4.62	60.3	3,600.0	0	4.40	59.2	3,600.0	0	4.40	57.9	3,600.0	0	0.46	3.5
	K	7.8	5	0.00	2.0	3,600.0	0	3.68	32.2	3,600.0	0	3.44	30.6	3,600.0	0	3.24	28.9	26.3	5	0.00	1.9
	U	2.4	5	0.00	1.6	3,600.0	0	3.38	52.9	3,600.0	0	3.18	54.0	3,600.0	0	2.90	49.9	2.1	5	0.00	1.8
$\Gamma^a_i = 10$	C	407.9	5	0.00	2.9	3,600.0	0	5.02	85.1	3,600.0	0	4.80	83.0	3,600.0	0	4.84	79.7	3,600.0	0	0.46	3.4
	K	0.8	5	0.00	1.4	3,600.0	0	1.43	39.8	3,600.0	0	1.48	41.1	3,600.0	0	1.56	40.3	0.6	5	0.00	1.6
	U	3.3	5	0.00	1.7	3,600.0	0	4.26	53.1	3,600.0	0	4.40	55.7	3,600.0	0	4.00	50.5	3.3	5	0.00	1.9
$\Gamma^a_i = 20$	C	1,273.2	5	0.00	2.8	3,600.0	0	5.10	61.3	3,600.0	0	5.10	60.4	3,600.0	0	4.86	54.4	3,600.0	0	0.47	3.5
	K	2.2	5	0.00	1.8	3,600.0	0	2.90	28.0	3,600.0	0	2.94	28.9	3,600.0	0	2.62	26.5	20.0	5	0.00	1.8
Average		114.6	5.0	0.00	2.2	3,570.7	0.1	3.39	47.1	3,570.7	0.1	3.36	46.9	3,570.7	0.1	3.24	44.8	1,204.8	3.3	0.16	2.4
$m = 5$		T	#	G	R	T	#	G	R	T	#	G	R	T	#	G	R	T	#	G	R
	U	24.0	5	0.00	2.1	3,600.0	0	2.98	41.5	3,600.0	0	2.98	41.6	3,600.0	0	2.98	41.6	262.8	5	0.00	2.1
$\Gamma^a_i = 0$	C	393.8	5	0.00	4.3	3,600.0	0	5.92	70.6	3,600.0	0	5.94	70.6	3,600.0	0	5.92	70.7	3,600.0	0	0.67	4.3
	K	21.8	5	0.00	2.0	3,600.0	0	1.21	23.7	3,600.0	0	1.21	23.7	3,600.0	0	1.21	23.7	197.8	5	0.00	2.0
	U	206.4	5	0.00	2.1	3,600.0	0	4.96	52.8	3,600.0	0	4.62	49.9	3,600.0	0	4.38	47.6	159.3	5	0.00	2.1
$\Gamma^a_i = 2$	C	1,299.6	5	0.00	3.4	3,600.0	0	5.64	53.3	3,600.0	0	5.80	53.3	3,600.0	0	5.66	52.6	3,600.0	0	0.68	3.9
	K	167.2	5	0.00	2.2	3,600.0	0	2.12	25.1	3,600.0	0	2.14	24.5	3,600.0	0	2.04	23.5	773.6	4	0.01	2.2
	U	480.0	5	0.00	2.2	3,600.0	0	5.40	67.2	3,600.0	0	5.66	67.7	3,600.0	0	4.84	58.4	10.3	5	0.00	2.1
$\Gamma^a_i = 4$	C	431.4	5	0.00	3.3	3,600.0	0	6.66	78.6	3,600.0	0	5.90	77.8	3,600.0	0	6.60	76.0	3,600.0	0	0.68	4.1
	K	378.2	5	0.00	2.1	3,600.0	0	1.98	33.3	3,600.0	0	2.06	33.4	3,600.0	0	1.94	31.8	786.6	4	0.01	2.2
	U	1,524.8	3	0.02	2.1	3,600.0	0	5.72	67.4	3,600.0	0	5.52	69.3	3,600.0	0	5.16	62.0	779.0	4	0.03	2.3
$\Gamma^a_i = 10$	C	3,600.0	4	0.01	3.1	3,600.0	0	6.16	68.4	3,600.0	0	6.14	68.0	3,600.0	0	6.66	64.6	3,600.0	0	0.65	4.0
	K	964.0	4	0.01	1.7	3,600.0	0	1.56	27.5	3,600.0	0	1.66	28.8	3,600.0	0	1.58	26.7	723.7	4	0.01	1.9
	U	182.0	5	0.00	1.9	3,600.0	0	4.70	53.4	3,600.0	0	4.86	56.2	3,600.0	0	4.36	50.5	180.9	5	0.00	2.2
$\Gamma^a_i = 20$	C	3,600.0	1	0.01	3.5	3,600.0	0	6.88	61.7	3,600.0	0	6.36	61.3	3,600.0	0	6.38	55.6	3,600.0	0	0.67	4.0
	K	238.2	5	0.00	2.0	3,600.0	0	2.20	23.3	3,600.0	0	2.24	25.1	3,600.0	0	1.86	21.3	969.5	5	0.00	2.2
Average		900.8	4.5	0.00	2.5	3,600.0	0.0	4.27	49.9	3,600.0	0.0	4.21	50.1	3,600.0	0.0	4.10	47.1	1,522.9	3.1	0.23	2.8

Table 11: Comparison of results for the best disjoint reformulation $\text{MILP}_{2'}^{\log}[\mathcal{U}^{ab}]$ versus joint reformulations ($\text{MILP}_2[\mathcal{U}^{\overline{ab}}]$, $\text{MILP}_2[\mathcal{U}_{=}^{\overline{ab}}]$, $\text{MILP}_2[\mathcal{U}_{\propto}^{\overline{ab}}]$ and $\text{MILP}[\mathcal{U}^{\overline{a}}]$). Average time (T) in seconds with the number (#) of instances solved within default optimality gap 0.01, and the average remaining optimality gap (G) along with the average relaxation quality (R) across instances for $n = 150$. We have: for $\mathcal{U}^{\overline{ab}}$ that $\Gamma_i = 2\,\Gamma_i^a$, for $\mathcal{U}_{=}^{\overline{ab}}$ and $\mathcal{U}_{\propto}^{\overline{ab}}$ that $\Gamma_i = \Gamma_i^a$, and for $\mathcal{U}^{\overline{a}}$ that $\Gamma = m\,\Gamma_i^a$.

$n = 150$	Cons.	$\text{MILP}_{2'}^{\log}[\mathcal{U}^{ab}]$				$\text{MILP}_2[\mathcal{U}^{\overline{ab}}]$				$\text{MILP}_2[\mathcal{U}_{=}^{\overline{ab}}]$				$\text{MILP}_2[\mathcal{U}_{\propto}^{\overline{ab}}]$				$\text{MILP}[\mathcal{U}^{\overline{a}}]$			
$m = 1$	type	T	#	G	R	T	#	G	R	T	#	G	R	T	#	G	R	T	#	G	R
	U	0.1	5	0.00	1.0	0.4	5	0.00	29.6	0.4	5	0.00	29.6	0.4	5	0.00	29.6	0.1	5	0.00	1.0
$\Gamma_i^a = 0$	C	0.1	5	0.00	2.4	30.5	5	0.00	45.6	170.6	5	0.00	45.6	30.5	5	0.00	45.6	3,600.0	0	0.32	2.4
	K	0.1	5	0.00	1.0	1.0	5	0.00	20.7	0.9	5	0.00	20.7	0.9	5	0.00	20.7	0.1	5	0.00	1.0
	U	0.2	5	0.00	1.0	2.8	5	0.00	35.4	3.7	5	0.00	33.5	3.1	5	0.00	32.7	0.2	5	0.00	1.0
$\Gamma_i^a = 3$	C	0.2	5	0.00	2.0	41.4	5	0.00	31.6	68.0	5	0.00	31.1	740.6	4	0.01	30.9	3,600.0	0	0.34	2.3
	K	0.1	5	0.00	1.0	1.3	5	0.00	27.9	2.9	5	0.00	25.5	1.1	5	0.00	24.2	0.1	5	0.00	1.0
	U	0.2	5	0.00	1.1	2.5	5	0.00	39.6	3.7	5	0.00	37.8	2.7	5	0.00	36.6	0.1	5	0.00	1.0
$\Gamma_i^a = 6$	C	0.2	5	0.00	2.4	790.4	4	0.70	90.2	1,451.6	3	0.07	87.7	751.8	4	0.30	86.1	3,600.0	0	0.34	3.1
	K	0.2	5	0.00	1.1	1.7	5	0.00	19.4	2.3	5	0.00	18.1	1.3	5	0.00	16.9	0.1	5	0.00	1.0
	U	0.2	5	0.00	1.0	2.1	5	0.00	46.6	3.0	5	0.00	46.8	1.7	5	0.00	42.0	0.1	5	0.00	1.0
$\Gamma_i^a = 15$	C	0.3	5	0.00	1.8	1,451.7	5	0.00	46.8	24.4	5	0.00	45.5	1,843.6	4	0.00	43.7	3,600.0	0	0.33	2.3
	K	0.2	5	0.00	1.0	2.0	5	0.00	42.7	3.1	5	0.00	43.0	1.9	5	0.00	35.7	0.1	5	0.00	1.0
	U	0.2	5	0.00	1.0	2.4	5	0.00	38.3	3.5	5	0.00	39.3	2.3	5	0.00	31.2	0.1	5	0.00	1.0
$\Gamma_i^a = 30$	C	0.4	5	0.00	2.1	690.5	5	0.00	71.2	823.7	4	0.46	68.7	1,539.2	3	0.70	63.9	3,600.0	0	0.40	2.4
	K	0.2	5	0.00	1.0	2.1	5	0.00	25.3	3.4	5	0.00	26.2	1.5	5	0.00	23.7	0.1	5	0.00	1.0
Average		0.2	5.0	0.00	1.4	201.5	4.9	0.05	40.7	171.0	4.8	0.04	39.9	328.2	4.7	0.07	37.6	1,200.1	3.3	0.12	1.5
$m = 3$		T	#	G	R	T	#	G	R	T	#	G	R	T	#	G	R	T	#	G	R
	U	0.7	5	0.00	1.8	3,600.0	0	3.64	46.0	3,600.0	0	3.64	46.0	3,600.0	0	3.64	46.0	721.0	4	0.01	1.8
$\Gamma_i^a = 0$	C	3.2	5	0.00	5.1	3,600.0	0	9.66	118.7	3,600.0	0	9.74	118.7	3,600.0	0	9.66	118.7	3,600.0	0	1.00	5.1
	K	0.8	5	0.00	1.7	3,600.0	0	2.86	38.9	3,600.0	0	2.86	38.9	3,600.0	0	2.86	38.9	2.9	5	0.00	1.7
	U	4.1	5	0.00	1.8	3,600.0	0	5.26	71.9	3,600.0	0	5.18	67.4	3,600.0	0	4.94	64.8	46.5	5	0.00	1.8
$\Gamma_i^a = 3$	C	30.8	5	0.00	3.9	3,600.0	0	9.42	109.8	3,600.0	0	9.40	109.3	3,600.0	0	10.04	108.1	3,600.0	0	0.91	4.5
	K	5.2	5	0.00	1.9	3,600.0	0	4.88	48.9	3,600.0	0	4.98	48.2	3,600.0	0	4.58	45.2	57.2	5	0.00	1.9
	U	5.4	5	0.00	2.0	3,600.0	0	7.04	71.5	3,600.0	0	6.48	70.5	3,600.0	0	6.18	65.8	322.0	5	0.00	1.9
$\Gamma_i^a = 6$	C	29.6	5	0.00	3.1	3,600.0	0	8.92	56.1	3,600.0	0	8.04	55.5	3,600.0	0	8.44	55.3	3,600.0	0	0.83	3.6
	K	6.9	5	0.00	1.8	3,600.0	0	4.67	38.8	3,600.0	0	4.26	37.7	3,600.0	0	3.90	35.0	375.8	5	0.00	1.8
	U	13.7	5	0.00	1.8	3,600.0	0	7.52	89.6	3,600.0	0	7.40	91.2	3,600.0	0	6.28	73.3	10.9	5	0.00	1.8
$\Gamma_i^a = 15$	C	49.9	5	0.00	2.9	3,600.0	0	9.04	61.4	3,600.0	0	8.56	60.6	3,600.0	0	8.72	58.5	3,600.0	0	0.83	3.7
	K	16.5	5	0.00	1.9	3,600.0	0	5.94	44.0	3,600.0	0	5.72	44.4	3,600.0	0	5.20	40.8	830.6	4	0.01	1.9
	U	30.0	5	0.00	1.7	3,600.0	0	6.54	56.7	3,600.0	0	6.58	59.6	3,600.0	0	6.04	53.2	722.6	4	0.01	1.8
$\Gamma_i^a = 30$	C	1,448.6	5	0.00	3.1	3,600.0	0	9.40	62.6	3,600.0	0	9.50	62.0	3,600.0	0	9.48	58.2	3,600.0	0	0.80	3.8
	K	204.1	5	0.00	1.6	3,600.0	0	5.32	45.3	3,600.0	0	5.02	47.2	3,600.0	0	4.52	41.7	761.3	4	0.00	1.8
Average		123.3	5.0	0.00	2.4	3,600.0	0.0	6.67	64.0	3,600.0	0.0	6.49	63.8	3,600.0	0.0	6.30	60.2	1,456.7	3.1	0.29	2.6
$m = 5$		T	#	G	R	T	#	G	R	T	#	G	R	T	#	G	R	T	#	G	R
	U	32.6	5	0.00	2.4	3,600.0	0	7.18	58.0	3,600.0	0	7.20	58.0	3,600.0	0	7.18	58.0	1,983.8	3	0.06	2.4
$\Gamma_i^a = 0$	C	666.6	5	0.00	4.5	3,600.0	0	11.58	63.6	3,600.0	0	11.58	63.6	3,600.0	0	11.58	63.7	3,600.0	0	1.18	4.5
	K	34.7	5	0.00	2.3	3,600.0	0	5.08	40.6	3,600.0	0	5.32	40.6	3,600.0	0	5.08	40.6	2,164.6	2	0.09	2.3
	U	960.0	5	0.00	2.5	3,600.0	0	10.08	95.2	3,600.0	0	9.84	91.5	3,600.0	0	9.24	83.4	2,172.4	2	0.14	2.5
$\Gamma_i^a = 3$	C	2,302.0	5	0.00	4.0	3,600.0	0	12.00	105.3	3,600.0	0	12.20	105.1	3,600.0	0	11.80	104.1	3,600.0	0	1.10	4.7
	K	734.0	5	0.00	2.4	3,600.0	0	7.70	61.1	3,600.0	0	6.94	57.5	3,600.0	0	7.06	55.3	2,882.6	1	0.12	2.5
	U	1,578.6	4	0.00	2.2	3,600.0	0	8.94	81.3	3,600.0	0	9.12	79.5	3,600.0	0	7.90	73.4	1,447.0	3	0.13	2.3
$\Gamma_i^a = 6$	C	1,608.8	5	0.00	4.3	3,600.0	0	13.40	159.5	3,600.0	0	12.40	159.0	3,600.0	0	13.60	156.1	3,600.0	0	1.22	5.3
	K	417.0	5	0.00	2.0	3,600.0	0	6.10	55.7	3,600.0	0	5.44	54.8	3,600.0	0	5.16	53.3	943.3	4	0.06	2.1
	U	1,018.0	4	0.02	1.9	3,600.0	0	8.44	77.1	3,600.0	0	8.76	77.6	3,600.0	0	8.44	70.9	809.2	4	0.09	2.1
$\Gamma_i^a = 15$	C	3,600.0	4	0.01	3.5	3,600.0	0	13.80	100.5	3,600.0	0	13.40	100.0	3,600.0	0	13.20	96.1	3,600.0	0	1.08	4.4
	K	1,806.0	3	0.03	2.0	3,600.0	0	9.60	71.1	3,600.0	0	9.84	73.1	3,600.0	0	8.24	62.9	2,129.0	3	0.19	2.6
	U	306.0	5	0.00	1.9	3,600.0	0	8.00	79.1	3,600.0	0	8.74	81.9	3,600.0	0	7.54	74.8	291.4	5	0.00	2.2
$\Gamma_i^a = 30$	C	3,600.0	1	0.01	3.5	3,600.0	0	13.20	102.9	3,600.0	0	13.40	102.8	3,600.0	0	13.00	95.5	3,600.0	0	1.03	4.5
	K	1,056.0	4	0.02	1.9	3,600.0	0	7.36	68.0	3,600.0	0	7.44	70.3	3,600.0	0	7.30	64.1	752.3	4	0.08	2.1
Average		1,314.7	4.3	0.01	2.8	3,600.0	0.0	9.50	81.3	3,600.0	0.0	9.44	81.0	3,600.0	0.0	9.09	76.8	2,238.4	2.1	0.44	3.1

4.0 Solving a Class of Feature Selection Problems via Fractional 0-1 Programming

4.1 Introduction

An essential preprocessing step for many data mining and machine learning tasks is the data set dimensionality reduction that can be performed either by sample or feature set reductions. In this chapter, we focus on the latter procedure as a high number of features may cause model overfitting, which results in poor validation results [23, 50].

Formally, a *feature* is a single measurable property of a process being observed. *Feature selection* is the process of identifying a subset of the most informative data features from the original feature set to include in a statistical model.

Feature selection is often used in many machine learning and pattern recognition settings that deal with large data sets including classification, clustering, and regression tasks. The corresponding applications arise in diverse areas such as e-commerce [102], medical diagnosis [34], bioinformatics [82] and biomedicine [21, 22, 52], among others. Moreover, apart from data dimensionality reduction, feature selection has many other potential side benefits including facilitating data visualization, decreasing training and utilization (computational) times, reducing the measurement and storage requirements, and improving noise to achieve a better prediction performance. We refer to [23, 40, 50, 91] and the references therein for an overview of applications and methods for feature selection.

In general, feature selection procedures are classified into three major categories, namely, filter, wrapper, and hybrid (embedded) methods [23, 50]. Wrapper and hybrid methods involve learning algorithms and the selection process is tailored based on the chosen algorithm [98]. In contrast, filter methods are not linked with any learning algorithm and are often a more appropriate choice for large-sized data sets [50, 69].

The main focus of this chapter is on the filter methods. These methods select a subset of features by evaluating them according to some predefined measures. The measures typically applied in the literature can be categorized into information, distance, similarity, consistency, and statistical-based ones [50]. In this chapter, we consider measures for the classification task in supervised learning wherein we are given a training data set. In this set, the classification of each sample is known. Then the aim is to predict unknown classes of new samples employing the information provided by the training data set. To this end, it is important to distinguish *relevant* features from *redundant* ones, and thus a desired measure (for feature selection) needs to differentiate the former from the latter. Relevant features are those that provide useful information for predicting the class of each given sample. Redundant features are either weakly informative for this predication or can be replaced with a set of some other relevant features.

The relevancy and redundancy are often characterized in terms of correlation and mutual information, which are widely used statistical tools to define the dependency of random variables [73]. The studies in [30, 73] and [41] propose a mutual-information-based and a correlation-based feature selection measures, called *minimal redundancy maximal relevance* (mRMR) and *correlation feature selection* (CFS), respectively. A key advantage of these two approaches is that they take into account the features' relevancy and redundancy simultaneously.

Once a measure is selected, a procedure must be developed to select a subset of features from the full feature set. Finding an optimal subset, i.e., a subset which has the best value for the considered measure (among exponentially many feature subsets) is often an NP-hard problem [23]. Hence, in order to find a high quality (but not necessarily an optimal) subset, various heuristic methods have been proposed in the literature based on the mRMR and CFS measures, see, e.g., [26, 30, 45, 56, 73, 101]. These heuristics are typically based on a (greedy) ranking of individual features with respect to the selected measure and then choosing a subset of the highest-ranking ones [23].

Nguyen et al. [67, 69] show that the mRMR and CFS feature selection problems can be posed as single-ratio polynomial fractional 0-1 programs (PFPs), where the objective function is a ratio of quadratic binary functions. The existing exact solution approaches for the mRMR and CFS problems are centered around their transformations into equivalent *mixed-integer linear programs* (MILPs). Notably, the PFPs of mRMR and CFS can be reformulated as MILPs either by exploiting the method of [24] or [67]; the latter method is also studied in [68, 69, 70]. These reformulations are based on the substitution of the denominator of the ratio with a continuous variable and then linearizing the resulting quadratic and cubic terms involving products of binary and at most one continuous variables.

Nevertheless, the single-ratio structure of the PFPs of the mRMR and CFS may allow us to use specialized approaches than the generic MILP reformulations. In particular, an alternative approach can be based on parametric algorithms; see [17, 46] for reviews of such algorithms. Applying parametric algorithms to solve mRMR and CFS involves solving a sequence of unconstrained binary quadratic problems (BQPs), which are also, in general, NP-hard [71]. However, due to recent advances in binary quadratic optimization softwares such as CPLEX [47] and Gurobi [39], reasonably sized BQPs can be solved efficiently [62]. Additionally, in the parametric algorithms solving BQPs to optimality may not be required and each iteration of the algorithms can be stopped when a feasible solution satisfying some conditions is found. This approach can lead to an improvement on the performance of the algorithms.

Contributions and the structure of the chapter. The aim of this chapter is to study exact approaches for the mRMR and CFS feature selection problems. Our main focus is on solution methods that can handle reasonably high-dimensional data sets, where the existing MILPs in the literature fail. To this end,

- In Section 4.2, we formally define mRMR and CFS measures and the corresponding fractional 0-1 optimization problems.
- In Section 4.3, first, we perform a comprehensive review of the existing MILP reformulations of the mRMR and CFS problems in the literature. Then by exploiting the structure of the fractional model of mRMR we propose a new MILP reformulation approach that outperforms the previous MILPs in the literature.
- In Section 4.4, we describe parametric methods such as binary-search [2, 53, 79] and Newton's method [31] algorithms for solving the mRMR and CFS problems.
- In Section 4.5, we conduct computational experiments with a collection of real data sets. From our results we observe that the performance of the existing MILPs in the literature is rather poor even for small- and medium-size problems. This observation is consistent with the earlier results in the literature [67, 69]. On the other hand, the parametric meth-

ods perform well across all considered problem sizes. We also provide some insights on the selection of an appropriate measure and solution method.

4.2 Problem formulations

In the supervised learning for the purpose of classification the input data is given as an $m \times (n+1)$ observation matrix, where m is the number of samples (observations). Each sample is a $(n+1)$-dimensional vector of n features, f_j, $j \in J = \{1, 2, \ldots, n\}$, and the label of the class that the sample belongs to.

The aim of classification is to predict the label of the target class variable, denoted by C, for a given sample that indicates the classification of the sample. Then the feature selection problem is to find a subset $S \subseteq \{f_1, f_2, \ldots, f_n\}$ such that the reduced $m \times (|S|+1)$ observation matrix provides sufficient information for a classification procedure to predict C. Throughout the chapter we let $\overline{C}$ denote the set of all possible labels for C, i.e., $C \in \overline{C}$. Next, we describe the mRMR and CFS feature selection measures and the corresponding optimization problems in Sections 4.2.1 and 4.2.2, respectively.

4.2.1 mRMR optimization problem

In the information theory, the mutual information (MI) quantifies the amount of information that a random variable provides about another one and it can be used as a measure of the mutual dependency between two random variables [73]. The notion of mutual information is related to the concept of entropy as the latter represents the uncertainty in the random variable. We refer to [58] for an additional discussion on the entropy and mutual information.

Formally, let X and Y be two discrete random variables. Then the entropy of variable X is defined as

$$\mathcal{H}(X) = -\sum_{x} \mathbb{P}(x) \log \mathbb{P}(x),$$

where $\mathbb{P}(x)$ is the probability that $X = x$. Moreover, the conditional entropy of X is given by

$$\mathcal{H}(X|Y) = -\sum_{x}\sum_{y} \mathbb{P}(x,y) \log \mathbb{P}(x|y),$$

which indicates the uncertainty that remains about X when we know the value of Y. Then the mutual information between X and Y, denoted by $\mathcal{I}(X,Y)$, is computed by

$$\mathcal{I}(X,Y) = \mathcal{H}(X) - \mathcal{H}(X|Y) = \mathcal{H}(Y) - \mathcal{H}(Y|X) = \sum_x \sum_y \mathbb{P}(x,y) \log\left[\frac{\mathbb{P}(x,y)}{\mathbb{P}(x)\mathbb{P}(y)}\right]. \tag{4.1}$$

Note that $\mathcal{I}(X,Y)$ has a non-negative value; if X and Y are independent then $\mathcal{I}(X,Y)$ is zero and a larger value of $\mathcal{I}(X,Y)$ indicates larger dependency between X and Y. Additionally, note that $\mathcal{I}(X,X) = \mathcal{H}(X)$. If X and Y are continuous variables, then similar definitions can be provided for $\mathcal{H}(X)$ and $\mathcal{I}(X,Y)$ by replacing the summations with integrations.

The task of feature selection using mRMR, proposed in [73], is to find the subset $S \subseteq \{1,\dots,n\}$, which has the maximum value for

$$\frac{1}{|S|}\sum_{f_j \in S} \mathcal{I}(f_j, C) - \frac{1}{|S|^2}\sum_{f_j, f_k \in S} \mathcal{I}(f_j, f_k), \tag{4.2}$$

over all 2^n possible feature subsets. The first term in (4.2) denotes the average MI between the features in set S and target class C, and thus, indicates the average relevancy of features in S. The second term denotes the average MI between features in S that also reflects the average redundancy of features in S.

In light of the above discussion, the maximization problem of (4.2) can be formulated as the fractional 0-1 program of the form [67]:

$$\text{(mRMR)} \qquad \max_{x \in \mathbb{B}^n} \left\{ \frac{\sum_{j \in J} \sum_{k \in J} \big(\mathcal{I}(f_j, C) - \mathcal{I}(f_j, f_k)\big) x_k x_j}{\sum_{j \in J} \sum_{k \in J} x_k x_j} \right\}, \tag{4.3}$$

where $\mathbb{B} := \{0,1\}$. Note that, $x_j = 1$ ($x_j = 0$) indicates the presence (absence) of feature f_j in set S.

4.2.2 CFS optimization problem

The mutual information is biased in favor of features that can take more number of values [101]. Moreover, for the purpose of comparing the degree of relevancy and redundancy of features normalized values (i.e., adjusted values to have the same scale) are preferred. An alternative measure that can be used as an indicator of the relevancy and redundancy is correlation. In fact, a feature is said to be relevant if it is highly correlated with the target class, and it is redundant if it is highly correlated with some other features. These interpretations lead to the hypothesis that "good feature sets contain features that are highly correlated with the class, yet uncorrelated with each other" [41].

The correlation – that is also referred to as symmetrical uncertainty [101] – between two random variables X and Y can be obtained by their scaled MI [74]:

$$\rho(X,Y) = \frac{2\mathcal{I}(X,Y)}{\mathcal{H}(X) + \mathcal{H}(Y)},$$

where $\rho(X,Y)$ compensates the bias in MI. Additionally, $\rho(X,Y) \in [0,1]$, where 0 denotes the independency of X and Y and a larger value implies some degree of dependency between these variables.

Then feature selection by means of CFS, proposed in [41], is to find subset S which has the maximum value for:

$$\frac{\sum_{f_j \in S} \rho(f_j, C)}{\sqrt{|S| + 2 \sum\limits_{\substack{f_j, f_k \in S, \\ j \neq k}} \rho(f_j, f_k)}}. \tag{4.4}$$

Relation (4.4) provides the correlation of subset S and the target class. The numerator of (4.4) is an indication of the relevancy (correlation) of features in S to the target class; its denominator encompasses both the size of $|S|$ and the redundancy (inter-correlation) of features in S.

In view of the above discussion, the maximization problem of (4.4) over all 2^n possible feature subsets can be posed as the fractional binary program of the form [68]:

$$\text{(CFS)} \qquad \max_{x \in \mathbb{B}^n} \left\{ \frac{\sum_{j \in J} \sum_{k \in J} \big(\rho(f_j, C) \cdot \rho(f_k, C)\big) x_k x_j}{\sum_{j \in J} x_j + \sum_{j \neq k} 2 \cdot \rho(f_j, f_k) x_k x_j} \right\}, \tag{4.5}$$

where $x_j = 1$ ($x_j = 0$) indicates the presence (absence) of feature f_j in set S.

4.3 Mixed-integer linear programming approaches

Both the mRMR and CFS feature selection problems given in (4.3) and (4.5), respectively, can be represented in the form of a single-ratio polynomial fractional 0-1 problem given by

$$\lambda^\star = \max_{x\in\mathbb{B}^n} \frac{f(x)}{g(x)} := \max_{x\in\mathbb{B}^n} \left\{ \frac{\sum_{j\in J} a_j x_j + \sum_{j\in J}\sum_{k\in J} b_{jk} x_j x_k}{\sum_{j\in J} c_j x_j + \sum_{j\in J}\sum_{k\in J} d_{jk} x_j x_k} \right\}, \tag{4.6}$$

where $a_j, b_{jk}, c_j, d_{jk} \in \mathbb{R}$, for all $j, k \in J := \{1, 2, \ldots, n\}$. Moreover, if $|S| \geqslant 1$, then the denominators of (4.3) and (4.5) are strictly positive; thus, throughout this chapter we assume that $g(x) > 0$.

Herein, we first review the existing MILP solution methods in the literature to solve (4.6). In particular, first, we apply the method proposed by Chang [24] to transform PFPs into MILPs, in order to reformulate (4.6) as an MILP, that is denoted by MILP_1 throughout this chapter; see Section 4.3.1. Second, we describe the approach of Nguyen et al. [67], denoted by MILP_2 throughout this chapter; see Section 4.3.2. Next, we propose two new MILP reformulations for (4.3), denoted by MILP_3 and MILP_4; see Section 4.3.3. Finally, in Section 4.3.4 we compare the sizes of the above MILPs.

4.3.1 Reformulation 1 (MILP_1)

We follow the approach of Chang [24] in transforming PFPs into MILPs. To this end, define

$$y := \frac{1}{\sum_{j\in J} c_j x_j + \sum_{j\in J}\sum_{k\in J} d_{jk} x_j x_k}. \tag{4.7}$$

Then the substitution with variable y in (4.6) yields

$$\max_{x\in\mathbb{B}^n, y} \quad \sum_{j\in J} a_j x_j y + \sum_{j\in J}\sum_{k\in J} b_{jk} x_j x_k y \tag{4.8a}$$

$$\text{s.t.} \quad \sum_{j\in J} c_j x_j y + \sum_{j\in J}\sum_{k\in J} d_{jk} x_j x_k y = 1. \tag{4.8b}$$

Since $x_j, x_k \in \mathbb{B}$, cubic terms $x_j x_k y$, for all $j, k \in J$, can be linearized as follows.

$$\Omega_{jk} := \big\{ (x_j, x_k, y, z_{jk}) \in \mathbb{B}^2 \times \mathbb{R}^2 \mid y^\ell x_j \leqslant z_{jk} \leqslant y^u x_j,\ y^\ell x_k \leqslant z_{jk} \leqslant y^u x_k,$$
$$y^u(x_j + x_k - 2) + y \leqslant z_{jk} \leqslant y^\ell(2 - x_j - x_k) + y \big\},$$

where y^ℓ and y^u are a lower bound and an upper bound on y, respectively, and note that $(x_j, x_k, y, z_{jk}) \in \Omega_{jk} \Leftrightarrow z_{jk} = x_j x_k y$. Similarity, we use $\overline{\Omega}_j$ as a variant of Ω_{jk} to linearize bilinear (quadratic) terms $x_j y$, for all $j \in J$; specifically,

$$\overline{\Omega}_j := \left\{(x_j, y, \bar{z}_j) \in \mathbb{B} \times \mathbb{R}^2 \mid y^\ell x_j \leqslant \bar{z}_j \leqslant y^u x_j,\ y^u(x_j - 1) + y \leqslant \bar{z}_j \leqslant y^\ell(1 - x_j) + y\right\},$$

and $(x_j, y, \bar{z}_j) \in \overline{\Omega}_j \Leftrightarrow \bar{z}_j = x_j y$.

Hence, non-linear (due to the presence of terms $x_j x_k y$ and $x_j y$) and non-convex (for $x \in [0,1]^n$) problem (4.8) is equivalent to MILP

$$(\text{MILP}_1) \qquad \begin{aligned} \max \quad & \sum_{j \in J} a_j \bar{z}_j + \sum_{j \in J} \sum_{k \in J} b_{jk} z_{jk} & \\ \text{s.t.} \quad & \sum_{j \in J} c_j \bar{z}_j + \sum_{j \in J} \sum_{k \in J} d_{jk} z_{jk} = 1 & \\ & (x_j, x_k, y, z_{jk}) \in \Omega_{jk} & \forall j \leqslant k \in J \\ & (x_j, y, \bar{z}_j) \in \overline{\Omega}_j & \forall j \in J. \end{aligned}$$

Let $a_j = c_j = 0$, $b_{jk} = \mathcal{I}(f_j, C) - \mathcal{I}(f_j, f_k)$, and $d_{jk} = 1$, for all $j, k \in J$, in MILP_1. Then we obtain an equivalent MILP of the mRMR feature selection problem (4.3). Similarly, in MILP_1, let $a_j = 0$, $b_{jk} = \rho(f_j, C) \cdot \rho(f_k, C)$, and $c_j = 1$, for all $j, k \in J$; additionally set $d_{jk} = 2\rho(f_j, f_k)$, for $j \neq k \in J$ and $d_{jk} = 0$, for $j = k \in J$. Then we obtain an equivalent MILP of the CFS feature selection problem (4.5).

4.3.2 Reformulation 2 (MILP_2)

Nguyen et al. [67] propose an alternative approach to transform (4.6) into an MILP described as follows. Note that problem (4.8) can be rewritten as

$$\max_{x \in \mathbb{B}^n, y} \quad \sum_{j \in J} a_j x_j y + \sum_{j \in J} \Big[\big(\sum_{k \in J} b_{jk} x_k\big) y\Big] x_j \tag{4.9a}$$

$$\text{s.t.} \quad \sum_{j \in J} c_j x_j y + \sum_{j \in J} \Big[\big(\sum_{k \in J} d_{jk} x_k\big) y\Big] x_j = 1, \tag{4.9b}$$

where y is given in (4.7).

Then define $v_j^b := [\sum_{k \in J} a_{jk} x_k y] x_j$ and $v_j^d := [\sum_{k \in J} b_{jk} x_k y] x_j$, for all $j \in J$. Observe that v_j^b and v_j^d are products of continuous terms, i.e., $\sum_{k \in J} b_{jk} x_k y$ and $\sum_{k \in J} d_{jk} x_k y$, respectively, and binary variable x_j. Hence, in contrast to the approach of Section 4.3.1 that directly linearizes cubic terms $x_k x_j y$ using Ω_{ij}, by employing the technique used in $\overline{\Omega}_j$ we first replace cubic terms with a set of constraints involving linear and bilinear terms.

$$
\begin{aligned}
\max_{x\in\mathbb{B}^n, y, v, \bar{v}} \quad & \sum_{j\in J} a_j x_j y + \sum_{j\in J} v_j^b && && (4.10a)\\
\text{s.t.} \quad & \sum_{j\in J} c_j x_j y + \sum_{j\in J} v_j^d = 1 && && (4.10b)\\
& -\mathcal{M}_j^b x_j \leqslant v_j^b \leqslant \mathcal{M}_j^b x_j && \forall j \in J && (4.10c)\\
& \mathcal{M}_j^b (x_j - 1) + \sum_{k\in J} b_{jk} x_k y \leqslant v_j^b \leqslant \mathcal{M}_j^b (1 - x_j) + \sum_{k\in J} b_{jk} x_k y && \forall j \in J && (4.10d)\\
& -\mathcal{M}_j^d x_j \leqslant v_j^d \leqslant \mathcal{M}_j^d x && \forall j \in J && (4.10e)\\
& \mathcal{M}_j^d (x_j - 1) + \sum_{k\in J} d_{ij} x_k y \leqslant v_j^d \leqslant \mathcal{M}_j^d (1 - x_j) + \sum_{k\in J} d_{ij} x_k y && \forall j \in J, && (4.10f)
\end{aligned}
$$

where $\mathcal{M}_j^b$ and $\mathcal{M}_j^d$ are sufficiently large values for all $j \in J$. Then to transform (4.10) to an MILP we can linearize bilinear terms $x_k y$, for all $k \in J$ by using $\overline{\Omega}_j$. Thus, we get

$$
\begin{aligned}
(\text{MILP}_2) \qquad \max \quad & \sum_{j\in J} a_j \bar{z}_j + \sum_{j\in J} v_j^b && \\
\text{s.t.} \quad & \sum_{j\in J} c_j \bar{z}_j + \sum_{j\in J} v_j^d = 1 && \\
& \mathcal{M}_j^b (x_j - 1) + \sum_{k\in J} b_{jk} \bar{z}_k \leqslant v_j^b \leqslant \mathcal{M}_j^b (1 - x_j) + \sum_{k\in J} b_{jk} \bar{z}_k && \forall j \in J\\
& -\mathcal{M}_j^b x_j \leqslant v_j^b \leqslant \mathcal{M}_j^b x && \forall j \in J\\
& \mathcal{M}_j^d (x_j - 1) + \sum_{k\in J} d_{jk} \bar{z}_k \leqslant v_j^d \leqslant \mathcal{M}_j^d (1 - x_j) + \sum_{k\in J} d_{jk} \bar{z}_k && \forall j \in J\\
& -\mathcal{M}^d x_j \leqslant v_j^d \leqslant \mathcal{M}_j^d x_j && \forall j \in J\\
& (x_j, y, \bar{z}_j) \in \overline{\Omega}_j && \forall j \in J.
\end{aligned}
$$

Let $a_j = c_j = 0$, $b_{jk} = \mathcal{I}(f_j, C) - \mathcal{I}(f_j, f_k)$, and $d_{jk} = 1$, for all $j, k \in J$, in MILP_2. Then we obtain an equivalent MILP of the mRMR feature selection problem (4.3). Similarly, in MILP_2, let $a_j = 0$, $b_{jk} = \rho(f_j, C) \cdot \rho(f_k, C)$, and $c_j = 1$, for all $j, k \in J$; additionally set $d_{jk} = 2\rho(f_j, f_k)$, for $j \neq k \in J$ and $d_{jk} = 0$, for $j = k \in J$. Then we obtain an equivalent MILP of the CFS feature selection problem (4.5).

4.3.3 New reformulations for mRMR (MILP_3 & MILP_4)

Here, we propose two new MILP reformulations for the mRMR problem given in (4.3) based on its special structure. Notably, the denominator of the objective function ratio in problem (4.3), i.e., $\sum_{j\in J}\sum_{k\in J} x_j x_k$, takes values in the set $\{1^2, 2^2, 3^2 \ldots, n^2\}$. Thus, using the standard value-disjunction approach we have

$$\frac{1}{\sum_j \sum_k x_k x_j} = \sum_{\ell\in J} \frac{1}{\ell^2} w_\ell,$$

where $w_\ell \in \mathbb{B}$ with $\sum_{\ell\in J} w_\ell = 1$ and $\sum_{j\in J} x_j = \sum_\ell \ell w_\ell$. Therefore, problem (4.3) can be reformulated as

$$\max_{x,w\in\mathbb{B}^n} \quad \sum_{\ell\in J}\sum_{j\in J}\sum_{k\in J} \frac{\mathcal{I}(f_j, C) - \mathcal{I}(f_j, f_k)}{\ell^2} x_k x_j w_\ell \tag{4.11a}$$

$$\text{s.t.} \quad \sum_{j\in J} x_j = \sum_{\ell\in J} \ell w_\ell \tag{4.11b}$$

$$\sum_{\ell\in J} w_\ell = 1. \tag{4.11c}$$

In order to transform (4.11) into an MILP, we define $u_{\ell jk} = x_k x_j w_\ell$ and use the technique of [36] to linearize cubic binary term $x_k x_j w_\ell$. The resulting MILP is

$$(\text{MILP}_3) \qquad \begin{aligned} \max_{x,w\in\mathbb{B}^n, u\geqslant 0} \quad & \sum_{\ell\in J}\sum_{j\in J}\sum_{k\in J} \frac{\mathcal{I}(f_j, C) - \mathcal{I}(f_j, f_k)}{\ell^2} u_{\ell jk} \\ \text{s.t.} \quad & \sum_{j\in J} x_j = \sum_{\ell\in J} \ell w_\ell \\ & \sum_{\ell\in J} w_\ell = 1 \\ & u_{\ell jk} \leqslant w_\ell, u_{\ell jk} \leqslant x_j, u_{\ell jk} \leqslant x_k \qquad \forall \ell \in J, \forall j \leqslant k \in J \\ & u_{\ell jk} \geqslant w_\ell + x_j + x_k - 2 \qquad \forall \ell \in J, \forall j \leqslant k \in J. \end{aligned}$$

An alternative approach to represent (4.11) as an MILP encompasses, first, the transformation of cubic expressions into bilinear terms, and then linearizing the latter. This approach is described as follows. Define $r := \sum_{j\in J}\sum_{k\in J} \big(\mathcal{I}(f_j, C) - \mathcal{I}(f_j, f_k)\big) x_k x_j$, then (4.11) can be written as

$$\max_{x,w\in\mathbb{B}^n,r} \quad \sum_{\ell\in J}\frac{1}{\ell^2} r w_\ell \tag{4.13a}$$

$$\text{s.t.} \quad r = \sum_{j\in J}\sum_{k\in J}\big(\mathcal{I}(f_j,C) - \mathcal{I}(f_j,f_k)\big)x_k x_j \tag{4.13b}$$

$$\sum_{j\in J} x_j = \sum_{\ell\in J} \ell w_\ell \tag{4.13c}$$

$$\sum_{\ell\in J} w_\ell = 1. \tag{4.13d}$$

Next, we introduce continuous variable $t_{jk} \coloneqq x_k x_j$ and use the technique of [36] to linearize binary quadratic term $x_k x_j$. Additionally, we define continuous variable $s_\ell \coloneqq r w_\ell$ and use a variant of $\overline{\Omega}_j$ to linearize $r w_\ell$. As a consequence, we get

(MILP$_4$)

$$\begin{aligned}
\max_{x,w\in\mathbb{B}^n,t\geqslant 0,s,r} \quad & \sum_{\ell\in J}\frac{1}{\ell^2} s_\ell \\
\text{s.t.} \quad & r = \sum_{j\in J}\sum_{k\in J}\big(\mathcal{I}(f_j,C) - \mathcal{I}(f_j,f_k)\big)t_{jk} \\
& \sum_{j\in J} x_j = \sum_{\ell\in J} \ell w_\ell \\
& \sum_{\ell\in J} w_\ell = 1 \\
& t_{jk} \leqslant x_j,\ t_{jk} \leqslant x_k,\ t_{jk} \geqslant x_j + x_k - 1 && \forall j \leqslant k \in J \\
& s_\ell \leqslant \mathcal{M} w_\ell, s_\ell \leqslant r + \mathcal{M}(1 - w_\ell) && \forall \ell \in J,
\end{aligned}$$

where $\mathcal{M}$ is a sufficiently large value. Note that since the MILP is in maximization form, upper-bounds on s_ℓ are sufficient.

4.3.4 Reformulations sizes

Table 12 shows the sizes (number of variables and constraints) of MILP reformulations presented in Sections 4.3.1, 4.3.2, and 4.3.3 for the feature selection problems (4.3) and (4.5). The sizes of MILP$_1$ and MILP$_2$ are $\mathcal{O}(n^2)$ and $\mathcal{O}(n)$, respectively. Thus, MILP$_2$ is significantly smaller than MILP$_1$, particularly in large instances. MILP$_3$ has the largest size among the MILPs provided for mRMR, both variables and constraints sizes are $\mathcal{O}(n^3)$; the size of MILP$_4$ is of the same order of magnitude as MILP$_1$.

Table 12: Sizes (number of variables and constraints) of MILP_1 to MILP_4 for the mRMR and CFS fractional 0-1 programs (4.3) and (4.5), respectively, where n is the total number of features.

Reformulation	Measure	Variables		Constraints
		Continous	Binary	
MILP_1 [24]	mRMR & CFS	$O(n^2)$	n	$O(n^2)$
MILP_2 [67]	mRMR & CFS	$O(n)$	n	$O(n)$
MILP_3	mRMR	$O(n^3)$	$2n$	$O(n^3)$
MILP_4	mRMR	$O(n^2)$	$2n$	$O(n^2)$

4.4 Parametric approaches

Parametric algorithms are typical solution methods to solve single-ratio fractional (either binary or continuous) programs; we refer to [17, 46] for a review of such algorithms. Simply speaking, parametric algorithms find an optimal solution of a single-ratio fractional problem by solving a sequence of non-fractional problems. In this section, we apply parametric approaches to solve problem (4.6).

Specifically, let $t \in \mathbb{R}$ be a parameter and consider the following parametric optimization problem.

$$v(t) = \max_{x \in \mathbb{B}^n} \Big\{ f(x) - t \cdot g(x) \Big\}, \tag{4.15}$$

where $f(x)$ and $g(x)$ are defined as in (4.6). Observe that, under the positive denominator assumption, i.e., $g(x) > 0$, function $v(t)$ is monotone and if $v(t) = 0$, then t is the optimal objective function value of (4.6), i.e., $t = \lambda^\star$. Otherwise, we have either $v(t) > 0$ or $v(t) < 0$, which indicates, respectively, that $t < \lambda^\star$ and $t > \lambda^\star$. Thus, problem (4.6) reduces to the problem of finding a root of function $v(t)$.

In particular, we use the well-known root-finding methods in order to find the optimal solution of (4.6) by solving a sequence of unconstrained quadratic 0-1 programs. We first discuss the binary-search method [53, 79] in Section 4.4.1, then we explain the Newton-like method [17, 31, 60] in Section 4.4.2.

4.4.1 Binary-search algorithm

Suppose that for the optimal objective function value $\lambda^\star$ at the beginning of iteration i of the algorithm an upper-bound, $\overline{\lambda}^i$, and a lower-bound, $\underline{\lambda}^i$, are given, i.e., it is known that $\lambda^\star \in [\underline{\lambda}^i, \overline{\lambda}^i]$. Then the binary-search algorithm [53, 79] evaluates $v(\lambda_M^i)$, where λ_M^i is the midpoint of the given interval, i.e., $\lambda_M^i = (\underline{\lambda}^i + \overline{\lambda}^i)/2$. If $v(\lambda_M^i) > 0$, then we update the lower-bound, $\underline{\lambda}^{i+1} = \lambda_M^i$; if $v(\lambda_M^i) < 0$, then we update upper-bound, $\overline{\lambda}^{i+1} = \lambda_M^i$; else, we have $v(\lambda_M^i) = 0$ and the midpoint λ_M^i is the optimal objective function value. The formal pseudo-code is given in Algorithm 1.

Algorithm 1 Binary-search algorithm

1: **Input:** ϵ_{rel}, relative gap parameter; ϵ_{abs}, absolute gap parameter;
2: **Output:** x; if $x_j = 1$, then feature j is selected
3: $i \leftarrow 0$
4: Compute $\overline{\lambda}^0$ and $\underline{\lambda}^0$
5: **while** time limit not exceeded & $|(\overline{\lambda}^i - \underline{\lambda}^i)/\underline{\lambda}^i| > \epsilon_{rel}$ & $|\overline{\lambda}^i - \underline{\lambda}^i| > \epsilon_{abs}$ **do**
6: $\quad\lambda_M^i \leftarrow (\underline{\lambda}^i + \overline{\lambda}^i)/2$
7: $\quad$Solve problem (4.15) for $t = \lambda_M^i$ and obtain $v(\lambda_M^i)$ and its optimal solution x^i
8: $\quad$**if** $v(\lambda_M^i) > 0$ **then**
9: $\quad\quad\underline{\lambda}^{i+1} \leftarrow \lambda_M^i$, $\overline{\lambda}^{i+1} \leftarrow \overline{\lambda}^i$
10: $\quad$**else if** $v(\lambda_M^i) < 0$ **then**
11: $\quad\quad\underline{\lambda}^{i+1} \leftarrow \underline{\lambda}^i$, $\overline{\lambda}^{i+1} \leftarrow \lambda_M^i$
12: $\quad$**else**
13: $\quad\quad$**return** x^i ▷ Optimal solution found
14: $\quad$**end if**
15: $\quad i \leftarrow i + 1$
16: **end while**
17: **return** x^i ▷ Best solution found within the time limit

Note that at each iteration of Algorithm 1 we can stop the optimization of problem (4.15) in line 7 whenever a feasible solution with a positive objective function value is found, which can potentially result in a better performance for the binary-search algorithm. In fact, mixed integer optimization algorithms find feasible and even optimal solutions in a portion of the time required to prove the optimality. Thus, if problem (4.15) is solved until the first feasible solution with positive objective function value is found, then in practice most of iterations except a few last ones are solved with a few branch-and-bound nodes. Although this approach may require more iterations, the total solution times are often improved significantly.

We define $h(x) := \frac{f(x)}{g(x)}$. Thus,

$$\lambda^\star = \max_{x\in\mathbb{B}^n} h(x) = \max_{x\in\mathbb{B}^n} \frac{f(x)}{g(x)}. \tag{4.16}$$

Next, let $x^\star$ denote an optimal of (4.16), i.e., $x^\star \in \operatorname*{argmax}_{x\in\mathbb{B}^n} h(x)$. Then for any feasible solution $\bar{x}$ we define the relative and absolute optimality gaps as follows.

$$\text{Relative gap: } \ \mathtt{gap}_{rel} := \left|\frac{h(x^\star) - h(\bar{x})}{h(\bar{x})}\right|, \qquad \text{Absolute gap: } \ \mathtt{gap}_{abs} := |h(x^\star) - h(\bar{x})|. \tag{4.17}$$

If Algorithm 1 terminates before reaching the time limit, then it yields a feasible solution with either $\mathtt{gap}_{rel} \leqslant \epsilon_{rel}$ or $\mathtt{gap}_{abs} \leqslant \epsilon_{abs}$. If the time limit is reached after processing the i-th iteration of the algorithm, then

$$\mathtt{gap}_{rel} \leqslant |(\overline{\lambda}^i - \underline{\lambda}^i)/\underline{\lambda}^i|, \qquad \text{and} \qquad \mathtt{gap}_{abs} \leqslant |\overline{\lambda}^i - \underline{\lambda}^i|. \tag{4.18}$$

4.4.2 Newton-like method algorithm

The second approach that we employ to find the root of problem (4.15) is based on Newton-like method [17, 31, 60] described as follows. Suppose that at the beginning of iteration i a lower-bound t^i on $\lambda^\star$ is known, which can be obtained, e.g., by computing the fractional objective function at any feasible solution. If $v(t^i) = 0$, then $t^i = \lambda^\star$; otherwise, the algorithm updates $t^{i+1} = h(x^i)$, where x^i is an optimal solution of $v(t^i)$, and proceeds to the next iteration. The formal pseudo-code is given in Algorithm 2.

Note that at each iteration of Algorithm 2 we can stop the optimization of problem (4.15) in line 6 whenever a feasible solution with an objective function value greater than $\epsilon_{rel} \cdot |t^i|$

and ϵ_{abs} is found, which, based on the discussion in Section 4.4.1, can result in more iterations but a better performance for the algorithm.

Algorithm 2 Newton-like method algorithm

Input: ϵ_{rel}, relative gap parameter; ϵ_{abs}, absolute gap parameter;
Output: x; if $x_j = 1$, then feature j is selected
$i \leftarrow 0$
Compute t^i ▷ e.g., $t^i = h(\mathbf{1}')$
while time limit not exceeded **do**
  Solve problem (4.15) for t^i and obtain $v(t^i)$ and its optimal solution x^i
  if $v(t^i) > \epsilon_{rel} \cdot |t^i|$ **and** $v(t^i) > \epsilon_{abs}$ **then**
    $t^{i+1} \leftarrow h(x^i)$
  else
    return x^i ▷ Solution found within either relative or optimality gaps
  end if
  $i \leftarrow i + 1$
end while
return x^i ▷ Best solution found within the time limit

Recall the relative and optimality gaps defined in (4.17). Following the proofs of similar results in [79] and [37, Proposition 4], if the time limit is not reached, then Algorithm 2 terminates with a feasible solution with either $\texttt{gap}_{rel} \leqslant \epsilon_{rel}$ or $\texttt{gap}_{abs} \leqslant \epsilon_{abs}$. If the time limit is reached after the operation of the i-th iteration of Algorithm 2, then we compute approximations of relative and absolute gaps by

$$\texttt{gap}_{rel} \simeq \frac{v(t^i)}{|t^i| \cdot g(x^i)}, \qquad \text{and} \qquad \texttt{gap}_{abs} \simeq \frac{v(t^i)}{g(x^i)}. \tag{4.19}$$

4.5 Computational results

The aim of our computational study is to evaluate the performances of the MILP reformulations provided in Section 4.3 versus the parametric approaches of Section 4.4. In Section 4.5.1, we outline the real-life test instances and settings used for computational experiments. Then we present our results in Section 4.5.2.

4.5.1 Computational environment and test instances

In all of the computational test instances, we solve MILPs and BQPs (in each iteration of the parametric Algorithms 1 and 2) using CPLEX 12.7.1 [47]. We run experiments on a computer, where we allocate 4 threads (CPU 2.90GHz) and 16 GB of RAM for each individual experiment. We use a time limit of one hour (3600 seconds). To avoid running-out-of-memory difficulties we use the "node-file storage-feature" of CPLEX to store some parts of the branch-and-cut tree on a disk when the size of the tree exceeds the allocated memory.

Furthermore, for computing the mutual information and correlation between a feature and the target class or between two features, as well as computing the classification accuracy score we use *scikit-learn* package [72] and Python 3.7.3 [78].

Test instances. We consider various real-world instances obtained from *UCI machine learning repository* [5] and *ASU feature selection repository* [55] available at `https://archive.ics.uci.edu` and http://featureselection.asu.edu, respectively. Table 13 provides the list of instances as well as their sizes and their key characteristics.

Linearization bounds. In both MILP_1 and MILP_2, we let $y^\ell = 0$ and $y^u = 1$. Moreover, for MILP_2 reformulation of mRMR we let $\mathcal{M}_j^b = \sum_{k \in J} |\mathcal{I}(f_j, C) - \mathcal{I}(f_j, f_k)|$ and $\mathcal{M}_j^d = n$, for all $j \in J$. For MILP_2 reformulation of CFS we set $\mathcal{M}_j^b = \sum_{k \in J} \rho(f_j, C) \cdot \rho(f_k, C)$ and $\mathcal{M}_j^d = \sum_{k \in J, k \neq j} 2\rho(f_j, f_k)$, for all $j \in J$. Finally, we consider $\mathcal{M} = \sum_{j \in J} \sum_{k \in J} |\mathcal{I}(f_j, C) - \mathcal{I}(f_j, f_k)|$ in MILP_4.

Gaps. We consider $\epsilon_{rel} = 0.01$ and $\epsilon_{abs} = 0.001$ in both Algorithms 1 and 2. If the time limit is reached, then $\texttt{gap}_{rel}$ and $\texttt{gap}_{rel}$ are computed by using formulas given in (4.18) and (4.19) for Algorithms 1 and 2, respectively. Similarly, in solving of the MILPs we set 0.01 and 0.001 for the relative and absolute optimality gaps in the solver which are computed by $\texttt{gap}_{rel} = |\frac{UB - LB}{LB}|$ and $\texttt{gap}_{abs} = |UB - LB|$, where UB and LB are the upper- and the lower-bound on the optimal objective function value at the termination of the solver, respectively.

Table 13: The sizes of the considered instances including the number of features, n, and the number of samples, m. Additionally, we provide some characteristics of the data instances such as the type of features values and the type of target class variable; if $|\overline{C}| = 2$, then the target class is binary, otherwise it is multi-class.

Instance	n	m	Data type	Class type
banknote_authentication[1]	4	1,372	continuous	binary
Breast_cancer[1]	9	286	discrete	binary
Letter_Recognition[1]	16	20,000	discrete	multi
Zoo[1]	17	101	discrete	multi
Breast_Cancer_Wisconsin_(Diagnostic)[1]	31	569	continuous	binary
SPECTF_Heart_Data[1]	44	267	continuous	binary
Lung_Cancer[1]	56	32	discrete	binary
Sports_articles_for_objectivity_analysis[1]	59	1,000	discrete	binary
Connectionist[1]	60	208	continuous	binary
Optical_Recognition[1]	62	3,823	discrete	multi
Hill-Valley[1]	100	606	continuous	binary
Urban_Land_Cover[1]	147	168	continuous	multi
Epileptic_Seizure_Recognition[1]	178	11,500	discrete	multi
SCADI[1]	205	70	discrete	multi
Semeion_Handwritten_Digit[1]	256	1,593	discrete	multi
USPS[2]	256	9,298	continuous	multi
lung_discrete[2]	325	73	discrete	multi
Madelon[1,2]	500	2,000	continuous	binary
ISOLET[1,2]	617	7,797	continuous	multi
Parkinson's_Disease[1]	754	756	continuous	binary
CNAE-9[1]	856	1,080	discrete	multi
Yale_32x32[2]	1,024	165	continuous	multi
ORL_32x32[2]	1,024	400	continuous	multi
colon[2]	2000	62	discrete	binary
PCMAC[2]	3289	1943	discrete	binary

[1]UCI machine learning repository [5]. [2]ASU feature selection repository [55].

Classification accuracy score. Given a sample, the accuracy of a subset of features in predicting the true class of the sample can be evaluated by the classification accuracy. We use the well-known *Naive Bayes classifier* method (commonly used in the related literature, see, e.g., [67, 68, 73]), described below with the 5-fold cross validation to evaluate the accuracy of a subset of features.

Recall that set $\overline{C}$ denotes the set of possible values for the target class variable, i.e., $C \in \overline{C}$. Let S be a subset of features and A be a vector of size $|S|$, where A_j is the value of feature $f_j \in S$ in the sample. Then in order to evaluate the classification accuracy of S in classifying sample A, under the assumption that features are independent, *Naive Bayes classifier* uses the following equation to find the class of sample C_A.

$$C_A = \underset{c_k \in \overline{C}}{\operatorname{argmax}}\ \mathbb{P}(c_k) \prod_{A_j \in A} \mathbb{P}(A_j | c_k), \tag{4.20}$$

where probabilities $\mathbb{P}(c_k)$ and $\mathbb{P}(a_j|c_k)$ are computed based on the training data set. Equation (4.20) implies that the most probable class is assigned as the class of sample A.

4.5.2 Results and analysis

In this section, we evaluate the performances of the MILPs of Section 4.3 versus Algorithms 1 and 2 of Section 4.4. First, we discuss the results for the MILPs in solving the mRMR feature selection problem, see Table 14. We observe that for "small" instances ($n \leqslant 60$), MILP_4 has, in general, the best performance among the MILPs. In particular, for $44 \leqslant n \leqslant 60$, MILP_1, MILP_2, and MILP_3 do not find an optimal solution within the time limit, while MILP_4 solves the same instances to optimality in only a few seconds.

For larger instances ($n > 60$), all MILPs reach the time limit. In these larger instances, if MILP_1 finds a feasible solution, then it typically has better (absolute and relative) gaps than the other MILPs. Nevertheless, for $n \geqslant 500$, MILP_1 and MILP_3 are not able to find even a feasible solution, while MILP_2 and MILP_4 report rather poor results (gaps larger than 100); see Table 14.

Next, we compare the results for the best two MILPs (i.e., MILP_1 and MILP_4 based on the above discussion) against Algorithms 1 and 2 in solving mRMR; see Table 15. The most important observation is that the parametric algorithms perform better than the MILPs for $n > 60$. These algorithms either find solutions within the optimality gaps or report much better gaps than MILPs if the time limit is reached. Additionally, their performances are competitive with those of MILPs for $n \leqslant 60$.

In Table 16, we report the results for the CFS feature selection problem. Similar to the aforementioned results for mRMR, we observe that for CFS the parametric algorithms outperform both MILP_1 and MILP_2. Additionally, we note that solving CFS is easier than solving mRMR with respect to the running time and gaps. For example, Algorithm 1 can find an optimal solution of CFS for the largest considered instance, i.e., "PCMAC", within the optimality absolute gap in 955 seconds; see Table 16. On the other hand, none of the solution methods are able to find an optimal solution of mRMR for this instance in the time limit; see Tables 14 and 15.

By comparing the performances of the parametric algorithms (Tables 15 and 16), we note that Algorithms 1 and 2 have similar running times for the instances that they solve to optimality. For the instances where an optimal solution is not found within the time limit, Algorithm 1 can be a better choice as for these instances gap_{rel} and gap_{abs} reported by Algorithm 2 are approximations of the relative and absolute gaps, respectively; thus, the reported gaps by Algorithm 2 are not properly comparable to the corresponding gaps' values for the other solution methods.

Additionally, recall that for the binary-search algorithm the objective function value of the full feature set is considered as an initial lower-bound on the optimal objective function value. Hence, for some instances such as "ORL_32x32" in Tables 15 and 16, Algorithm 1 takes most of the time to improve the upper bound. Therefore, the best reported solution at the termination of the algorithm is the full feature set. In case of the Newton method, the full feature set is considered as the initial solution. Observe that Algorithm 2 cannot process more than one iteration either in Table 15 or Table 16 within the time limit for some instances such as "ORL_32x32". Therefore, for these instances the best reported solution at the termination is the full set. However, note that both algorithms report significantly better gaps than the best MILPs, which are promising for finding optimal or near optimal solutions in a larger time limit.

It is worth mentioning that the choice of an appropriate feature selection measure may depend on the instance data set and its application setting (see, e.g., [23, 50] for rather comprehensive discussions). In particular, due to the different structures and also coefficients values of the problems, the sizes of the selected subsets of features by CFS are typically smaller than those selected by mRMR. For example, compare the columns of $|S|$ in Table 16 versus those of Table 15 for “small” instances ($n \leqslant 60$).

Finally, the classification accuracy score of the (optimal) output result of each feature selection measure depends on the test instance. For example, the optimal subset of features selected by mRMR for test instance “Zoo” has a better score than the optimal subset selected by CFS (0.84 vs. 0.41 based on the results for Algorithm 1); see also test instance “Letter-Recognition” for the opposite case.

4.6 Concluding remarks

Feature selection is an essential preprocessing step in many data mining and machine learning tasks and involves finding a small subset of the most characterizing features from the data set. In this chapter, we focus on feature selection problems based on mRMR and CFS measures that are typically tackled either by heuristic methods or their reformulations as MILPs. However, heuristics do not guarantee the optimality of the output subset and MILPs given in the literature have rather poor performances even for small- and medium-sized instances.

To address the aforementioned shortcomings, we consider approaches that ensure globally optimal solutions. To this end, we propose an MILP reformulation approach for the mRMR feature selection problem which outperforms existing MILPs in the literature. Additionally, we apply parametric approaches to solve both the mRMR and CFS feature selection problems. Our computational experiments with real-world data sets show that the proposed approaches lead to encouraging improvements on the performance of solution methods for the mRMR and CFS problems.

Table 14: Comparison of results for MILP_1 to MILP_4 in solving the mRMR feature selection problem (4.3). For each test instance, the size of the full set of features (n) and its accuracy score (`score`) is reported, where the latter is computed as discussed in Section 4.5.1. Moreover, for each test instance and solution method we present absolute ($\texttt{gap}_{abs}$) and relative ($\texttt{gap}_{rel}$) gaps and `score`, as well as time (`time`, in seconds), the number of selected features ($|S|$, for the best found integer solution) at the termination of solver (for MILPs) and the algorithms.

Instance	Full set		MILP_1 [24]					MILP_2 [67]					MILP_3					MILP_4				
	n	score	$\texttt{gap}_{abs}$	$\texttt{gap}_{rel}$	time	$\|S\|$	score	$\texttt{gap}_{abs}$	$\texttt{gap}_{rel}$	time	$\|S\|$	score	$\texttt{gap}_{abs}$	$\texttt{gap}_{rel}$	time	$\|S\|$	score	$\texttt{gap}_{abs}$	$\texttt{gap}_{rel}$	time	$\|S\|$	score
banknote_authentication	4	0.84	0.000	0.00	0.5	4	0.84	0.000	0.00	0.4	4	0.84	0.000	0.00	0.4	4	0.84	0.000	0.00	0.5	4	0.84
Breast_cancer	9	0.75	0.000	0.00	0.5	7	0.74	0.000	0.00	0.4	7	0.76	0.000	0.00	0.8	7	0.74	0.000	0.00	0.5	7	0.76
Letter_Recognition	16	0.34	0.002	0.01	3.5	14	0.37	0.002	0.01	1.2	14	0.37	0.002	0.01	15.5	14	0.37	0.002	0.01	0.7	14	0.37
Zoo	17	0.79	0.003	0.01	1.2	8	0.81	0.003	0.01	3.0	8	0.81	0.003	0.01	105.2	8	0.85	0.003	0.01	1.1	8	0.80
Breast_Cancer_Wisconsin_(Diagnostic)	31	0.62	0.001	0.03	61.3	22	0.93	0.001	0.03	30.1	22	0.93	0.001	0.03	1447.6	22	0.93	0.001	0.03	3.7	22	0.94
SPECTF_Heart_Data	44	0.72	0.063	0.90	T	5	0.72	1.202	17.09	T	5	0.70	4.994	73.33	T	5	0.74	0.001	0.01	50.9	5	0.72
Lung_Cancer	56	0.79	0.028	2.90	T	9	0.72	1.413	+	T	9	0.72	1.809	+	T	14	0.88	0.001	0.10	9.7	9	0.71
Sports_articles_for_objectivity_analysis	59	0.82	0.004	+	T	1	0.64	5.559	+	T	2	0.64	1.347	+	T	1	0.64	0.000	0.00	4.1	1	0.64
Connectionist	60	0.68	0.001	0.55	153.3	39	0.66	0.001	0.55	1649.0	39	0.66	0.434	+	T	39	0.68	0.001	0.54	21.8	39	0.65
Optical_Recognition	62	0.92	0.077	0.41	T	32	0.90	5.071	26.93	T	32	0.90	85.229	+	T	57	0.91	0.295	1.56	T	32	0.90
Hill-Valley	100	0.52	0.037	0.06	T	10	0.52	0.602	0.96	T	9	0.52	0.645	1.00	T	4	0.52	0.632	1.00	T	7	0.52
Urban_Land_Cover	147	0.77	0.275	1.11	T	53	0.79	28.586	+	T	47	0.84	+	+	T	147	0.76	8.516	33.75	T	45	0.82
Epileptic_Seizure_Recognition	178	0.44	0.022	0.42	T	115	0.43	59.495	+	T	90	0.44	+	+	T	178	0.44	2.636	48.96	T	152	0.44
SCADI	205	0.81	0.104	0.42	T	15	0.84	46.687	+	T	15	0.84	+	+	T	190	0.80	5.321	23.30	T	10	0.76
Semeion_Handwritten_Digit	256	0.84	0.143	1.37	T	29	0.71	93.234	+	T	26	0.64	+	+	T	1	0.07	+	+	T	113	0.84
USPS	256	0.78	0.148	0.76	T	27	0.48	47.627	+	T	33	0.49	+	+	T	1	0.16	+	+	T	50	0.60
lung_discrete	325	0.78	0.191	0.79	T	22	0.59	95.927	+	T	33	0.72	+	+	T	1	0.22	+	+	T	292	0.82
Madelon	500	0.58	-	-	T	-	-	0.064	+	T	429	0.55	-	-	T	-	-	1.115	57.48	T	495	0.58
ISOLET	617	0.84	-	-	T	-	-	+	+	T	43	0.65	-	-	T	-	-	+	+	T	617	0.84
Parkinson's_Disease	754	0.74	-	-	T	-	-	+	+	T	166	0.78	-	-	T	-	-	+	+	T	754	0.73
CNAE-9	856	1.00	-	-	T	-	-	31.078	+	T	655	1.00	-	-	T	-	-	0.122	+	T	856	1.00
Yale_32x32	1024	0.55	-	-	T	-	-	+	+	T	94	0.57	-	-	T	-	-	+	+	T	1024	0.55
ORL_32x32	1024	0.83	-	-	T	-	-	+	+	T	884	0.83	-	-	T	-	-	+	+	T	1024	0.84
colon	2000	0.67	-	-	T	-	-	+	+	T	260	0.76	-	-	T	-	-	+	+	T	2000	0.69
PCMAC	3289	0.92	-	-	T	-	-	+	+	T	3289	0.92	-	-	T	-	-	+	+	T	3289	0.92

"-": No feasible solution is found within the time limit. "+": `gap` is larger than 100. "T": Time limit (3600 sec.) is reached.

Table 15: Comparison of results for the best MILPs (MILP_1 and MILP_4) versus Algorithms 1 and 2 in solving the mRMR feature selection problem (4.3). For each test instance, the size of the full set of features (n) and its accuracy score (`score`) is reported, where the latter is computed as discussed in Section 4.5.1. Moreover, for each test instance and solution method we present absolute ($\texttt{gap}_{abs}$) and relative ($\texttt{gap}_{rel}$) gaps and `score`, as well as time (`time`, in seconds), the number of selected features ($|S|$, for the best found integer solution) at the termination of solver (for MILPs) and the algorithms, and the number of iterations of the algorithms (`#`).

Instance	Full set		MILP_1 [24]					MILP_4					Algorithm 1 (Binary search)						Algorithm 2 (Newton method)					
	n	score	$\texttt{gap}_{abs}$	$\texttt{gap}_{rel}$	time	$\|S\|$	score	$\texttt{gap}_{abs}$	$\texttt{gap}_{rel}$	time	$\|S\|$	score	$\texttt{gap}_{abs}$	$\texttt{gap}_{rel}$	time	$\|S\|$	score	#	$\texttt{gap}_{abs}$	$\texttt{gap}_{rel}$	time	$\|S\|$	score	#
banknote_authentication	4	0.84	0.000	0.00	0.5	4	0.84	0.000	0.00	0.5	4	0.84	0.004	0.01	1.6	4	0.84	10	0.001	0.01	0.3	4	0.84	1
Breast_cancer	9	0.75	0.000	0.00	0.5	7	0.74	0.000	0.00	0.5	7	0.76	0.001	0.01	1.4	7	0.75	12	0.001	0.01	0.9	7	0.75	2
Letter_Recognition	16	0.34	0.002	0.01	3.5	14	0.37	0.002	0.01	0.7	14	0.37	0.002	0.01	2.4	14	0.37	11	0.001	0.01	1.9	14	0.37	2
Zoo	17	0.79	0.003	0.01	1.2	8	0.81	0.003	0.01	1.1	8	0.80	0.003	0.01	2.7	8	0.84	10	0.001	0.01	2.1	8	0.82	4
Breast_Cancer_Wisconsin_(Diagnostic)	31	0.62	0.001	0.03	61.3	22	0.93	0.001	0.03	3.7	22	0.94	0.001	0.02	23.2	25	0.93	12	0.001	0.01	9.6	22	0.93	4
SPECTF_Heart_Data	44	0.72	0.063	0.90	T	5	0.72	0.001	0.01	50.9	5	0.72	0.001	0.01	2.5	5	0.72	12	0.001	0.01	4.2	5	0.72	5
Lung_Cancer	56	0.79	0.028	2.90	T	9	0.72	0.001	0.10	9.7	9	0.71	0.001	0.08	3.4	9	0.71	12	0.001	0.01	6.1	9	0.71	7
Sports_articles_for_objectivity_analysis	59	0.82	0.004	+	T	1	0.64	0.000	0.00	4.1	1	0.64	0.001	3.13	0.8	2	0.64	12	0.001	0.01	2.5	2	0.64	10
Connectionist	60	0.68	0.001	0.55	153.3	39	0.66	0.001	0.54	21.8	39	0.65	0.001	0.47	16.0	45	0.67	12	0.001	0.01	24.7	38	0.66	3
Optical_Recognition	62	0.92	0.077	0.41	T	32	0.90	0.295	1.56	T	32	0.90	0.001	0.01	9.3	32	0.90	11	0.001	0.01	7.0	32	0.90	3
Hill-Valley	100	0.52	0.037	0.06	T	10	0.52	0.632	1.00	T	7	0.52	0.004	0.01	7.8	10	0.52	10	0.001	0.01	17.4	10	0.52	7
Urban_Land_Cover	147	0.77	0.275	1.11	T	53	0.79	8.516	33.75	T	45	0.82	0.087	0.42	T	147	0.77	5	0.027	0.11	T	90	0.80	5
Epileptic_Seizure_Recognition	178	0.44	0.022	0.42	T	115	0.43	2.636	48.96	T	152	0.44	0.003	0.05	T	78	0.43	10	0.003	0.06	T	87	0.44	17
SCADI	205	0.81	0.104	0.42	T	15	0.84	5.321	23.30	T	10	0.76	0.001	0.01	1128.9	15	0.83	11	0.035	0.15	T	15	0.84	13
Semeion_Handwritten_Digit	256	0.84	0.143	1.37	T	29	0.71	+	+	T	113	0.84	0.001	0.01	162.0	27	0.63	11	0.001	0.01	511.5	26	0.63	17
USPS	256	0.78	0.148	0.76	T	27	0.48	+	+	T	50	0.60	0.023	0.12	T	36	0.48	7	0.019	0.11	T	55	0.55	8
lung_discrete	325	0.78	0.191	0.79	T	22	0.59	+	+	T	292	0.82	0.003	0.01	T	29	0.70	10	0.000	0.00	T	33	0.78	14
Madelon	500	0.58	-	-	T	-	-	1.115	57.48	T	495	0.58	0.001	0.04	1993.4	500	0.58	12	0.000	0.01	T	500	0.59	1
ISOLET	617	0.84	-	-	T	-	-	+	+	T	617	0.84	0.047	5.18	T	617	0.84	6	0.009	0.49	T	617	0.84	1
Parkinson's_Disease	754	0.74	-	-	T	-	-	+	+	T	754	0.73	0.024	1.30	T	754	0.75	7	0.001	0.78	T	656	0.75	2
CNAE-9	856	1.00	-	-	T	-	-	0.122	+	T	856	1.00	0.012	1.17	T	856	1.00	8	0.000	2.13	T	856	1.00	1
Yale_32x32	1024	0.55	-	-	T	-	-	+	+	T	1024	0.55	0.091	1.17	T	1024	0.55	5	0.034	0.39	T	1024	0.55	1
ORL_32x32	1024	0.83	-	-	T	-	-	+	+	T	1024	0.84	0.092	1.28	T	1024	0.83	5	0.032	0.39	T	1024	0.83	1
colon	2000	0.67	-	-	T	-	-	+	+	T	2000	0.67	0.097	0.86	T	2000	0.67	5	0.037	0.36	T	2000	0.67	1
PCMAC	3289	0.92	-	-	T	-	-	+	+	T	3289	0.92	0.047	5.44	T	3289	0.92	6	0.001	0.54	T	3289	0.92	1
Average	475.6	0.7	0.065	0.61	2746.5	24.2	0.69	1.165	11.12	7430.2	434.76	0.73	0.022	0.83	2174.5	422.3	0.72	9.3	0.007	0.22	7470.1	422.2	0.72	3.1

"-": No feasible solution is found within the time limit. "+": gap is larger than 100. "T": Time limit (3600 sec.) is reached.

Table 16: Comparison of results for MILP_1 and MILP_2 versus Algorithms 1 and 2 in solving the CFS feature selection problem (4.5). For each test instance, the size of the full set of features (n) and its accuracy score (`score`) is reported, where the latter is computed as discussed in Section 4.5.1. Moreover, for each test instance and solution method we present absolute ($\texttt{gap}_{abs}$) and relative ($\texttt{gap}_{rel}$) gaps and `score`, as well as time (`time`, in seconds), the number of selected features ($|S|$, for the best found integer solution) at the termination of solver (for MILPs) and the algorithms, and the number of iterations of the algorithms (`#`).

Instance	Full set		MILP_1 [24]					MILP_2 [67]					Algorithm 1 (Binary search)						Algorithm 2 (Newton method)					
	n	score	gap_{abs}	gap_{rel}	time	$\|S\|$	score	gap_{abs}	gap_{rel}	time	$\|S\|$	score	gap_{abs}	gap_{rel}	time	$\|S\|$	score	#	gap_{abs}	gap_{rel}	time	$\|S\|$	score	#
banknote_authentication	4	0.84	0.000	0.00	0.3	1	0.84	0.000	0.00	0.3	1	0.84	0.001	0.01	1.1	1	0.84	12	0.001	0.01	0.4	1	0.84	3
Breast_cancer	9	0.75	0.001	0.11	0.3	1	0.76	0.000	0.00	0.3	1	0.76	0.001	0.08	0.8	1	0.76	12	0.001	0.01	0.4	2	0.76	3
Letter_Recognition	16	0.34	0.000	0.00	0.4	1	0.40	0.001	0.01	0.4	1	0.40	0.001	0.01	0.9	1	0.40	12	0.001	0.01	1.5	1	0.40	5
Zoo	17	0.79	0.025	0.01	0.4	1	0.41	0.000	0.00	0.4	1	0.41	0.018	0.01	0.4	1	0.41	7	0.001	0.01	0.9	1	0.40	3
Breast_Cancer_Wisconsin_(Diagnostic)	31	0.62	0.000	0.00	0.6	1	0.92	0.000	0.00	0.3	1	0.92	0.001	0.01	0.9	1	0.92	11	0.001	0.01	1.1	1	0.92	4
SPECTF_Heart_Data	44	0.72	0.042	0.06	T	1	0.73	0.007	0.01	25.1	1	0.72	0.005	0.01	0.9	1	0.72	9	0.001	0.01	2.8	1	0.72	7
Lung_Cancer	56	0.79	0.001	0.01	36.0	1	0.72	0.001	0.01	1634.8	1	0.73	0.001	0.01	0.7	1	0.72	12	0.001	0.01	1.6	1	0.71	7
Sports_articles_for_objectivity_analysis	59	0.82	0.002	0.01	67.7	1	0.64	0.003	0.01	1.6	1	0.64	0.003	0.01	0.7	1	0.64	10	0.001	0.01	1.9	1	0.64	6
Connectionist	60	0.68	0.001	0.26	3.2	1	0.57	0.001	0.26	0.9	1	0.57	0.001	0.20	0.8	1	0.58	12	0.001	0.01	0.5	4	0.68	3
Optical_Recognition	62	0.92	0.321	0.50	T	13	0.86	1.170	1.81	T	8	0.80	0.005	0.01	7.4	13	0.86	9	0.001	0.01	14.2	13	0.86	9
Hill-Valley	100	0.52	0.001	2.16	701.1	1	0.51	0.001	3.15	T	1	0.49	0.001	1.98	0.9	1	0.48	12	0.001	0.01	1.6	5	0.52	7
Urban_Land_Cover	147	0.77	2.327	2.21	T	1	0.44	5.749	5.45	T	1	0.45	0.010	0.01	3.9	1	0.42	8	0.001	0.01	36.9	1	0.40	8
Epileptic_Seizure_Recognition	178	0.44	1.104	9.97	T	109	0.44	1.911	16.24	T	69	0.44	0.003	0.02	T	64	0.43	10	0.004	0.03	T	73	0.44	19
SCADI	205	0.81	11.271	18.41	T	7	0.77	2.614	4.25	T	6	0.71	0.020	0.03	T	12	0.83	7	0.179	0.30	T	19	0.84	9
Semeion_Handwritten_Digit	256	0.84	58.200	+	T	256	0.84	3.778	7.06	T	106	0.84	0.323	0.77	T	256	0.84	3	0.112	0.22	T	154	0.85	7
USPS	256	0.78	21.983	60.55	T	256	0.79	5.706	13.24	T	36	0.55	0.165	0.47	T	256	0.78	4	0.204	0.49	T	39	0.52	10
lung_discrete	325	0.78	32.728	54.96	T	325	0.78	29.708	30.35	T	29	0.72	0.604	1.03	T	325	0.81	2	4.236	5.03	T	25	0.61	10
Madelon	500	0.58	0.006	4.11	T	2	0.59	0.015	9.83	T	2	0.59	0.001	0.60	15.7	3	0.61	12	0.001	0.01	17.1	3	0.61	7
ISOLET	617	0.84	+	+	T	617	0.84	1.166	11.12	T	73	0.77	0.046	0.82	T	617	0.83	6	0.129	1.42	T	77	0.76	9
Parkinson's_Disease	754	0.74	0.284	85.52	T	754	0.74	0.336	25.27	T	12	0.82	0.003	0.27	T	15	0.80	10	0.032	2.43	T	12	0.83	13
CNAE-9	856	1.00	0.004	52.33	T	3	1.00	0.001	4.77	7.6	13	1.00	0.001	2.45	13.6	22	1.00	12	0.001	0.01	4.0	23	1.00	4
Yale_32x32	1024	0.55	-	-	T	-	-	+	+	T	2	0.32	0.005	0.01	506.9	3	0.39	9	0.033	0.12	T	1024	0.53	1
ORL_32x32	1024	0.83	-	-	T	-	-	+	+	T	7	0.68	0.176	0.93	T	1024	0.84	4	0.032	0.16	T	1024	0.83	1
colon	2000	0.67	-	-	T	-	-	+	+	T	1	0.64	0.023	0.27	T	9	0.79	7	0.009	0.45	T	571	0.76	3
PCMAC	3289	0.92	-	-	T	-	-	20.059	+	T	3	0.79	0.001	0.03	955.0	3	0.67	12	0.003	0.57	T	1997	0.93	2

"-": No feasible solution is found within the time limit. "+": `gap` is larger than 100. "T": Time limit (3600 sec.) is reached.

5.0 Conclusions

This dissertation considers generally structured single- and multiple-ratio fractional binary programs, FPs, which have traditionally been tackled by reformulating the problems as MILPs with a large number of variables and constraints. However, new techniques have recently been proposed to improve upon the classical MILP formulations. Chapter 2 focuses on two such recent enhancements including binary-expansion technique as well as conic and submodular strengthenings. Naturally, there is a trade-off between using these two techniques. The former reduces the size of a problem at the cost of a weaker relaxation, and the latter improves the relaxation quality at the expense of a larger problem. However, the synthesis of these ideas leads to new moderately sized formulations which yet retain the relaxation strength of formulations of much larger sizes. As a consequence, in our computations using benchmark instances, we observe that the new formulations perform typically as well as the best existing methods for small problems, and often significantly outperform existing methods for larger instances.

Chapter 3 addresses RFPs, defined as the robust counterparts of the fractional binary programs, under various disjoint and joint uncertainty sets. We demonstrate that single-ratio RFP, contrary to its deterministic counterpart, is NP-hard for a general polyhedral uncertainty set. However, if the uncertainties are in the form of the dis/joint budgeted uncertainty sets, then we develop polynomial-time solution methods for single-ratio RFP provided that the nominal problem is polynomial-time solvable.

In case of multiple-ratio RFPs, we exploit the structure of the budgeted dis/joint uncertainty sets in order to propose various MILPs to solve them. Particularly, based on our extensive computational experiments we observe that RFPs are more challenging to solve under the joint uncertainty sets than under the disjoint one, as the former cannot take advantage of the binary-expansion technique.

We also explore the value of the robust optimal solution for instances with both the real and synthetic data and find that ignoring the data uncertainty can lead to poor decisions. These results coupled with the insights on the selection of budget(s) of uncertainties can provide guidance to identify suitable solution methods and level of uncertainty in practice. It is worth mentioning that conic quadratic programming approaches that lead to strong convex relaxations for the deterministic case can be pursued as a promising future research direction to improve the performance of solution approaches for RFPs.

Chapter 4 studies fractional 0-1 programs in the application setting of correlation-based and mutual-information-based feature selection optimization problems. We propose a new MILP reformulation approach for the latter problem. Moreover, we apply parametric approaches to tackle fractional models of these problems and report encouraging results. Finally, for the future research it is of interest to model other suitable feature selection measures as fractional 0-1 programs and extend the advanced approaches of Chapters 2 and 3 in these contexts.

Appendix

Supplement for Chapter 2

A.1 Assumption justifications

We make the following assumptions in Chapter 2.

Assumption 3. *All data are integers, i.e.,* $a_{ij}, b_{ij} \in \mathbb{Z}$ *for all* $i \in I, j \in J \cup \{0\}$.

Assumption 4. *All data are non-negative, i.e.,* $a_{ij}, b_{ij} \geqslant 0$ *for all* $i \in I$ *and* $j \in J \cup \{0\}$.

Assumption 3 is without loss of generality, as otherwise rational coefficients can be scaled. Assumption 4 is naturally satisfied in most application settings, as the data typically represents probabilities, prices, weights, utilities etc. – see, e.g., [17] and the applications described therein.

Nonetheless, Assumption 4 is without loss of generality provided that (the weaker and commonly made assumption in the FP literature, see, e.g., [15, 16, 43]) $b_{i0} + \sum_{j \in J} b_{ij} x_j > 0$ for all $x \in \mathbb{B}^n$ holds. In each ratio $i \in I$, for every $j \in J$ such that $b_{ij} < 0$ and every j such that $b_{ij} = 0$ and $a_{ij} < 0$, replace x_j with $\bar{x}_j = 1 - x_j$, resulting in a problem satisfying $b_{ij} \geqslant 0$ (possibly with at most n additional variables and constraints). Then observe that for any $k_i \in \mathbb{R}$

$$\frac{a_{i0} + \sum_{j \in J} a_{ij} x_j}{b_{i0} + \sum_{j \in J} b_{ij} x_j} = \frac{(a_{i0} + k_i b_{i0}) + \sum_{j \in J} (a_{ij} + k_i b_{ij}) x_j}{b_{i0} + \sum_{j \in J} b_{ij} x_j} - k_i. \tag{.1}$$

Thus, by letting k_i sufficiently large for each $i \in I$, we find a problem where all coefficients are non-negative.

Finally, note that if a fractional program is in maximization form and satisfies $b_{i0} + \sum_{j \in J} b_{ij} x_j > 0$ for all $x \in \mathbb{B}^n$, then it can be transformed into an equivalent problem in minimization form (by negating all coefficients a_{i0} and a_{ij}), and then applying the process above to obtain a problem satisfying Assumption 4.

A.2 Additional computational results

In this appendix, we compare the performance of the formulations presented in Chapter 2 (not restricted to those discussed in Section 2.4.3 and presented in Tables 3 and 4 as well as their extended versions, i.e., Tables 17 and 18) to evaluate the individual and combined effects of the enhancements. In order to have a better comparison of the results, we repeat the results for some of the formulations in different subsections.

In particular, first, in Section A.2.1, we compare the basic MILP and basic MICQP formulations without using additional enhancements. Then in Section A.2.2, we focus on the effect of the binary-expansion technique on the basic formulations. Next, in Section A.2.3, we focus on the impact of polymatroid cuts. In Section A.2.4, we test the formulations that benefit from the integration of the binary-expansion technique with the polymatroid cuts. Recall that, in the following tables, the "†" symbol is used if CPLEX is unable to fully process the root node of the branch-and-bound tree within the time limit for a given formulation.

A.2.1 Linear vs. conic formulations

Here, we evaluate the basic MILP (LF, LEF) and the basic MICQP (CF, CEF) reformulations, see Tables 19 and 20. Observe that, in most cases, LEF, CF, and CEF are stronger than LF, i.e., they have better `Rlx-gap`. Additionally, as expected, the extended formulations LEF and CEF are stronger than compact formulations, i.e., LF and CF, respectively. The extended formulations also shows better running time and `End-gap` than the corresponding compact formulations. In general, CEF performs better than LEF for low values of the parameter κ, while LEF is comparatively better for high values of κ. Moreover, none of the formulations except CF (with a very poor performance) are able to scale to $n = 1000$ for all instances. These results justify the development of enhanced formulations for the medium and large size instances.

A.2.2 Binary-expansion

Here, we explore the individual impact of binary-expansion technique on the performance and size of the basic formulations. Specifically, we compare LF and CEF versus their binarized versions, i.e., LF_{log} and CEF_{log}, respectively.

In Tables 21 and 22, we observe that LF has a poor performance even for $n = 100$. In contrast, its binarization leads to significant improvements in the results due to the reduction in the size of the formulation. These results are consistent with the previous results in the literature that LF_{log} has a superior performance over LF, LEF, and LEF_{log} – see [16, 61] and also the results for LEF in Tables 19 and 20.

On the other hand, for $n \leqslant 500$ formulation CEF outperforms CEF_{log} with respect to either time or the considered gaps; e.g., for $n = 500$ and $\kappa = 10\% \cdot n$ in Table 21, CEF reports the 0.2% average `End-gap`, compared to 5.1% for CEF_{log}. Nonetheless, CEF_{log} is able to scale to problems with $n \geqslant 1000$ while formulation CEF is not. Additionally, for the instances with $n \geqslant 2000$ we observe that in most cases CEF_{log} outperforms (the superior MILP formulation) LF_{log}, as well.

Tables 23 and 24 show the impact of binarization in the reduction of the number of continuous variables and linear as well as rotated cone constraints for the assortment and the uniformly generated data sets, respectively. It can be seen that the binary-expansion technique substantially reduces the number of (continuous) variables and constraints with a slight increase in the number of binary variables; the percent of these reductions gets larger as n grows. For example, in Table 23 for $n = 1000$, LF_{log} and CEF_{log} have at least 97,900 and 391,500 fewer continuous variables and linear constraints, respectively, than LF and CEF with the cost of at most 2,100 more binary variables. The binary-expansion technique also leads to a reduction of 97,900 rotated cone constraints for CEF.

A.2.3 Polymatroid cuts

Next, we explore the individual impact of polymatroid cuts on the basic formulations, namely, LF, LEF, CF, and CEF. Notably, for $n \leqslant 500$ in Tables 25 and 26, we observe that polymatroid cuts have a significant improvement effect on the performance (running time and

`End-gap`) of compact formulations LF and CF. However, the cuts are not that effective for LEF and CEF, as these extended formulations are much stronger and the cuts provide only a marginal improvement in the relaxation quality while increasing the sizes of the formulations.

Additionally, for $n \geqslant 1000$ polymatroid cuts are not beneficial and employing them makes the results worse, see, e.g., in Table 25 and $n = 1000$ that `End-gap` of LEF from 13.9% increases to 81% after employing the cuts. The reason is that CPLEX consumes the allocated time only to manage the cuts and process the root node.

A.2.4 Integration of binary-expansion and polymatroid cuts

Here, we explore the effect of simultaneous usage of both techniques, i.e., the impact of the incorporation of polymatroid cuts with binary expansion on LF and CEF. Tables 27 and 28 present the results and we make the following observations. Formulation $\mathrm{LF}^{\mathrm{P}}_{\log}$ either outperforms LF, LF^{P}, and $\mathrm{LF}_{\log}$ or (in a few cases) has a competitive performance with LF^{P}. On the other hand, for the small- and medium- size instances CEF and $\mathrm{CEF}^{\mathrm{P}}$ are competitive and they have better performances than $\mathrm{CEF}_{\log}$ and $\mathrm{CEF}^{\mathrm{P}}_{\log}$. However, for large instances $\mathrm{CEF}^{\mathrm{P}}_{\log}$ outperforms CEF, $\mathrm{CEF}_{\log}$ and $\mathrm{CEF}^{\mathrm{P}}$. These observations imply that - specially in large instances - the integration of binarization and polymatroid cuts in both MILPs and MICQPs leads to superior formulations. Specifically, $\mathrm{LF}^{\mathrm{P}}_{\log}$ and $\mathrm{CEF}^{\mathrm{P}}_{\log}$ perform better than the corresponding basic formulations and the enhanced ones that only use one of the improving techniques.

Additionally, it appears that for instances up to 500 variables, in general, CEF and $\mathrm{CEF}^{\mathrm{P}}$ are the most efficient formulations. For instances with $n \geqslant 1000$, $\mathrm{CEF}^{\mathrm{P}}_{\log}$ and $\mathrm{LF}^{\mathrm{P}}_{\log}$ outperform the others. Finally, we observe that, in general, $\mathrm{CEF}^{\mathrm{P}}_{\log}$ has a better performance in the constrained instances, while $\mathrm{LF}^{\mathrm{P}}_{\log}$ is superior in the unconstrained instances.

Table 17: Computational results to evaluate the best existing methods in the literature against the standout formulations for the assortment data set [85]. For each combination of n, m, κ and each formulation, we present averages over five instances for: time (`Time`) in seconds, number of nodes (`Nodes`) processed, end gap (`End-gap`), continuous relaxation gap (`Rlx-gap`), root node gap (`Ron-gap`), best-bound gap (`Bbn-gap`), and optimality gap (`Opt-gap`). For each choice of n, m, and κ, among the solution methods, the best average time and the best average `End-gap` (if `Time`$\geqslant$ 3600 sec.) are in **bold**.

	κ	$10\% \cdot n$							$20\% \cdot n$							Unconstrained						
n, m	Ref.	`Time`	`Nodes`	`End-gap`	`Rlx-gap`	`Ron-gap`	`Bbn-gap`	`Opt-gap`	`Time`	`Nodes`	`End-gap`	`Rlx-gap`	`Ron-gap`	`Bbn-gap`	`Opt-gap`	`Time`	`Nodes`	`End-gap`	`Rlx-gap`	`Ron-gap`	`Bbn-gap`	`Opt-gap`
25,2*	LF_{log}	**0**	5	0.0%	15.3%	1.1%	0.0%	0.0%	**0**	27	0.0%	28.1%	4.3%	0.0%	0.0%	1	386	0.0%	56.1%	18.1%	0.0%	0.0%
	LEF	**0**	2	0.0%	0.7%	0.7%	0.0%	0.0%	**0**	9	0.0%	0.9%	0.8%	0.0%	0.0%	**0**	0	0.0%	0.0%	0.0%	0.0%	0.0%
	CF^P	**0**	0	0.0%	1.5%	0.8%	0.0%	0.0%	**0**	0	0.0%	2.8%	1.9%	0.0%	0.0%	1	0	0.0%	8.2%	1.2%	0.0%	0.0%
	CEF	**0**	0	0.0%	0.0%	0.0%	0.0%	0.0%	**0**	1	0.0%	0.1%	0.1%	0.0%	0.0%	**0**	0	0.0%	0.0%	0.0%	0.0%	0.0%
	LF^P_{log}	**0**	0	0.0%	0.5%	0.3%	0.0%	0.0%	**0**	1	0.0%	0.9%	0.7%	0.0%	0.0%	**0**	20	0.0%	1.8%	0.1%	0.0%	0.0%
	CEF^P_{log}	1	1	0.0%	0.1%	0.1%	0.0%	0.0%	**0**	0	0.0%	0.6%	0.1%	0.0%	0.0%	2	379	0.0%	1.8%	1.5%	0.0%	0.0%
50,5*	LF_{log}	1	233	0.0%	30.0%	5.7%	0.0%	0.0%	2	2,109	0.0%	44.4%	14.5%	0.0%	0.0%	18	35,496	0.0%	65.6%	28.2%	0.0%	0.0%
	LEF	**0**	123	0.0%	3.0%	3.0%	0.0%	0.0%	**1**	1,111	0.0%	3.2%	3.2%	0.0%	0.0%	**0**	2	0.0%	0.0%	0.0%	0.0%	0.0%
	CF^P	1	0	0.0%	2.3%	0.1%	0.0%	0.0%	2	0	0.0%	5.5%	0.0%	0.0%	0.0%	4	4	0.0%	13.4%	0.9%	0.0%	0.0%
	CEF	1	1	0.0%	0.0%	0.0%	0.0%	0.0%	**1**	7	0.0%	0.2%	0.2%	0.0%	0.0%	1	2	0.0%	0.0%	0.0%	0.0%	0.0%
	LF^P_{log}	**0**	0	0.0%	0.9%	0.3%	0.0%	0.0%	**1**	19	0.0%	2.4%	0.1%	0.0%	0.0%	6	6,721	0.0%	5.9%	0.3%	0.0%	0.0%
	CEF^P_{log}	**0**	4	0.0%	0.5%	0.3%	0.0%	0.0%	2	250	0.0%	1.9%	0.9%	0.0%	0.0%	21	11,132	0.0%	5.9%	5.2%	0.0%	0.0%
100,10*	LF_{log}	979	364,141	0.0%	42.7%	14.5%	0.0%	0.0%	3155	1,732,777	0.4%	55.3%	23.7%	0.4%	0.0%	3600	1,543,428	1.6%	75.9%	38.5%	1.4%	0.1%
	LEF	3357	345,641	1.6%	8.3%	8.3%	1.6%	0.0%	2190	361,599	0.2%	5.0%	5.0%	0.2%	0.0%	**1**	35	0.0%	0.1%	0.0%	0.1%	0.1%
	CF^P	10	0	0.0%	3.8%	0.0%	0.0%	0.0%	20	0	0.0%	7.8%	0.0%	0.0%	0.0%	25	370	0.0%	17.8%	0.1%	0.1%	0.1%
	CEF	6	14	0.0%	0.7%	0.2%	0.0%	0.0%	**4**	23	0.0%	0.6%	0.2%	0.0%	0.0%	6	17	0.0%	0.3%	0.0%	0.1%	0.1%
	LF^P_{log}	**1**	86	0.0%	1.9%	0.1%	0.0%	0.0%	6	2,434	0.0%	4.4%	0.1%	0.0%	0.0%	3600	1,535,465	0.8%	12.1%	1.5%	0.7%	0.1%
	CEF^P_{log}	2	215	0.0%	1.2%	0.3%	0.0%	0.0%	22	3,199	0.0%	3.7%	2.1%	0.0%	0.0%	3600	411,139	0.3%	12.1%	8.4%	0.2%	0.1%
200,20*	LF_{log}	3600	549,079	6.7%	52.9%	24.7%	6.1%	0.6%	3600	383,827	8.7%	64.7%	31.9%	7.7%	1.1%	3600	300,111	24.1%	82.7%	49.9%	21.9%	2.9%
	LEF	3600	42,413	8.6%	10.6%	10.6%	8.6%	0.0%	3600	129,049	1.1%	3.0%	3.0%	1.0%	0.1%	**29**	1,327	0.0%	0.3%	0.3%	0.1%	0.1%
	CF^P	**27**	5	0.0%	5.6%	0.0%	0.0%	0.0%	64	131	0.0%	13.1%	0.0%	0.0%	0.1%	1562	26,768	0.2%	23.6%	0.5%	0.1%	0.1%
	CEF	73	190	0.0%	1.5%	0.4%	0.0%	0.0%	**40**	31	0.0%	1.5%	0.1%	0.1%	0.1%	59	332	0.0%	1.1%	0.1%	0.1%	0.1%
	LF^P_{log}	710	158,569	0.0%	4.1%	0.2%	0.0%	0.0%	3400	715,941	0.3%	10.9%	0.4%	0.2%	0.1%	3600	374,382	6.3%	22.3%	6.0%	5.6%	0.7%
	CEF^P_{log}	2353	74,047	0.5%	2.6%	2.2%	0.4%	0.1%	3600	112,151	2.2%	8.2%	6.8%	1.8%	0.4%	3600	144,453	6.4%	22.2%	13.8%	4.8%	1.7%
500,50*	LF_{log}	3600	102,004	39.8%	53.0%	35.4%	31.8%	16.0%	3600	84,392	54.0%	68.6%	35.0%	31.4%	137.4%	3600	92,414	55.7%	91.8%	74.7%	53.1%	5.9%
	LEF	3600	2,548	8.3%	8.7%	8.7%	8.2%	0.1%	**2520**	8,118	0.2%	0.8%	0.2%	0.6%	0.8%	3501	12,727	**0.4%**	1.8%	0.4%	0.9%	1.3%
	CF^P	**1194**	311	0.0%	8.9%	0.1%	0.1%	0.1%	3452	707	0.3%	22.2%	0.7%	0.7%	1.0%	3600	440	7.7%	26.7%	0.5%	0.4%	10.0%
	CEF	3611	779	0.2%	32.6%	0.3%	0.1%	0.1%	2620	842	0.0%	38.4%	0.5%	0.7%	0.8%	3604	272	0.5%	23.9%	0.7%	0.8%	1.3%
	LF^P_{log}	3600	110,452	0.8%	8.8%	0.3%	0.3%	0.5%	3600	57,797	3.3%	24.8%	1.3%	1.3%	2.1%	3600	65,850	15.2%	33.6%	13.4%	13.0%	2.6%
	CEF^P_{log}	3600	55,687	4.7%	5.4%	5.0%	4.3%	0.4%	3600	63,450	12.2%	15.8%	14.0%	11.5%	0.8%	3601	129,520	26.1%	33.5%	20.6%	16.2%	13.5%
1000,100**	LF_{log}	3600	55,776	55.9%	60.6%	51.0%	46.3%	25.3%	3600	58,847	62.7%	77.4%	61.2%	59.1%	9.5%	3600	55,641	76.5%	94.3%	79.1%	73.8%	11.5%
	LEF	3600	215	13.9%	4.4%	4.4%	4.3%	16.8%	3722	488	**0.9%**	1.2%	0.0%	0.1%	1.0%	3600	351	**1.7%**	1.7%	1.1%	1.0%	0.7%
	CF^P	3600	†	†	†	†	†	†	3600	†	†	†	†	†	†	3600	†	†	†	†	†	†
	CEF	3605	†	†	†	†	†	†	3600	†	†	†	†	†	†	3600	†	†	†	†	†	†
	LF^P_{log}	3601	†	†	†	†	†	†	3601	6,378	20.9%	39.4%	15.8%	15.6%	6.7%	3601	30,129	26.1%	43.6%	23.9%	23.4%	3.6%
	CEF^P_{log}	3601	32,326	**10.0%**	9.9%	9.5%	9.1%	1.0%	3600	26,843	22.6%	23.9%	22.5%	21.8%	1.1%	3600	4,283	33.8%	42.9%	25.0%	24.6%	14.0%
2000,100**	LF_{log}	3600	58,217	57.8%	68.0%	62.7%	55.1%	6.3%	3600	56,546	70.5%	84.0%	79.1%	67.9%	9.7%	3600	39,585	78.3%	96.0%	81.6%	77.7%	2.7%
	LEF	3601	†	†	†	†	†	†	3600	†	†	†	†	†	†	3601	†	†	†	†	†	†
	CF^P	3600	†	†	†	†	†	†	3600	†	†	†	†	†	†	3600	†	†	†	†	†	†
	CEF	3600	†	†	†	†	†	†	3600	†	†	†	†	†	†	3600	†	†	†	†	†	†
	LF^P_{log}	3601	†	†	†	†	†	†	3600	32,898	41.4%	48.9%	41.2%	39.5%	3.2%	3601	8,660	**33.1%**	52.2%	31.2%	31.1%	3.0%
	CEF^P_{log}	3600	38,716	**16.1%**	15.8%	15.4%	14.7%	1.6%	3600	28,575	**30.7%**	32.5%	31.9%	30.6%	0.1%	3600	931	53.4%	48.2%	33.0%	33.0%	268.5%
5000,100**	LF_{log}	3600	23,558	78.1%	86.8%	84.7%	77.3%	3.9%	3600	37,298	80.6%	93.4%	93.4%	80.2%	1.9%	3601	12,870	83.5%	96.8%	96.8%	83.0%	3.6%
	LEF	7807	†	†	†	†	†	†	8155	†	†	†	†	†	†	7241	†	†	†	†	†	†
	CF^P	3600	†	†	†	†	†	†	3600	†	†	†	†	†	†	3600	†	†	†	†	†	†
	CEF	3600	†	†	†	†	†	†	3600	†	†	†	†	†	†	3600	†	†	†	†	†	†
	LF^P_{log}	3601	7,220	**29.2%**	50.1%	25.1%	25.0%	5.9%	3601	15,186	49.0%	59.9%	47.6%	45.8%	6.3%	3601	6,818	**50.7%**	61.1%	49.3%	49.2%	3.1%
	CEF^P_{log}	3600	13,966	39.3%	27.5%	30.7%	28.7%	402.8%	3600	13,736	**40.6%**	33.9%	40.1%	39.1%	2.6%	3600	3,257	58.4%	50.8%	47.5%	47.5%	26.4%
10000,100**	LF_{log}	3600	13,230	88.4%	90.0%	90.0%	82.0%	224.1%	3600	8,857	83.1%	94.7%	94.7%	80.6%	39.2%	3602	5,082	93.0%	97.6%	97.6%	91.6%	22.0%
	LEF	4225	†	†	†	†	†	†	4026	†	†	†	†	†	†	3603	†	†	†	†	†	†
	CF^P	3600	†	†	†	†	†	†	3600	†	†	†	†	†	†	3600	†	†	†	†	†	†
	CEF	3600	†	†	†	†	†	†	3600	†	†	†	†	†	†	3600	†	†	†	†	†	†
	LF^P_{log}	3601	5,481	55.4%	58.6%	54.7%	53.2%	5.4%	3601	9,440	53.2%	61.4%	49.2%	48.2%	10.5%	3601	5,482	**54.7%**	65.0%	54.3%	54.3%	1.0%
	CEF^P_{log}	3600	9,979	**33.4%**	5.0%	34.9%	31.0%	3.7%	3601	7,247	**45.4%**	22.0%	37.6%	35.6%	17.9%	3601	†	†	†	†	†	†

**easy* instances

***hard* instances

Table 18: Computational results to evaluate the best existing methods in the literature against the standout formulations for the uniformly generated data set [16]. For each combination of n, m, κ and each formulation, we present averages over five instances for: time (`Time`) in seconds, number of nodes (`Nodes`) processed, end gap (`End-gap`), continuous relaxation gap (`Rlx-gap`), root node gap (`Ron-gap`), best-bound gap (`Bbn-gap`), and optimality gap (`Opt-gap`). For each choice of n, m, and κ, among the solution methods, the best average time and the best average `End-gap` (if `Time`$\geqslant$ 3600 sec.) are in **bold**.

	κ	$10\% \cdot n$						
n, m	Ref.	Time	Nodes	End-gap	Rlx-gap	Ron-gap	Bbn-gap	Opt-gap
25,2*	LF_{log}	**0**	23	0.0%	49.8%	28.0%	0.0%	0.0%
	LEF	**0**	1	0.0%	6.8%	5.4%	0.0%	0.0%
	CF^P	3	2	0.0%	31.4%	4.6%	0.0%	0.0%
	CEF	**0**	2	0.0%	4.0%	3.8%	0.0%	0.0%
	LF^P_{log}	**0**	13	0.0%	29.5%	11.5%	0.0%	0.0%
	CEF^P_{log}	1	4	0.0%	10.3%	6.9%	0.0%	0.0%
50,5*	LF_{log}	3	3,364	0.0%	50.7%	43.7%	0.0%	0.0%
	LEF	**2**	381	0.0%	18.5%	16.4%	0.0%	0.0%
	CF^P	78	22,068	0.0%	56.2%	23.9%	0.0%	0.0%
	CEF	3	172	0.0%	14.6%	13.7%	0.0%	0.0%
	LF^P_{log}	9	5,014	0.0%	45.9%	21.3%	0.0%	0.0%
	CEF^P_{log}	6	2,477	0.0%	24.4%	22.6%	0.0%	0.0%
100,10**	LF_{log}	3600	2,079,337	**5.0%**	54.5%	48.7%	4.8%	0.2%
	LEF	3600	480,988	12.3%	29.4%	27.5%	12.3%	0.0%
	CF^P	3600	462,737	43.5%	72.8%	50.8%	41.0%	4.3%
	CEF	3600	221,990	10.7%	25.6%	24.8%	10.3%	0.4%
	LF^P_{log}	3600	2,588,756	7.5%	54.1%	45.5%	6.8%	0.8%
	CEF^P_{log}	3600	482,188	7.2%	34.8%	34.5%	6.3%	0.9%
200,20**	LF_{log}	3600	612,063	41.7%	56.8%	54.5%	37.5%	7.3%
	LEF	3600	47,486	**30.0%**	36.2%	35.4%	30.0%	0.0%
	CF^P	3600	25,375	65.8%	80.4%	66.0%	63.7%	6.0%
	CEF	3600	20,677	30.9%	30.9%	33.3%	28.9%	2.9%
	LF^P_{log}	3600	1,104,491	41.6%	56.7%	53.5%	36.5%	8.7%
	CEF^P_{log}	3600	174,404	35.5%	40.1%	39.9%	31.2%	6.6%
500,50**	LF_{log}	3600	81,055	48.7%	49.0%	48.9%	47.0%	3.2%
	LEF	3600	636	**42.8%**	43.0%	42.7%	42.4%	0.7%
	CF^P	3600	†	†	†	†	†	†
	CEF	3603	17	**42.8%**	20.4%	41.3%	41.2%	2.8%
	LF^P_{log}	3600	108,291	48.4%	49.0%	49.0%	46.8%	3.1%
	CEF^P_{log}	3600	29,818	46.3%	45.2%	44.9%	42.7%	7.1%
1000,100**	LF_{log}	3600	52,994	50.3%	48.7%	48.7%	48.0%	4.7%
	LEF	3601	†	†	†	†	†	†
	CF^P	3600	†	†	†	†	†	†
	CEF	3600	†	†	†	†	†	†
	LF^P_{log}	3600	48,719	50.2%	48.7%	48.7%	48.0%	4.4%
	CEF^P_{log}	3600	10,436	**48.0%**	45.3%	45.3%	44.5%	6.9%
2000,100**	LF_{log}	3600	41,092	50.7%	51.2%	51.2%	50.6%	0.1%
	LEF	3601	†	†	†	†	†	†
	CF^P	3600	†	†	†	†	†	†
	CEF	3600	†	†	†	†	†	†
	LF^P_{log}	3600	15,925	50.8%	51.1%	51.2%	50.8%	0.0%
	CEF^P_{log}	3600	9,576	**47.8%**	48.3%	48.2%	47.7%	0.1%
5000,100**	LF_{log}	3600	18,499	67.9%	68.6%	68.6%	67.1%	2.8%
	LEF	4755	†	†	†	†	†	†
	CF^P	3600	†	†	†	†	†	†
	CEF	3600	†	†	†	†	†	†
	LF^P_{log}	3600	9,434	68.8%	68.6%	68.6%	68.0%	2.7%
	CEF^P_{log}	3600	4,295	**46.7%**	47.2%	47.0%	46.7%	0.0%
10000,100**	LF_{log}	3600	15,052	68.6%	69.0%	69.0%	68.1%	1.6%
	LEF	9500	†	†	†	†	†	†
	CF^P	3600	†	†	†	†	†	†
	CEF	3600	†	†	†	†	†	†
	LF^P_{log}	3601	5,732	68.5%	69.0%	69.0%	68.0%	1.6%
	CEF^P_{log}	3601	896	**47.5%**	47.9%	47.7%	47.5%	0.0%

	κ	$20\% \cdot n$						
n, m	Ref.	Time	Nodes	End-gap	Rlx-gap	Ron-gap	Bbn-gap	Opt-gap
25,2*	LF_{log}	1	95	0.0%	50.1%	32.3%	0.0%	0.0%
	LEF	**0**	24	0.0%	10.8%	9.2%	0.0%	0.0%
	CF^P	4	179	0.0%	40.5%	6.3%	0.0%	0.0%
	CEF	**0**	14	0.0%	7.9%	7.0%	0.0%	0.0%
	LF^P_{log}	1	222	0.0%	42.3%	12.0%	0.0%	0.0%
	CEF^P_{log}	1	106	0.0%	19.9%	18.7%	0.0%	0.0%
50,5*	LF_{log}	20	21,061	0.0%	54.2%	45.0%	0.0%	0.0%
	LEF	**13**	9,831	0.0%	20.9%	19.9%	0.0%	0.0%
	CF^P	3601	1,058,360	6.5%	56.3%	25.4%	6.4%	0.1%
	CEF	18	6,093	0.0%	15.6%	15.3%	0.0%	0.0%
	LF^P_{log}	27	29,157	0.0%	51.4%	30.0%	0.0%	0.0%
	CEF^P_{log}	26	8,630	0.0%	29.9%	29.1%	0.0%	0.0%
100,10**	LF_{log}	3600	2,153,102	**5.0%**	56.4%	49.8%	4.8%	0.3%
	LEF	3600	616,551	17.1%	30.4%	29.7%	17.1%	0.0%
	CF^P	3600	166,635	44.3%	66.9%	47.3%	42.1%	4.1%
	CEF	3600	275,594	15.5%	25.0%	25.7%	14.9%	0.8%
	LF^P_{log}	3600	2,821,692	6.1%	56.3%	48.3%	5.3%	0.9%
	CEF^P_{log}	3603	463,914	5.2%	36.6%	36.6%	3.6%	1.7%
200,20**	LF_{log}	3600	490,278	37.7%	58.0%	54.7%	33.9%	6.0%
	LEF	3600	58,945	31.1%	36.0%	35.6%	31.0%	0.1%
	CF^P	3600	1,113	61.6%	72.0%	58.9%	58.8%	7.1%
	CEF	3600	22,387	**30.0%**	23.0%	32.7%	28.9%	1.5%
	LF^P_{log}	3600	938,882	35.6%	57.9%	54.5%	31.5%	6.3%
	CEF^P_{log}	3600	113,509	34.3%	40.0%	39.9%	30.8%	5.4%
500,50**	LF_{log}	3600	60,815	48.7%	47.2%	47.1%	45.0%	7.2%
	LEF	3600	1,324	**41.1%**	38.6%	38.5%	38.0%	5.2%
	CF^P	3600	†	†	†	†	†	†
	CEF	3604	26	41.8%	5.7%	37.3%	37.2%	7.8%
	LF^P_{log}	3600	70,193	48.1%	47.2%	47.1%	44.6%	6.7%
	CEF^P_{log}	3600	24,696	43.1%	40.7%	40.7%	39.1%	7.0%
1000,100**	LF_{log}	3600	41,825	50.1%	50.9%	50.8%	50.0%	0.0%
	LEF	3601	†	†	†	†	†	†
	CF^P	3600	†	†	†	†	†	†
	CEF	3600	†	†	†	†	†	†
	LF^P_{log}	3600	24,225	50.2%	50.7%	50.8%	50.2%	0.0%
	CEF^P_{log}	3600	9,445	**44.5%**	44.7%	45.0%	44.5%	0.1%
2000,100**	LF_{log}	3600	30,062	50.6%	50.8%	50.7%	50.2%	0.8%
	LEF	3602	†	†	†	†	†	†
	CF^P	3600	†	†	†	†	†	†
	CEF	3600	†	†	†	†	†	†
	LF^P_{log}	3600	14,228	50.7%	50.8%	50.8%	50.3%	0.8%
	CEF^P_{log}	3600	7,815	**44.6%**	45.1%	45.0%	44.6%	0.1%
5000,100**	LF_{log}	3600	34,661	65.0%	69.5%	69.5%	65.0%	0.0%
	LEF	3938	†	†	†	†	†	†
	CF^P	3600	†	†	†	†	†	†
	CEF	3600	†	†	†	†	†	†
	LF^P_{log}	3600	12,867	67.9%	69.5%	69.2%	67.9%	0.1%
	CEF^P_{log}	3601	3,406	**45.2%**	45.7%	45.5%	45.1%	0.0%
10000,100**	LF_{log}	3600	11,855	68.2%	69.2%	69.2%	67.9%	1.0%
	LEF	6022	†	†	†	†	†	†
	CF^P	3600	†	†	†	†	†	†
	CEF	3600	†	†	†	†	†	†
	LF^P_{log}	3601	6,058	68.4%	69.2%	68.8%	68.0%	1.2%
	CEF^P_{log}	3600	1,165	**44.8%**	45.1%	45.0%	44.8%	0.0%

	κ	Unconstrained						
n, m	Ref.	Time	Nodes	End-gap	Rlx-gap	Ron-gap	Bbn-gap	Opt-gap
25,2*	LF_{log}	1	199	0.0%	93.0%	59.8%	0.0%	0.0%
	LEF	**0**	21	0.0%	28.1%	22.7%	0.0%	0.0%
	CF^P	4	24	0.0%	37.1%	3.4%	0.0%	0.0%
	CEF	1	16	0.0%	18.6%	13.2%	0.0%	0.0%
	LF^P_{log}	1	142	0.0%	45.5%	14.6%	0.0%	0.0%
	CEF^P_{log}	6	133	0.0%	42.8%	27.1%	0.0%	0.0%
50,5*	LF_{log}	52	55,437	0.0%	96.9%	77.1%	0.0%	0.0%
	LEF	**43**	35,334	0.0%	56.3%	49.6%	0.0%	0.0%
	CF^P	2903	1,043,778	3.0%	56.6%	23.9%	3.0%	0.0%
	CEF	100	35,410	0.0%	43.6%	40.9%	0.0%	0.0%
	LF^P_{log}	85	81,310	0.0%	60.3%	25.0%	0.0%	0.0%
	CEF^P_{log}	86	25,435	0.0%	59.4%	46.4%	0.0%	0.0%
100,10**	LF_{log}	3600	2,487,103	11.2%	98.6%	84.6%	10.7%	0.6%
	LEF	3600	654,126	38.5%	72.5%	66.6%	38.5%	0.0%
	CF^P	3600	330,256	42.0%	71.5%	50.4%	40.4%	2.9%
	CEF	3600	130,787	40.1%	63.7%	61.1%	39.3%	1.5%
	LF^P_{log}	3600	1,928,384	17.2%	74.5%	48.6%	16.1%	1.4%
	CEF^P_{log}	3600	417,221	**10.9%**	73.6%	61.2%	10.9%	0.0%
200,20**	LF_{log}	3600	519,981	58.2%	99.3%	89.9%	54.9%	7.9%
	LEF	3600	63,610	70.6%	83.0%	78.8%	70.5%	0.3%
	CF^P	3600	7,872	70.9%	81.0%	66.3%	64.8%	21.0%
	CEF	3600	4,559	76.4%	76.0%	75.8%	71.3%	21.6%
	LF^P_{log}	3600	434,136	58.0%	82.1%	65.9%	52.7%	13.0%
	CEF^P_{log}	3600	279,263	**54.4%**	81.4%	73.4%	50.1%	9.4%
500,50**	LF_{log}	3600	139,697	87.0%	99.9%	96.1%	86.1%	7.1%
	LEF	3600	113	90.3%	91.4%	89.3%	89.2%	11.5%
	CF^P	3600	1,208	84.9%	89.8%	83.1%	82.5%	15.9%
	CEF	3603	1	93.4%	80.5%	90.9%	90.9%	51.3%
	LF^P_{log}	3600	181,247	**82.9%**	90.5%	85.9%	81.3%	9.9%
	CEF^P_{log}	3600	23,870	86.7%	89.7%	86.6%	84.6%	20.1%
1000,100**	LF_{log}	3600	48,644	96.6%	99.9%	97.3%	96.3%	8.5%
	LEF	3601	†	†	†	†	†	†
	CF^P	3600	406	95.6%	83.4%	95.5%	95.1%	27.6%
	CEF	3600	†	†	†	†	†	†
	LF^P_{log}	3600	108,734	**91.9%**	93.1%	92.0%	91.2%	8.4%
	CEF^P_{log}	3600	476	92.2%	92.5%	90.2%	90.2%	35.7%
2000,100**	LF_{log}	3600	35,408	97.8%	100.0%	98.2%	97.7%	3.4%
	LEF	3601	†	†	†	†	†	†
	CF^P	3600	†	†	†	†	†	†
	CEF	3600	†	†	†	†	†	†
	LF^P_{log}	3600	69,565	**94.8%**	95.5%	95.1%	94.7%	2.6%
	CEF^P_{log}	3600	339	96.6%	95.1%	93.7%	93.7%	101.2%
5000,100**	LF_{log}	3601	13,907	98.8%	100.0%	98.8%	98.6%	22.2%
	LEF	3603	†	†	†	†	†	†
	CF^P	3600	†	†	†	†	†	†
	CEF	3600	†	†	†	†	†	†
	LF^P_{log}	3601	16,900	**96.9%**	96.7%	96.5%	96.4%	14.4%
	CEF^P_{log}	3601	34	98.3%	96.4%	96.0%	96.0%	132.1%
10000,100**	LF_{log}	3601	2,471	99.4%	100.0%	99.3%	99.3%	25.6%
	LEF	5619	†	†	†	†	†	†
	CF^P	3600	†	†	†	†	†	†
	CEF	3600	†	†	†	†	†	†
	LF^P_{log}	3601	6,595	**97.8%**	98.0%	97.9%	97.8%	0.0%
	CEF^P_{log}	3600	†	†	†	†	†	†

**easy* instances

***hard* instances

Table 19: Computational results to compare basic MILP and MICQP formulations for the assortment data set [85]. For each combination of n, m, κ and each formulation, we present averages over five instances for: time (`Time`) in seconds, number (`#`) of instances solved to optimality, number of nodes (`Nodes`) processed, end gap (`End-gap`), continuous relaxation gap (`Rlx-gap`), root-node gap (`Ron-gap`), best bound gap (`Bbn-gap`) the optimality gap (`Opt-gap`). For each choice of n, m, and κ, among the solution methods, the best average time and the best average `End-gap` (if `Time`$\geqslant$ 3600 sec.) are in **bold**.

	κ	$10\% \cdot n$						$20\% \cdot n$						Unconstrained					
n, m	Ref.	`Time`	`#`	`Nodes`	`End-gap`	`Rlx-gap`	`Ron-gap`	`Time`	`#`	`Nodes`	`End-gap`	`Rlx-gap`	`Ron-gap`	`Time`	`#`	`Nodes`	`End-gap`	`Rlx-gap`	`Ron-gap`
25,2	LF	**0**	5	30	0.0%	14.1%	4.7%	2	5	1,395	0.0%	42.6%	24.5%	2	5	1,538	0.0%	35.0%	12.1%
	LEF	**0**	5	2	0.0%	0.7%	0.7%	**0**	5	9	0.0%	0.9%	0.8%	**0**	5	0	0.0%	0.0%	0.0%
	CF	1	5	13	0.0%	1.5%	1.5%	1	5	70	0.0%	2.8%	2.8%	2	5	1,278	0.0%	8.2%	8.2%
	CEF	**0**	5	0	0.0%	0.0%	0.0%	**0**	5	1	0.0%	0.1%	0.1%	**0**	5	0	0.0%	0.0%	0.0%
50,5	LF	55	5	40,100	0.0%	51.7%	41.3%	3600	0	935,667	29.4%	60.1%	49.8%	3600	0	1,495,669	19.3%	52.9%	41.4%
	LEF	**0**	5	123	0.0%	3.0%	3.0%	**1**	5	1,111	0.0%	3.2%	3.2%	**0**	5	2	0.0%	0.0%	0.0%
	CF	2	5	157	0.0%	2.3%	2.2%	7	5	3,016	0.0%	5.5%	5.5%	2916	2	802,844	1.0%	13.4%	13.4%
	CEF	1	5	1	0.0%	0.0%	0.0%	**1**	5	7	0.0%	0.2%	0.2%	1	5	2	0.0%	0.0%	0.0%
100,10	LF	3600	0	170,481	70.1%	78.3%	77.7%	3600	0	473,158	61.7%	72.1%	70.2%	3600	0	730,892	51.5%	67.5%	64.2%
	LEF	3357	1	345,641	1.6%	8.3%	8.3%	2190	3	361,599	0.2%	5.0%	5.0%	**1**	5	35	0.0%	0.1%	0.0%
	CF	67	5	7,786	0.0%	3.8%	3.6%	3371	1	231,729	3.5%	7.8%	8.6%	3600	0	94,567	11.9%	17.8%	17.9%
	CEF	**6**	5	14	0.0%	0.7%	0.2%	**4**	5	23	0.0%	0.6%	0.2%	6	5	17	0.0%	0.3%	0.0%
200,20	LF	3600	0	28,528	82.3%	84.4%	84.4%	3600	0	47,569	77.0%	79.4%	79.3%	3600	0	49,188	73.4%	78.3%	77.9%
	LEF	3600	0	42,413	8.6%	10.6%	10.6%	3600	1	129,049	1.1%	3.0%	3.0%	**29**	5	1,327	0.0%	0.3%	0.3%
	CF	3600	0	55,709	32.9%	5.6%	36.0%	3600	0	29,821	62.3%	13.1%	63.7%	3600	0	14,939	65.2%	23.6%	73.1%
	CEF	**73**	5	190	0.0%	1.5%	0.4%	**40**	5	31	0.0%	1.5%	0.1%	59	5	332	0.0%	1.1%	0.1%
500,50	LF	3600	0	1,620	90.3%	89.0%	89.0%	3600	0	2,097	86.7%	86.2%	86.2%	3601	0	5,755	86.2%	86.4%	86.4%
	LEF	3600	0	2,548	8.3%	8.7%	8.7%	**2520**	2	8,118	0.2%	0.8%	0.2%	3501	1	12,727	**0.4%**	1.8%	0.4%
	CF	3600	0	11,986	96.4%	8.9%	100.0%	3600	0	11,367	100.0%	22.2%	100.0%	3600	0	4,110	96.1%	26.7%	100.0%
	CEF	3611	0	779	**0.2%**	32.6%	0.3%	2620	5	842	0.0%	38.4%	0.5%	3604	0	272	0.5%	23.9%	0.7%
1000,100	LF	3601	0	3	99.2%	93.1%	93.1%	3600	0	10	99.0%	90.4%	90.4%	3601	0	33	99.0%	90.5%	90.5%
	LEF	3600	0	215	**13.9%**	4.4%	4.4%	3722	0	488	**0.9%**	1.2%	0.0%	3600	0	351	**1.7%**	1.7%	1.1%
	CF	3600	0	4,962	100.0%	15.7%	100.0%	3605	0	4,612	100.0%	29.7%	100.0%	3600	0	1,882	100.0%	30.2%	100.0%
	CEF	3605	0	†	†	†	†	3600	0	†	†	†	†	3600	0	†	†	†	†

Table 20: Computational results to compare basic MILP and MICQP formulations for the uniformly generated data set [16]. For each combination of n, m, κ and each formulation, we present averages over five instances for: time (`Time`) in seconds, number (`#`) of instances solved to optimality, number of nodes (`Nodes`) processed, end gap (`End-gap`), continuous relaxation gap (`Rlx-gap`), root-node gap (`Ron-gap`), best bound gap (`Bbn-gap`) and optimality gap (`Opt-gap`). For each choice of n, m, and κ, among the solution methods, the best average time and the best average `End-gap` (if `Time`$\geqslant$ 3600 sec.) are in **bold**.

	κ	$10\% \cdot n$						$20\% \cdot n$						Unconstrained					
n, m	Ref.	Time	#	Nodes	End-gap	Rlx-gap	Ron-gap	Time	#	Nodes	End-gap	Rlx-gap	Ron-gap	Time	#	Nodes	End-gap	Rlx-gap	Ron-gap
25,2	LF	**0**	5	38	0.0%	81.5%	48.1%	1	5	433	0.0%	70.4%	41.5%	1	5	497	0.0%	89.7%	46.9%
	LEF	**0**	5	1	0.0%	6.8%	5.4%	**0**	5	24	0.0%	10.8%	9.2%	**0**	5	21	0.0%	28.1%	22.7%
	CF	1	5	114	0.0%	31.4%	24.8%	1	5	1,198	0.0%	40.5%	40.5%	1	5	1,657	0.0%	37.1%	37.1%
	CEF	**0**	5	2	0.0%	4.0%	3.8%	**0**	5	14	0.0%	7.9%	7.0%	1	5	16	0.0%	18.6%	13.2%
50,5	LF	1554	5	589,909	0.0%	85.7%	82.8%	3600	0	2,006,223	46.9%	75.2%	69.9%	3600	0	2,791,336	44.9%	96.0%	91.5%
	LEF	**2**	5	381	0.0%	18.5%	16.4%	**13**	5	9,831	0.0%	20.9%	19.9%	**43**	5	35,334	0.0%	56.3%	49.6%
	CF	75	5	31,692	0.0%	56.2%	56.2%	3606	0	1,112,962	6.0%	56.3%	56.3%	2594	2	677,850	7.3%	56.6%	56.6%
	CEF	3	5	172	0.0%	14.6%	13.7%	18	5	6,093	0.0%	15.6%	15.3%	100	5	35,410	0.0%	43.6%	40.9%
100,10	LF	3600	0	1,100,713	82.1%	87.5%	87.1%	3600	0	1,252,405	71.1%	77.1%	76.0%	3600	0	1,308,606	90.2%	98.3%	96.7%
	LEF	3600	0	480,988	12.3%	29.4%	27.5%	3600	0	616,551	17.1%	30.4%	29.7%	3600	0	654,126	**38.5%**	72.5%	66.6%
	CF	3600	0	223,083	52.3%	72.8%	72.8%	3600	0	154,439	54.9%	66.9%	66.9%	3600	0	220,110	53.2%	71.5%	71.4%
	CEF	3600	0	221,990	**10.7%**	25.6%	24.8%	3600	0	275,594	**15.5%**	25.0%	25.7%	3600	0	130,787	40.1%	63.7%	61.1%
200,20	LF	3600	0	118,471	87.8%	88.6%	88.6%	3600	0	107,640	77.4%	78.1%	77.9%	3600	0	184,061	97.5%	99.2%	98.5%
	LEF	3600	0	47,486	**30.0%**	36.2%	35.4%	3600	0	58,945	31.1%	36.0%	35.6%	3600	0	63,610	**70.6%**	83.0%	78.8%
	CF	3600	0	27,323	99.5%	80.4%	99.5%	3600	0	17,547	89.6%	72.0%	88.7%	3600	0	24,168	99.7%	81.0%	100.0%
	CEF	3600	0	20,677	30.9%	30.9%	33.3%	3600	0	22,387	**30.0%**	23.0%	32.7%	3600	0	4,559	76.4%	76.0%	75.8%
500,50	LF	3600	0	1,232	90.2%	89.4%	89.4%	3600	0	369	82.3%	78.3%	78.3%	3600	0	6,619	99.5%	99.8%	99.5%
	LEF	3600	0	636	**42.8%**	43.0%	42.7%	3600	0	1,324	**41.1%**	38.6%	38.5%	3600	0	113	**90.3%**	91.4%	89.3%
	CF	3600	0	9,997	100.0%	86.1%	100.0%	3602	0	3,292	100.0%	75.0%	100.0%	3600	0	13,213	100.0%	89.8%	100.0%
	CEF	3603	0	17	**42.8%**	20.4%	41.3%	3604	0	26	41.8%	5.7%	37.3%	3603	0	1	93.4%	80.5%	90.9%
1000,100	LF	3601	0	†	†	†	†	3600	0	21	**81.6%**	79.9%	79.9%	3601	0	4	**99.9%**	99.9%	99.8%
	LEF	3601	0	†	†	†	†	3601	0	†	†	†	†	3601	0	†	†	†	†
	CF	3600	0	911	**100.0%**	87.3%	100.0%	3600	0	1,210	100.0%	77.8%	100.0%	3600	0	663	100.0%	92.8%	100.0%
	CEF	3600	0	†	†	†	†	3600	0	†	†	†	†	3600	0	†	†	†	†

Table 21: Computational results to compare binary-expansion formulations with their basic counterparts for the assortment data set [16]. For each combination of n, m, κ and each formulation, we present averages over five instances for: time (`Time`) in seconds, number of nodes (`Nodes`) processed, end gap (`End-gap`), continuous relaxation gap (`Rlx-gap`), and root-node gap (`Ron-gap`). For each choice of n, m, and κ, among the solution methods, the best average time and the best average `End-gap` (if `Time`$\geqslant$ 3600 sec.) are in **bold**.

	κ	$10\% \cdot n$					$20\% \cdot n$					Unconstrained				
n, m	Ref.	`Time`	`Nodes`	`End-gap`	`Rlx-gap`	`Ron-gap`	`Time`	`Nodes`	`End-gap`	`Rlx-gap`	`Ron-gap`	`Time`	`Nodes`	`End-gap`	`Rlx-gap`	`Ron-gap`
25,2	LF	**0**	30	0.0%	14.1%	4.7%	2	1,395	0.0%	42.6%	24.5%	2	1,538	0.0%	35.0%	12.1%
	LF_{log}	**0**	5	0.0%	15.3%	1.1%	**0**	27	0.0%	28.1%	4.3%	1	386	0.0%	56.1%	18.1%
	CEF	**0**	0	0.0%	0.0%	0.0%	**0**	1	0.0%	0.1%	0.1%	**0**	0	0.0%	0.0%	0.0%
	CEF_{log}	1	1	0.0%	0.1%	0.1%	1	3	0.0%	0.7%	0.3%	1	316	0.0%	5.8%	5.8%
50,5	LF	55	40,100	0.0%	51.7%	41.3%	3600	935,667	29.4%	60.1%	49.8%	3600	1,495,669	19.3%	52.9%	41.4%
	LF_{log}	**1**	233	0.0%	30.0%	5.7%	2	2,109	0.0%	44.4%	14.5%	18	35,496	0.0%	65.6%	28.2%
	CEF	**1**	1	0.0%	0.0%	0.0%	**1**	7	0.0%	0.2%	0.2%	**1**	2	0.0%	0.0%	0.0%
	CEF_{log}	**1**	6	0.0%	0.6%	0.3%	2	371	0.0%	2.1%	2.0%	18	13,917	0.0%	12.4%	12.3%
100,10	LF	3600	170,481	70.1%	78.3%	77.7%	3600	473,158	61.7%	72.1%	70.2%	3600	730,892	51.5%	67.5%	64.2%
	LF_{log}	979	364,141	0.0%	42.7%	14.5%	3155	1,732,777	0.4%	55.3%	23.7%	3600	1,543,428	1.6%	75.9%	38.5%
	CEF	**6**	14	0.0%	0.7%	0.2%	**4**	23	0.0%	0.6%	0.2%	**6**	17	0.0%	0.3%	0.0%
	CEF_{log}	10	1,457	0.0%	1.3%	1.2%	212	27,571	0.0%	4.0%	3.9%	3465	292,906	1.0%	20.2%	20.2%
200,20	LF	3600	28,528	82.3%	84.4%	84.4%	3600	47,569	77.0%	79.4%	79.3%	3600	49,188	73.4%	78.3%	77.9%
	LF_{log}	3600	549,079	6.7%	52.9%	24.7%	3600	383,827	8.7%	64.7%	31.9%	3600	300,111	24.1%	82.7%	49.9%
	CEF	**73**	190	0.0%	1.5%	0.4%	**40**	31	0.0%	1.5%	0.1%	**59**	332	0.0%	1.1%	0.1%
	CEF_{log}	3600	137,672	0.9%	2.6%	2.6%	3600	121,652	3.8%	8.2%	8.2%	3600	96,988	22.0%	28.9%	29.0%
500,50	LF	3600	1,620	90.3%	89.0%	89.0%	3600	2,097	86.7%	86.2%	86.2%	3601	5,755	86.2%	86.4%	86.4%
	LF_{log}	3600	102,004	39.8%	53.0%	35.4%	3600	84,392	54.0%	68.6%	35.0%	3600	92,414	55.7%	91.8%	74.7%
	CEF	3611	779	**0.2%**	32.6%	0.3%	**2620**	842	0.0%	38.4%	0.5%	3604	272	**0.5%**	23.9%	0.7%
	CEF_{log}	3600	49,090	5.1%	5.4%	5.4%	3600	55,757	13.3%	15.8%	15.7%	3600	53,280	36.9%	37.3%	38.5%
1000,100	LF	3601	3	99.2%	93.1%	93.1%	3600	10	99.0%	90.4%	90.4%	3601	33	99.0%	90.5%	90.5%
	LF_{log}	3600	55,776	55.9%	60.6%	51.0%	3600	58,847	62.7%	77.4%	61.2%	3600	55,641	76.5%	94.3%	79.1%
	CEF	3605	†	†	†	†	3600	†	†	†	†	3600	†	†	†	†
	CEF_{log}	3600	36,151	**25.3%**	9.8%	9.9%	3600	36,647	**23.7%**	23.6%	23.9%	3600	35,213	**48.7%**	44.6%	45.7%
2000,100	LF_{log}	3600	58,217	57.8%	68.0%	62.7%	3600	56,546	70.5%	84.0%	79.1%	3600	39,585	78.3%	96.0%	81.6%
	CEF_{log}	3600	26,386	**30.3%**	15.5%	16.1%	3600	22,548	**60.0%**	31.9%	38.2%	3600	26,785	**71.0%**	50.7%	52.7%
5000,100	LF_{log}	3600	23,558	78.1%	86.8%	84.7%	3600	37,298	80.6%	93.4%	93.4%	3601	12,870	**83.5%**	96.8%	96.8%
	CEF_{log}	3600	15,535	**48.0%**	26.7%	57.7%	3600	10,662	**77.7%**	39.3%	60.0%	3600	11,067	86.5%	57.2%	86.5%
10000,100	LF_{log}	3600	13,230	88.4%	90.0%	90.0%	3600	8,857	**83.1%**	94.7%	94.7%	3602	5,082	**93.0%**	97.6%	97.6%
	CEF_{log}	3600	7,551	**53.8%**	29.5%	52.2%	3600	3,781	84.6%	45.0%	85.1%	3600	2,786	95.4%	70.3%	95.0%

Table 22: Computational results to compare binary-expansion formulations with their basic counterparts for the uniformly generated data set [16]. For each combination of n, m, κ and each formulation, we present averages over five instances for: time (`Time`) in seconds, number of nodes (`Nodes`) processed, end gap (`End-gap`), continuous relaxation gap (`Rlx-gap`), the root-node gap (`Ron-gap`). For each choice of n, m, and κ, among the solution methods, the best average time and the best average `End-gap` (if `Time`$\geqslant$ 3600 sec.) are in **bold**.

	κ	$10\% \cdot n$					$20\% \cdot n$					Unconstrained				
n, m	Ref.	`Time`	`Nodes`	`End-gap`	`Rlx-gap`	`Ron-gap`	`Time`	`Nodes`	`End-gap`	`Rlx-gap`	`Ron-gap`	`Time`	`Nodes`	`End-gap`	`Rlx-gap`	`Ron-gap`
25,2	LF	**0**	38	0.0%	81.5%	48.1%	1	433	0.0%	70.4%	41.5%	**1**	497	0.0%	89.7%	46.9%
	LF_{log}	**0**	23	0.0%	49.8%	28.0%	1	95	0.0%	50.1%	32.3%	**1**	199	0.0%	93.0%	59.8%
	CEF	**0**	2	0.0%	4.0%	3.8%	**0**	14	0.0%	7.9%	7.0%	**1**	16	0.0%	18.6%	13.2%
	CEF_{log}	**0**	6	0.0%	10.3%	7.9%	1	89	0.0%	19.9%	19.3%	3	353	0.0%	45.7%	44.2%
50,5	LF	1554	589,909	0.0%	85.7%	82.8%	3600	2,006,223	46.9%	75.2%	69.9%	3600	2,791,336	44.9%	96.0%	91.5%
	LF_{log}	**3**	3,364	0.0%	50.7%	43.7%	20	21,061	0.0%	54.2%	45.0%	**52**	55,437	0.0%	96.9%	77.1%
	CEF	**3**	172	0.0%	14.6%	13.7%	**18**	6,093	0.0%	15.6%	15.3%	100	35,410	0.0%	43.6%	40.9%
	CEF_{log}	11	4,746	0.0%	24.4%	24.0%	22	8,046	0.0%	30.0%	29.6%	521	78,437	0.0%	64.4%	62.1%
100,10	LF	3600	1,100,713	82.1%	87.5%	87.1%	3600	1,252,405	71.1%	77.1%	76.0%	3600	1,308,606	90.2%	98.3%	96.7%
	LF_{log}	3600	2,079,337	**5.0%**	54.5%	48.7%	3600	2,153,102	**5.0%**	56.4%	49.8%	3600	2,487,103	**11.2%**	98.6%	84.6%
	CEF	3600	221,990	10.7%	25.6%	24.8%	3600	275,594	15.5%	25.0%	25.7%	3600	130,787	40.1%	63.7%	61.1%
	CEF_{log}	3601	433,421	8.1%	34.8%	34.7%	3600	394,433	7.9%	36.6%	36.5%	3600	368,512	20.1%	76.0%	74.6%
200,20	LF	3600	118,471	87.8%	88.6%	88.6%	3600	107,640	77.4%	78.1%	77.9%	3600	184,061	97.5%	99.2%	98.5%
	LF_{log}	3600	612,063	41.7%	56.8%	54.5%	3600	490,278	37.7%	58.0%	54.7%	3600	519,981	**58.2%**	99.3%	89.9%
	CEF	3600	20,677	**30.9%**	30.9%	33.3%	3600	22,387	**30.0%**	23.0%	32.7%	3600	4,559	76.4%	76.0%	75.8%
	CEF_{log}	3600	131,182	39.6%	40.1%	40.1%	3600	88,037	36.6%	40.0%	40.0%	3600	285,525	64.6%	83.6%	83.3%
500,50	LF	3600	1,232	90.2%	89.4%	89.4%	3600	369	82.3%	78.3%	78.3%	3600	6,619	99.5%	99.8%	99.5%
	LF_{log}	3600	81,055	48.7%	49.0%	48.9%	3600	60,815	48.7%	47.2%	47.1%	3600	139,697	**87.0%**	99.9%	96.1%
	CEF	3603	17	**42.8%**	20.4%	41.3%	3604	26	**41.8%**	5.7%	37.3%	3603	1	93.4%	80.5%	90.9%
	CEF_{log}	3600	34,703	53.2%	45.2%	45.1%	3600	26,390	42.8%	40.7%	40.7%	3600	82,878	91.0%	90.8%	90.8%
1000,100	LF	3601	†	91.0%	89.5%	89.5%	3600	21	81.6%	79.9%	79.9%	3601	4	99.9%	99.9%	99.8%
	LF_{log}	3600	52,994	50.3%	48.7%	48.7%	3600	41,825	50.1%	50.9%	50.8%	3600	48,644	96.6%	99.9%	97.3%
	CEF	3600	†	†	†	†	3600	†	†	†	†	3600	†	†	†	†
	CEF_{log}	3600	12,062	**46.0%**	45.3%	45.3%	3601	8,843	**47.9%**	44.7%	45.3%	3600	37,767	**93.7%**	93.3%	93.3%
2000,100	LF_{log}	3600	41,092	50.7%	51.2%	51.2%	3600	30,062	50.6%	50.8%	50.7%	3600	35,408	97.8%	100.0%	98.2%
	CEF_{log}	3601	5,139	**48.8%**	47.9%	48.4%	3600	4,909	**48.5%**	44.4%	45.2%	3600	25,840	**97.0%**	95.5%	95.6%
5000,100	LF_{log}	3600	18,499	67.9%	68.6%	68.6%	3600	34,661	65.0%	69.5%	69.5%	3601	13,907	98.8%	100.0%	98.8%
	CEF_{log}	3600	5,092	**51.4%**	46.4%	47.1%	3600	3,305	**48.0%**	44.6%	45.6%	3600	11,678	**97.0%**	96.7%	96.7%
10000,100	LF_{log}	3600	15,052	68.6%	69.0%	69.0%	3600	11,855	68.2%	69.2%	69.2%	3601	2,471	**99.4%**	100.0%	99.3%
	CEF_{log}	3600	1,873	**50.5%**	47.2%	47.7%	3600	1,010	**48.2%**	44.3%	45.1%	3601	475	**99.4%**	98.0%	99.0%

Table 23: The size of selected formulations versus their binary-expansion versions for the assortment data set [85]. In each row, the average number of continuous (`C-var`) and binary (`B-var`) variables as well as the average number of linear (`L-const`) and rotated conic quadratic (`C-const`) constraints are presented.

	κ	$10\% \cdot n$				$20\% \cdot n$				Unconstrained			
n, m	Ref.	`C-var`	`B-var`	`L-const`	`C-const`	`C-var`	`B-var`	`L-const`	`C-const`	`C-var`	`B-var`	`L-const`	`C-const`
25,2	LF	52	25	203	-	52	25	203	-	52	25	202	-
	LF_{log}	12	35	35	-	14	37	41	-	17	40	50	-
	CEF	56	25	207	52	56	25	207	52	56	25	206	52
	CEF_{log}	16	45	49	12	18	49	57	14	21	55	65	17
50,5	LF	255	50	1,006	-	255	50	1,006	-	255	50	1,005	-
	LF_{log}	35	80	101	-	40	85	116	-	47	92	135	-
	CEF	265	50	1,016	255	265	50	1,016	255	265	50	1,015	255
	CEF_{log}	49	114	152	39	53	123	168	43	57	133	183	47
100,10	LF	1,010	100	4,011	-	1,010	100	4,011	-	1,010	100	4,010	-
	LF_{log}	80	170	231	-	90	180	261	-	104	194	302	-
	CEF	1,030	100	4,031	1,010	1,030	100	4,031	1,010	1,030	100	4,030	1,010
	CEF_{log}	110	250	351	90	114	264	367	94	124	286	406	104
200,20	LF	4,020	200	16,021	-	4,020	200	16,021	-	4,020	200	16,020	-
	LF_{log}	180	360	521	-	200	380	581	-	226	406	657	-
	CEF	4,060	200	16,061	4,020	4,060	200	16,061	4,020	4,060	200	16,060	4,020
	CEF_{log}	240	540	781	200	244	564	797	204	264	608	876	224
500,50	LF	25,050	500	100,051	-	25,050	500	100,051	-	25,050	500	100,050	-
	LF_{log}	500	950	1,451	-	550	1,000	1,601	-	650	1,100	1,900	-
	CEF	25,150	500	100,151	25,050	25,150	500	100,151	25,050	25,150	500	100,150	25,050
	CEF_{log}	650	1,450	2,151	550	700	1,550	2,351	600	750	1,700	2,550	650
1000,100	LF	100,100	1,000	400,101	-	100,100	1,000	400,101	-	100,100	1,000	400,100	-
	LF_{log}	1,100	2,000	3,201	-	1,200	2,100	3,501	-	1,400	2,300	4,100	-
	CEF	100,300	1,000	400,301	100,100	100,300	1,000	400,301	100,100	100,300	1,000	400,300	100,100
	CEF_{log}	1,400	3,100	4,701	1,200	1,500	3,300	5,101	1,300	1,600	3,600	5,500	1,400
2000,100	LF_{log}	1,200	3,100	3,501	-	1,300	3,200	3,801	-	1,500	3,400	4,400	-
	CEF_{log}	1,500	4,300	5,101	1,300	1,600	4,500	5,501	1,400	1,700	4,800	5,900	1,500
5000,100	LF_{log}	1,400	6,300	4,101	-	1,500	6,400	4,401	-	1,600	6,500	4,700	-
	CEF_{log}	1,600	7,600	5,501	1,400	1,700	7,800	5,901	1,500	1,800	8,000	6,300	1,600
10000,100	LF_{log}	1,500	11,400	4,401	-	1,600	11,500	4,701	-	1,700	11,600	5,000	-
	CEF_{log}	1,700	12,800	5,901	1,500	1,800	13,000	6,301	1,600	1,900	13,200	6,700	1,700

Table 24: The size of selected formulations versus their binary-expansion versions for the uniformly generated data set [16]. In each row, the average number of continuous (`C-var`) and binary (`B-var`) variables as well as the average number of linear (`L-const`) and rotated conic quadratic (`C-const`) constraints are presented.

	κ	$10\% \cdot n$				$20\% \cdot n$				Unconstrained			
n, m	Ref.	`C-var`	`B-var`	`L-const`	`C-const`	`C-var`	`B-var`	`L-const`	`C-const`	`C-var`	`B-var`	`L-const`	`C-const`
25,2	LF	52	25	203	-	52	25	203	-	52	25	202	-
	LF_{log}	14	37	41	-	16	39	47	-	19	42	54	-
	CEF	56	25	207	52	56	25	207	52	56	25	206	52
	CEF_{log}	18	49	55	14	20	53	63	16	23	59	73	19
50,5	LF	255	50	1,006	-	255	50	1,006	-	255	50	1,005	-
	LF_{log}	40	85	116	-	45	90	131	-	51	96	149	-
	CEF	265	50	1,016	255	265	50	1,016	255	265	50	1,015	255
	CEF_{log}	50	120	156	40	55	130	176	45	63	144	207	53
100,10	LF	1,010	100	4,011	-	1,010	100	4,011	-	1,010	100	4,010	-
	LF_{log}	90	180	261	-	100	190	291	-	113	203	329	-
	CEF	1,030	100	4,031	1,010	1,030	100	4,031	1,010	1,030	100	4,030	1,010
	CEF_{log}	110	260	351	90	120	280	391	100	132	306	438	112
200,20	LF	4,020	200	16,021	-	4,020	200	16,021	-	4,020	200	16,020	-
	LF_{log}	200	380	581	-	220	400	641	-	244	424	713	-
	CEF	4,060	200	16,061	4,020	4,060	200	16,061	4,020	4,060	200	16,060	4,020
	CEF_{log}	240	560	781	200	260	600	861	220	288	656	972	248
500,50	LF	25,050	500	100,051	-	25,050	500	100,051	-	25,050	500	100,050	-
	LF_{log}	550	1,000	1,601	-	600	1,050	1,751	-	700	1,150	2,050	-
	CEF	25,150	500	100,151	25,050	25,150	500	100,151	25,050	25,150	500	100,150	25,050
	CEF_{log}	650	1,500	2,151	550	700	1,600	2,351	600	800	1,800	2,750	700
1000,100	LF	100,100	1,000	400,101	-	100,100	1,000	400,101	-	100,100	1,000	400,100	-
	LF_{log}	1,200	2,100	3,501	-	1,300	2,200	3,801	-	1,500	2,400	4,400	-
	CEF	100,300	1,000	400,301	100,100	100,300	1,000	400,301	100,100	100,300	1,000	400,300	100,100
	CEF_{log}	1,400	3,200	4,701	1,200	1,500	3,400	5,101	1,300	1,700	3,800	5,900	1,500
2000,100	LF_{log}	1,300	3,200	3,801	-	1,400	3,300	4,101	-	1,600	3,500	4,700	-
	CEF_{log}	1,500	4,400	5,101	1,300	1,600	4,600	5,501	1,400	1,800	5,000	6,300	1,600
5000,100	LF_{log}	1,500	6,400	4,401	-	1,600	6,500	4,701	-	1,700	6,600	5,000	-
	CEF_{log}	1,700	7,800	5,901	1,500	1,800	8,000	6,301	1,600	1,900	8,200	6,700	1,700
10000,100	LF_{log}	1,600	11,500	4,701	-	1,700	11,600	5,001	-	1,800	11,700	5,300	-
	CEF_{log}	1,800	13,000	6,301	1,600	1,900	13,200	6,701	1,700	2,000	13,400	7,100	1,800

Table 25: Computational results to evaluate the impact of the polymatroid cuts on the basic formulations for the assortment data set [85]. For each combination of n, m, κ and each formulation, we present averages over five instances for: time (`Time`) in seconds, number of nodes (`Nodes`) processed, end gap (`End-gap`), continuous relaxation gap (`Rlx-gap`), and root-node gap (`Ron-gap`). For each choice of n, m, and κ, among the solution methods, the best average time and the best average `End-gap` (if `Time`$\geqslant$ 3600 sec.) are in **bold**.

	κ	$10\% \cdot n$					$20\% \cdot n$					Unconstrained				
n, m	Ref.	Time	Nodes	End-gap	Rlx-gap	Ron-gap	Time	Nodes	End-gap	Rlx-gap	Ron-gap	Time	Nodes	End-gap	Rlx-gap	Ron-gap
25,2	LF	**0**	30	0.0%	14.1%	4.7%	2	1,395	0.0%	42.6%	24.5%	2	1,538	0.0%	35.0%	12.1%
	LF^P	**0**	0	0.0%	0.4%	0.3%	**0**	1	0.0%	0.9%	0.7%	**0**	4	0.0%	1.8%	0.0%
	LEF	**0**	2	0.0%	0.7%	0.7%	**0**	9	0.0%	0.9%	0.8%	**0**	0	0.0%	0.0%	0.0%
	LEF^P	**0**	0	0.0%	0.2%	0.1%	**0**	1	0.0%	0.4%	0.2%	**0**	0	0.0%	0.0%	0.0%
	CF	1	13	0.0%	1.5%	1.5%	1	70	0.0%	2.8%	2.8%	2	1,278	0.0%	8.2%	8.2%
	CF^P	**0**	0	0.0%	1.5%	0.8%	**0**	0	0.0%	2.8%	1.9%	1	0	0.0%	8.2%	1.2%
	CEF	**0**	0	0.0%	0.0%	0.0%	**0**	1	0.0%	0.1%	0.1%	**0**	0	0.0%	0.0%	0.0%
	CEF^P	**0**	0	0.0%	0.0%	0.0%	**0**	1	0.0%	0.1%	0.1%	**0**	0	0.0%	0.0%	0.0%
50,5	LF	55	40,100	0.0%	51.7%	41.3%	3600	935,667	29.4%	60.1%	49.8%	3600	1,495,669	19.3%	52.9%	41.4%
	LF^P	**0**	1	0.0%	0.9%	0.1%	**0**	17	0.0%	2.3%	0.1%	4	3,492	0.0%	5.9%	0.1%
	LEF	**0**	123	0.0%	3.0%	3.0%	1	1,111	0.0%	3.2%	3.2%	**0**	2	0.0%	0.0%	0.0%
	LEF^P	**0**	1	0.0%	0.7%	0.0%	**0**	10	0.0%	1.8%	0.2%	**0**	2	0.0%	0.0%	0.0%
	CF	2	157	0.0%	2.3%	2.2%	7	3,016	0.0%	5.5%	5.5%	2916	802,844	1.0%	13.4%	13.4%
	CF^P	1	0	0.0%	2.3%	0.1%	2	0	0.0%	5.5%	0.0%	4	4	0.0%	13.4%	0.9%
	CEF	1	1	0.0%	0.0%	0.0%	1	7	0.0%	0.2%	0.2%	1	2	0.0%	0.0%	0.0%
	CEF^P	1	0	0.0%	0.0%	0.0%	1	8	0.0%	0.2%	0.2%	**0**	2	0.0%	0.0%	0.0%
100,10	LF	3600	170,481	70.1%	78.3%	77.7%	3600	473,158	61.7%	72.1%	70.2%	3600	730,892	51.5%	67.5%	64.2%
	LF^P	4	152	0.0%	1.9%	0.1%	10	1,761	0.0%	4.4%	0.1%	2884	410,429	1.1%	12.0%	1.2%
	LEF	3357	345,641	1.6%	8.3%	8.3%	2190	361,599	0.2%	5.0%	5.0%	**1**	35	0.0%	0.1%	0.0%
	LEF^P	**2**	55	0.0%	1.8%	0.1%	6	792	0.0%	3.5%	1.9%	**1**	28	0.0%	0.1%	0.0%
	CF	67	7,786	0.0%	3.8%	3.6%	3371	231,729	3.5%	7.8%	8.6%	3600	94,567	11.9%	17.8%	17.9%
	CF^P	10	0	0.0%	3.8%	0.0%	20	0	0.0%	7.8%	0.0%	25	370	0.0%	17.8%	0.1%
	CEF	6	14	0.0%	0.7%	0.2%	**4**	23	0.0%	0.6%	0.2%	6	17	0.0%	0.3%	0.0%
	CEF^P	6	14	0.0%	1.6%	0.2%	**4**	25	0.0%	0.4%	0.2%	5	16	0.0%	0.3%	0.0%
200,20	LF	3600	28,528	82.3%	84.4%	84.4%	3600	47,569	77.0%	79.4%	79.3%	3600	49,188	73.4%	78.3%	77.9%
	LF^P	1755	27,150	0.0%	4.1%	0.2%	3600	87,785	0.3%	10.9%	0.4%	3600	37,866	4.5%	22.1%	4.4%
	LEF	3600	42,413	8.6%	10.6%	10.6%	3600	129,049	1.1%	3.0%	3.0%	**29**	1,327	0.0%	0.3%	0.3%
	LEF^P	1236	22,636	0.2%	3.8%	1.1%	3524	152,434	1.0%	3.0%	2.9%	31	1,309	0.0%	0.3%	0.3%
	CF	3600	55,709	32.9%	5.6%	36.0%	3600	29,821	62.3%	13.1%	63.7%	3600	14,939	65.2%	23.6%	73.1%
	CF^P	**27**	5	0.0%	5.6%	0.0%	64	131	0.0%	13.1%	0.0%	1562	26,768	0.2%	23.6%	0.5%
	CEF	73	190	0.0%	1.5%	0.4%	40	31	0.0%	1.5%	0.1%	59	332	0.0%	1.1%	0.1%
	CEF^P	61	230	0.0%	1.6%	0.4%	**38**	37	0.0%	1.3%	0.1%	69	407	0.0%	2.7%	0.1%
500,50	LF	3600	1,620	90.3%	89.0%	89.0%	3600	2,097	86.7%	86.2%	86.2%	3601	5,755	86.2%	86.4%	86.4%
	LF^P	3600	†	†	†	†	3600	†	13.1%	24.8%	12.0%	3600	†	100.0%	33.4%	100.0%
	LEF	3600	2,548	8.3%	8.7%	8.7%	**2520**	8,118	0.2%	0.8%	0.2%	3501	12,727	**0.4%**	1.8%	0.4%
	LEF^P	3600	884	6.9%	7.2%	7.2%	2554	9,193	0.2%	0.8%	0.2%	3552	13,582	**0.4%**	1.8%	0.4%
	CF	3600	11,986	96.4%	8.9%	100.0%	3600	11,367	100.0%	22.2%	100.0%	3600	4,110	96.1%	26.7%	100.0%
	CF^P	**1194**	311	0.0%	8.9%	0.1%	3452	707	0.3%	22.2%	0.7%	3600	440	7.7%	26.7%	0.5%
	CEF	3611	779	0.2%	32.6%	0.3%	2620	842	0.0%	38.4%	0.5%	3604	272	0.5%	23.9%	0.7%
	CEF^P	3609	534	0.2%	29.0%	0.3%	2778	648	0.0%	42.9%	0.5%	3613	160	0.6%	30.9%	0.7%
1000,100	LF	3601	3	99.2%	93.1%	93.1%	3600	10	99.0%	90.4%	90.4%	3601	33	99.0%	90.5%	90.5%
	LF^P	3600	†	†	†	†	3600	†	†	†	†	3600	†	†	†	†
	LEF	3600	215	**13.9%**	4.4%	4.4%	3722	488	**0.9%**	1.2%	0.0%	3600	351	**1.7%**	1.7%	1.1%
	LEF^P	3600	40	81.0%	4.4%	4.4%	3600	1	**0.9%**	1.2%	0.0%	3600	3	1.8%	1.7%	1.1%
	CF	3600	4,962	100.0%	15.7%	100.0%	3605	4,612	100.0%	29.7%	100.0%	3600	1,882	100.0%	30.2%	100.0%
	CF^P	3600	†	†	†	†	3600	†	†	†	†	3600	†	†	†	†
	CEF	3605	†	†	†	†	3600	†	†	†	†	3600	†	†	†	†
	CEF^P	3600	†	†	†	†	3600	†	†	†	†	3600	†	†	†	†
2000,100	LEF	3601	†	†	†	†	3600	†	†	†	†	3601	†	†	†	†
	LEF^P	3600	†	†	†	†	3600	†	†	†	†	3600	†	†	†	†
5000,100	LEF	7807	†	†	†	†	8155	†	†	†	†	7241	†	†	†	†
	LEF^P	3600	†	†	†	†	3600	†	†	†	†	3600	†	†	†	†
10000,100	LEF	4225	†	†	†	†	4026	†	†	†	†	3603	†	†	†	†
	LEF^P	3600	†	†	†	†	3600	†	†	†	†	3600	†	†	†	†

Table 26: Computational results to evaluate the impact of the polymatroid cuts in the basic formulations for the uniformly generated data set [16]. For each combination of n, m, κ and each formulation, we present averages over five instances for: time (`Time`) in seconds, number of nodes (`Nodes`) processed, end gap (`End-gap`), continuous relaxation gap (`Rlx-gap`), and root-node gap (`Ron-gap`). For each choice of n, m, and κ, among the solution methods, the best average time and the best average `End-gap` (if `Time`$\geqslant$ 3600 sec.) are in **bold**.

=	κ	$10\% \cdot n$					$20\% \cdot n$					Unconstrained				
n, m	Ref.	`Time`	`Nodes`	`End-gap`	`Rlx-gap`	`Ron-gap`	`Time`	`Nodes`	`End-gap`	`Rlx-gap`	`Ron-gap`	`Time`	`Nodes`	`End-gap`	`Rlx-gap`	`Ron-gap`
25,2	LF	**0**	38	0.0%	81.5%	48.1%	1	433	0.0%	70.4%	41.5%	1	497	0.0%	89.7%	46.9%
	LF^{P}	1	22	0.0%	33.6%	10.2%	2	340	0.0%	46.9%	12.6%	1	275	0.0%	44.7%	12.0%
	LEF	**0**	1	0.0%	6.8%	5.4%	**0**	24	0.0%	10.8%	9.2%	**0**	21	0.0%	28.1%	22.7%
	LEF^{P}	**0**	2	0.0%	6.7%	4.7%	**0**	20	0.0%	10.8%	9.0%	1	17	0.0%	26.6%	15.5%
	CF	1	114	0.0%	31.4%	24.8%	1	1,198	0.0%	40.5%	40.5%	1	1,657	0.0%	37.1%	37.1%
	CF^{P}	3	2	0.0%	31.4%	4.6%	4	179	0.0%	40.5%	6.3%	4	24	0.0%	37.1%	3.4%
	CEF	**0**	2	0.0%	4.0%	3.8%	**0**	14	0.0%	7.9%	7.0%	1	16	0.0%	18.6%	13.2%
	CEF^{P}	**0**	2	0.0%	4.0%	2.8%	**0**	14	0.0%	7.9%	7.0%	2	16	0.0%	18.6%	5.9%
50,5	LF	1554	589,909	0.0%	85.7%	82.8%	3600	2,006,223	46.9%	75.2%	69.9%	3600	2,791,336	44.9%	96.0%	91.5%
	LF^{P}	437	126,715	0.0%	55.9%	27.2%	3600	1,542,663	20.2%	61.2%	35.5%	2530	826,742	4.2%	59.6%	26.5%
	LEF	2	381	0.0%	18.5%	16.4%	13	9,831	0.0%	20.9%	19.9%	**43**	35,334	0.0%	56.3%	49.6%
	LEF^{P}	**1**	327	0.0%	18.5%	16.2%	**10**	10,679	0.0%	20.9%	19.9%	65	34,778	0.0%	50.1%	28.3%
	CF	75	31,692	0.0%	56.2%	56.2%	3606	1,112,962	6.0%	56.3%	56.3%	2594	677,850	7.3%	56.6%	56.6%
	CF^{P}	78	22,068	0.0%	56.2%	23.9%	3601	1,058,360	6.5%	56.3%	25.4%	2903	1,043,778	3.0%	56.6%	23.9%
	CEF	3	172	0.0%	14.6%	13.7%	18	6,093	0.0%	15.6%	15.3%	100	35,410	0.0%	43.6%	40.9%
	CEF^{P}	2	162	0.0%	14.6%	13.7%	17	6,266	0.0%	15.6%	15.3%	311	32,574	0.0%	40.6%	24.1%
100,10	LF	3600	1,100,713	82.1%	87.5%	87.1%	3600	1,252,405	71.1%	77.1%	76.0%	3600	1,308,606	90.2%	98.3%	96.7%
	LF^{P}	3600	303,009	51.7%	74.3%	56.5%	3600	337,167	51.4%	73.5%	54.7%	3600	153,764	42.5%	73.9%	49.7%
	LEF	3600	480,988	12.3%	29.4%	27.5%	3600	616,551	17.1%	30.4%	29.7%	3600	654,126	38.5%	72.5%	66.6%
	LEF^{P}	3600	526,364	12.3%	29.4%	27.4%	3600	687,839	17.0%	30.4%	29.7%	3600	438,734	**37.2%**	66.9%	51.5%
	CF	3600	223,083	52.3%	72.8%	72.8%	3600	154,439	54.9%	66.9%	66.9%	3600	220,110	53.2%	71.5%	71.4%
	CF^{P}	3600	462,737	43.5%	72.8%	50.8%	3600	166,635	44.3%	66.9%	47.3%	3600	330,256	42.0%	71.5%	50.4%
	CEF	3600	221,990	**10.7%**	25.6%	24.8%	3600	275,594	15.5%	25.0%	25.7%	3600	130,787	40.1%	63.7%	61.1%
	CEF^{P}	3601	204,084	**10.7%**	25.7%	24.8%	3600	260,464	**15.4%**	25.2%	25.7%	3600	132,248	40.3%	60.5%	50.8%
200,20	LF	3600	118,471	87.8%	88.6%	88.6%	3600	107,640	77.4%	78.1%	77.9%	3600	184,061	97.5%	99.2%	98.5%
	LF^{P}	3600	17,965	72.1%	82.0%	71.2%	3600	18,122	65.5%	77.0%	65.0%	3600	9,401	73.1%	81.8%	70.4%
	LEF	3600	47,486	30.0%	36.2%	35.4%	3600	58,945	31.1%	36.0%	35.6%	3600	63,610	70.6%	83.0%	78.8%
	LEF^{P}	3600	74,821	**29.3%**	36.2%	35.3%	3600	100,973	30.6%	36.0%	35.6%	3600	37,575	**65.4%**	77.8%	67.3%
	CF	3600	27,323	99.5%	80.4%	99.5%	3600	17,547	89.6%	72.0%	88.7%	3600	24,168	99.7%	81.0%	100.0%
	CF^{P}	3600	25,375	65.8%	80.4%	66.0%	3600	1,113	61.6%	72.0%	58.9%	3600	7,872	70.9%	81.0%	66.3%
	CEF	3600	20,677	30.9%	30.9%	33.3%	3600	22,387	**30.0%**	23.0%	32.7%	3600	4,559	76.4%	76.0%	75.8%
	CEF^{P}	3600	17,843	30.8%	32.3%	33.3%	3600	19,082	30.1%	25.5%	32.7%	3601	5,142	71.5%	72.7%	67.8%
500,50	LF	3600	1,232	90.2%	89.4%	89.4%	3600	369	82.3%	78.3%	78.3%	3600	6,619	99.5%	99.8%	99.5%
	LF^{P}	3600	606	88.4%	87.1%	87.1%	3600	395	81.5%	78.0%	77.4%	3600	†	†	†	†
	LEF	3600	636	42.8%	43.0%	42.7%	3600	1,324	41.1%	38.6%	38.5%	3600	113	90.3%	91.4%	89.3%
	LEF^{P}	3600	850	**42.5%**	43.0%	42.7%	3600	1,616	**41.0%**	38.6%	38.5%	3600	39	**83.1%**	88.0%	82.8%
	CF	3600	9,997	100.0%	86.1%	100.0%	3602	3,292	100.0%	75.0%	100.0%	3600	13,213	100.0%	89.8%	100.0%
	CF^{P}	3600	†	†	†	†	3600	†	†	†	†	3600	1,208	84.9%	89.8%	83.1%
	CEF	3603	17	42.8%	20.4%	41.3%	3604	26	41.8%	5.7%	37.3%	3603	1	93.4%	80.5%	90.9%
	CEF^{P}	3603	7	42.9%	23.3%	41.3%	3604	11	53.6%	7.2%	37.3%	3600	†	†	†	†
1000,100	LF	3601	†	†	†	†	3600	21	81.6%	79.9%	79.9%	3601	4	99.9%	99.9%	99.8%
	LF^{P}	3600	†	†	†	†	3600	†	†	†	†	3600	†	†	†	†
	LEF	3601	†	†	†	†	3601	†	†	†	†	3601	†	†	†	†
	LEF^{P}	3600	†	†	†	†	3600	†	†	†	†	3600	†	†	†	†
	CF	3600	911	**100.0%**	87.3%	100.0%	3600	1,210	**100.0%**	77.8%	100.0%	3600	663	100.0%	92.8%	100.0%
	CF^{P}	3600	†	†	†	†	3600	†	†	†	†	3600	406	**95.6%**	83.4%	95.5%
	CEF	3600	†	†	†	†	3600	†	†	†	†	3600	†	†	†	†
	CEF^{P}	3600	†	†	†	†	3600	†	†	†	†	3600	†	†	†	†
2000,100	LEF	3601	†	†	†	†	3602	†	†	†	†	3601	†	†	†	†
	LEF^{P}	3600	†	†	†	†	3600	†	†	†	†	3600	†	†	†	†
5000,100	LEF	4755	†	†	†	†	3938	†	†	†	†	3603	†	†	†	†
	LEF^{P}	3600	†	†	†	†	3600	†	†	†	†	3600	†	†	†	†
10000,100	LEF	9500	†	†	†	†	6022	†	†	†	†	5619	†	†	†	†
	LEF^{P}	3600	†	†	†	†	3600	†	†	†	†	3600	†	†	†	†

Table 27: Computational results to evaluate the combined effect of binarization and polymatroid cuts on the performance of selected basic MILP and MICQP for the assortment data set [85]. For each combination of n, m, κ and each formulation, we present averages over five instances for: time (`Time`) in seconds, number of nodes (`Nodes`) processed, end gap (`End-gap`), continuous relaxation gap (`Rlx-gap`), and root-node gap (`Ron-gap`). For each choice of n, m, and κ, among the solution methods, the best average time and the best average `End-gap` (if `Time`$\geqslant$ 3600 sec.) are in **bold**.

	κ	$10\% \cdot n$					$20\% \cdot n$					Unconstrained				
n, m	Ref.	Time	Nodes	End-gap	Rlx-gap	Ron-gap	Time	Nodes	End-gap	Rlx-gap	Ron-gap	Time	Nodes	End-gap	Rlx-gap	Ron-gap
25,2	LF	**0**	30	0.0%	14.1%	4.7%	2	1,395	0.0%	42.6%	24.5%	2	1,538	0.0%	35.0%	12.1%
	LF^P	**0**	0	0.0%	0.4%	0.3%	**0**	1	0.0%	0.9%	0.7%	**0**	4	0.0%	1.8%	0.0%
	LF_{log}	**0**	5	0.0%	15.3%	1.1%	**0**	27	0.0%	28.1%	4.3%	1	386	0.0%	56.1%	18.1%
	LF^P_{log}	**0**	0	0.0%	0.5%	0.3%	**0**	1	0.0%	0.9%	0.7%	**0**	20	0.0%	1.8%	0.1%
	CEF	**0**	0	0.0%	0.0%	0.0%	**0**	1	0.0%	0.1%	0.1%	**0**	0	0.0%	0.0%	0.0%
	CEF^P	**0**	0	0.0%	0.0%	0.0%	**0**	1	0.0%	0.1%	0.1%	**0**	0	0.0%	0.0%	0.0%
	CEF_{log}	1	1	0.0%	0.1%	0.1%	1	3	0.0%	0.7%	0.3%	1	316	0.0%	5.8%	5.8%
	CEF^P_{log}	1	1	0.0%	0.1%	0.1%	**0**	0	0.0%	0.6%	0.1%	2	379	0.0%	1.8%	1.5%
50,5	LF	55	40,100	0.0%	51.7%	41.3%	3600	935,667	29.4%	60.1%	49.8%	3600	1,495,669	19.3%	52.9%	41.4%
	LF^P	**0**	1	0.0%	0.9%	0.1%	**0**	17	0.0%	2.3%	0.1%	4	3,492	0.0%	5.9%	0.1%
	LF_{log}	1	233	0.0%	30.0%	5.7%	2	2,109	0.0%	44.4%	14.5%	18	35,496	0.0%	65.6%	28.2%
	LF^P_{log}	**0**	0	0.0%	0.9%	0.3%	1	19	0.0%	2.4%	0.1%	6	6,721	0.0%	5.9%	0.3%
	CEF	1	1	0.0%	0.0%	0.0%	1	7	0.0%	0.2%	0.2%	1	2	0.0%	0.0%	0.0%
	CEF^P	1	0	0.0%	0.0%	0.0%	1	8	0.0%	0.2%	0.2%	**0**	2	0.0%	0.0%	0.0%
	CEF_{log}	1	6	0.0%	0.6%	0.3%	2	371	0.0%	2.1%	2.0%	18	13,917	0.0%	12.4%	12.3%
	CEF^P_{log}	**0**	4	0.0%	0.5%	0.3%	2	250	0.0%	1.9%	0.9%	21	11,132	0.0%	5.9%	5.2%
100,10	LF	3600	170,481	70.1%	78.3%	77.7%	3600	473,158	61.7%	72.1%	70.2%	3600	730,892	51.5%	67.5%	64.2%
	LF^P	4	152	0.0%	1.9%	0.1%	10	1,761	0.0%	4.4%	0.1%	2884	410,429	1.1%	12.0%	1.2%
	LF_{log}	979	364,141	0.0%	42.7%	14.5%	3155	1,732,777	0.4%	55.3%	23.7%	3600	1,543,428	1.6%	75.9%	38.5%
	LF^P_{log}	**1**	86	0.0%	1.9%	0.1%	6	2,434	0.0%	4.4%	0.1%	3600	1,535,465	0.8%	12.1%	1.5%
	CEF	6	14	0.0%	0.7%	0.2%	**4**	23	0.0%	0.6%	0.2%	6	17	0.0%	0.3%	0.0%
	CEF^P	6	14	0.0%	1.6%	0.2%	**4**	25	0.0%	0.4%	0.2%	**5**	16	0.0%	0.3%	0.0%
	CEF_{log}	10	1,457	0.0%	1.3%	1.2%	212	27,571	0.0%	4.0%	3.9%	3465	292,906	1.0%	20.2%	20.2%
	CEF^P_{log}	2	215	0.0%	1.2%	0.3%	22	3,199	0.0%	3.7%	2.1%	3600	411,139	0.3%	12.1%	8.4%
200,20	LF	3600	28,528	82.3%	84.4%	84.4%	3600	47,569	77.0%	79.4%	79.3%	3600	49,188	73.4%	78.3%	77.9%
	LF^P	1755	27,150	0.0%	4.1%	0.2%	3600	87,785	0.3%	10.9%	0.4%	3600	37,866	4.5%	22.1%	4.4%
	LF_{log}	3600	549,079	6.7%	52.9%	24.7%	3600	383,827	8.7%	64.7%	31.9%	3600	300,111	24.1%	82.7%	49.9%
	LF^P_{log}	710	158,569	0.0%	4.1%	0.2%	3400	715,941	0.3%	10.9%	0.4%	3600	374,382	6.3%	22.3%	6.0%
	CEF	73	190	0.0%	1.5%	0.4%	40	31	0.0%	1.5%	0.1%	**59**	332	0.0%	1.1%	0.1%
	CEF^P	**61**	230	0.0%	1.6%	0.4%	**38**	37	0.0%	1.3%	0.1%	69	407	0.0%	2.7%	0.1%
	CEF_{log}	3600	137,672	0.9%	2.6%	2.6%	3600	121,652	3.8%	8.2%	8.2%	3600	96,988	22.0%	28.9%	29.0%
	CEF^P_{log}	2353	74,047	0.5%	2.6%	2.2%	3600	112,151	2.2%	8.2%	6.8%	3600	144,453	6.4%	22.2%	13.8%
500,50	LF	3600	1,620	90.3%	89.0%	89.0%	3600	2,097	86.7%	86.2%	86.2%	3601	5,755	86.2%	86.4%	86.4%
	LF^P	3600	†	†	†	†	3600	†	†	†	†	3600	†	†	†	†
	LF_{log}	3600	102,004	39.8%	53.0%	35.4%	3600	84,392	54.0%	68.6%	35.0%	3600	92,414	55.7%	91.8%	74.7%
	LF^P_{log}	3600	110,452	0.8%	8.8%	0.3%	3600	57,797	3.3%	24.8%	1.3%	3600	65,850	15.2%	33.6%	13.4%
	CEF	3611	779	**0.2%**	32.6%	0.3%	**2620**	842	0.0%	38.4%	0.5%	3604	272	**0.5%**	23.9%	0.7%
	CEF^P	3609	534	**0.2%**	29.0%	0.3%	2778	648	0.0%	42.9%	0.5%	3613	160	0.6%	30.9%	0.7%
	CEF_{log}	3600	49,090	5.1%	5.4%	5.4%	3600	55,757	13.3%	15.8%	15.7%	3600	53,280	36.9%	37.3%	38.5%
	CEF^P_{log}	3600	55,687	4.7%	5.4%	5.0%	3600	63,450	12.2%	15.8%	14.0%	3601	129,520	26.1%	33.5%	20.6%
1000,100	LF	3601	3	99.2%	93.1%	93.1%	3600	10	99.0%	90.4%	90.4%	3601	33	99.0%	90.5%	90.5%
	LF^P	3600	†	†	†	†	3600	†	†	†	†	3600	†	†	†	†
	LF_{log}	3600	55,776	55.9%	60.6%	51.0%	3600	58,847	62.7%	77.4%	61.2%	3600	55,641	76.5%	94.3%	79.1%
	LF^P_{log}	3601	†	†	†	†	3601	6,378	**20.9%**	39.4%	15.8%	3601	30,129	**26.1%**	43.6%	23.9%
	CEF	3605	†	†	†	†	3600	†	†	†	†	3600	†	†	†	†
	CEF^P	3600	†	†	†	†	3600	†	†	†	†	3600	†	†	†	†
	CEF_{log}	3600	36,151	25.3%	9.8%	9.9%	3600	36,647	23.7%	23.6%	23.9%	3600	35,213	48.7%	44.6%	45.7%
	CEF^P_{log}	3601	32,326	**10.0%**	9.9%	9.5%	3600	26,843	22.6%	23.9%	22.5%	3600	4,283	33.8%	42.9%	25.0%
2000,100	LF_{log}	3600	58,217	57.8%	68.0%	62.7%	3600	56,546	70.5%	84.0%	79.1%	3600	39,585	78.3%	96.0%	81.6%
	LF^P_{log}	3601	†	†	†	†	3600	32,898	41.4%	48.9%	41.2%	3601	8,660	**33.1%**	52.2%	31.2%
	CEF_{log}	3600	26,386	30.3%	15.5%	16.1%	3600	22,548	60.0%	31.9%	38.2%	3600	26,785	71.0%	50.7%	52.7%
	CEF^P_{log}	3600	38,716	**16.1%**	15.8%	15.4%	3600	28,575	**30.7%**	32.5%	31.9%	3600	931	53.4%	48.2%	33.0%
5000,100	LF_{log}	3600	23,558	78.1%	86.8%	84.7%	3600	37,298	80.6%	93.4%	93.4%	3601	12,870	83.5%	96.8%	96.8%
	LF^P_{log}	3601	7,220	**29.2%**	50.1%	25.1%	3601	15,186	49.0%	59.9%	47.6%	3601	6,818	**50.7%**	61.1%	49.3%
	CEF_{log}	3600	15,535	48.0%	26.7%	57.7%	3600	10,662	77.7%	39.3%	60.0%	3600	11,067	86.5%	57.2%	86.5%
	CEF^P_{log}	3600	13,966	39.3%	27.5%	30.7%	3600	13,736	**40.6%**	33.9%	40.1%	3600	3,257	58.4%	50.8%	47.5%
10000,100	LF_{log}	3600	13,230	88.4%	90.0%	90.0%	3600	8,857	83.1%	94.7%	94.7%	3602	5,082	93.0%	97.6%	97.6%
	LF^P_{log}	3601	5,481	55.4%	58.6%	54.7%	3601	9,440	53.2%	61.4%	49.2%	3601	5,482	**54.7%**	65.0%	54.3%
	CEF_{log}	3600	7,551	53.8%	29.5%	52.2%	3600	3,781	84.6%	45.0%	85.1%	3600	2,786	95.4%	70.3%	95.0%
	CEF^P_{log}	3600	9,979	**33.4%**	5.0%	34.9%	3601	7,247	**45.4%**	22.0%	37.6%	3601	†	†	†	†

Table 28: Computational results to evaluate the combined effect of binarization and polymatroid cuts on the performance of selected basic MILP and MICQP formulations for the uniformly generated data set [16]. For each combination of n, m, κ and each formulation, we present averages over five instances for: time (`Time`) in seconds, number of nodes (`Nodes`) processed, end gap (`End-gap`), continuous relaxation gap (`Rlx-gap`), and root-node gap (`Ron-gap`). For each choice of n, m, and κ, among the solution methods, the best average time and the best average `End-gap` (if `Time`$\geqslant$ 3600 sec.) are in **bold**.

	κ	$10\% \cdot n$					$20\% \cdot n$					Unconstrained				
n, m	Ref.	Time	Nodes	End-gap	Rlx-gap	Ron-gap	Time	Nodes	End-gap	Rlx-gap	Ron-gap	Time	Nodes	End-gap	Rlx-gap	Ron-gap
25,2	LF	**0**	38	0.0%	81.5%	48.1%	1	433	0.0%	70.4%	41.5%	**1**	497	0.0%	89.7%	46.9%
	LF^{P}	1	22	0.0%	33.6%	10.2%	2	340	0.0%	46.9%	12.6%	**1**	275	0.0%	44.7%	12.0%
	LF_{log}	**0**	23	0.0%	49.8%	28.0%	1	95	0.0%	50.1%	32.3%	**1**	199	0.0%	93.0%	59.8%
	LF^{P}_{log}	**0**	13	0.0%	29.5%	11.5%	1	222	0.0%	42.3%	12.0%	**1**	142	0.0%	45.5%	14.6%
	CEF	**0**	2	0.0%	4.0%	3.8%	**0**	14	0.0%	7.9%	7.0%	**1**	16	0.0%	18.6%	13.2%
	CEF^{P}	**0**	2	0.0%	4.0%	2.8%	**0**	14	0.0%	7.9%	7.0%	2	16	0.0%	18.6%	5.9%
	CEF_{log}	**0**	6	0.0%	10.3%	7.9%	1	89	0.0%	19.9%	19.3%	3	353	0.0%	45.7%	44.2%
	CEF^{P}_{log}	1	4	0.0%	10.3%	6.9%	1	106	0.0%	19.9%	18.7%	6	133	0.0%	42.8%	27.1%
50,5	LF	1554	589,909	0.0%	85.7%	82.8%	3600	2,006,223	46.9%	75.2%	69.9%	3600	2,791,336	44.9%	96.0%	91.5%
	LF^{P}	437	126,715	0.0%	55.9%	27.2%	3600	1,542,663	20.2%	61.2%	35.5%	2530	826,742	4.2%	59.6%	26.5%
	LF_{log}	3	3,364	0.0%	50.7%	43.7%	20	21,061	0.0%	54.2%	45.0%	**52**	55,437	0.0%	96.9%	77.1%
	LF^{P}_{log}	9	5,014	0.0%	45.9%	21.3%	27	29,157	0.0%	51.4%	30.0%	85	81,310	0.0%	60.3%	25.0%
	CEF	3	172	0.0%	14.6%	13.7%	18	6,093	0.0%	15.6%	15.3%	100	35,410	0.0%	43.6%	40.9%
	CEF^{P}	**2**	162	0.0%	14.6%	13.7%	**17**	6,266	0.0%	15.6%	15.3%	311	32,574	0.0%	40.6%	24.1%
	CEF_{log}	11	4,746	0.0%	24.4%	24.0%	22	8,046	0.0%	30.0%	29.6%	521	78,437	0.0%	64.4%	62.1%
	CEF^{P}_{log}	6	2,477	0.0%	24.4%	22.6%	26	8,630	0.0%	29.9%	29.1%	86	25,435	0.0%	59.4%	46.4%
100,10	LF	3600	1,100,713	82.1%	87.5%	87.1%	3600	1,252,405	71.1%	77.1%	76.0%	3600	1,308,606	90.2%	98.3%	96.7%
	LF^{P}	3600	303,009	51.7%	74.3%	56.5%	3600	337,167	51.4%	73.5%	54.7%	3600	153,764	42.5%	73.9%	49.7%
	LF_{log}	3600	2,079,337	**5.0%**	54.5%	48.7%	3600	2,153,102	**5.0%**	56.4%	49.8%	3600	2,487,103	11.2%	98.6%	84.6%
	LF^{P}_{log}	3600	2,588,756	7.5%	54.1%	45.5%	3600	2,821,692	6.1%	56.3%	48.3%	3600	1,928,384	17.2%	74.5%	48.6%
	CEF	3600	221,990	10.7%	25.6%	24.8%	3600	275,594	15.5%	25.0%	25.7%	3600	130,787	40.1%	63.7%	61.1%
	CEF^{P}	3601	204,084	10.7%	25.7%	24.8%	3600	260,464	15.4%	25.2%	25.7%	3600	132,248	40.3%	60.5%	50.8%
	CEF_{log}	3601	433,421	8.1%	34.8%	34.7%	3600	394,433	7.9%	36.6%	36.5%	3600	368,512	20.1%	76.0%	74.6%
	CEF^{P}_{log}	3600	482,188	7.2%	34.8%	34.5%	3603	463,914	5.2%	36.6%	36.6%	3600	417,221	**10.9%**	73.6%	61.2%
200,20	LF	3600	118,471	87.8%	88.6%	88.6%	3600	107,640	77.4%	78.1%	77.9%	3600	184,061	97.5%	99.2%	98.5%
	LF^{P}	3600	17,965	72.1%	82.0%	71.2%	3600	18,122	65.5%	77.0%	65.0%	3600	9,401	73.1%	81.8%	70.4%
	LF_{log}	3600	612,063	41.7%	56.8%	54.5%	3600	490,278	37.7%	58.0%	54.7%	3600	519,981	58.2%	99.3%	89.9%
	LF^{P}_{log}	3600	1,104,491	41.6%	56.7%	53.5%	3600	938,882	35.6%	57.9%	54.5%	3600	434,136	58.0%	82.1%	65.9%
	CEF	3600	20,677	30.9%	30.9%	33.3%	3600	22,387	**30.0%**	23.0%	32.7%	3600	4,559	76.4%	76.0%	75.8%
	CEF^{P}	3600	17,843	**30.8%**	32.3%	33.3%	3600	19,082	30.1%	25.5%	32.7%	3601	5,142	71.5%	72.7%	67.8%
	CEF_{log}	3600	131,182	39.6%	40.1%	40.1%	3600	88,037	36.6%	40.0%	40.0%	3600	285,525	64.6%	83.6%	83.3%
	CEF^{P}_{log}	3600	174,404	35.5%	40.1%	39.9%	3600	113,509	34.3%	40.0%	39.9%	3600	279,263	**54.4%**	81.4%	73.4%
500,50	LF	3600	1,232	90.2%	89.4%	89.4%	3600	369	82.3%	78.3%	78.3%	3600	6,619	99.5%	99.8%	99.5%
	LF^{P}	3600	606	88.4%	87.1%	87.1%	3600	395	81.5%	78.0%	77.4%	3600	†	†	†	†
	LF_{log}	3600	81,055	48.7%	49.0%	48.9%	3600	60,815	48.7%	47.2%	47.1%	3600	139,697	87.0%	99.9%	96.1%
	LF^{P}_{log}	3600	108,291	48.4%	49.0%	49.0%	3600	70,193	48.1%	47.2%	47.1%	3600	181,247	**82.9%**	90.5%	85.9%
	CEF	3603	17	**42.8%**	20.4%	41.3%	3604	26	**41.8%**	5.7%	37.3%	3603	1	93.4%	80.5%	90.9%
	CEF^{P}	3603	7	42.9%	23.3%	41.3%	3604	11	53.6%	7.2%	37.3%	3600	†	†	†	†
	CEF_{log}	3600	34,703	53.2%	45.2%	45.1%	3600	26,390	42.8%	40.7%	40.7%	3600	82,878	91.0%	90.8%	90.8%
	CEF^{P}_{log}	3600	29,818	46.3%	45.2%	44.9%	3600	24,696	43.1%	40.7%	40.7%	3600	23,870	86.7%	89.7%	86.6%
1000,100	LF	3601	†	†	†	†	3600	21	81.6%	79.9%	79.9%	3601	4	99.9%	99.9%	99.8%
	LF^{P}	3600	†	†	†	†	3600	†	†	†	†	3600	†	†	†	†
	LF_{log}	3600	52,994	50.3%	48.7%	48.7%	3600	41,825	50.1%	50.9%	50.8%	3600	48,644	96.6%	99.9%	97.3%
	LF^{P}_{log}	3600	48,719	50.2%	48.7%	48.7%	3600	24,225	50.2%	50.7%	50.8%	3600	108,734	**91.9%**	93.1%	92.0%
	CEF	3600	†	†	†	†	3600	†	†	†	†	3600	†	†	†	†
	CEF^{P}	3600	†	†	†	†	3600	†	†	†	†	3600	†	†	†	†
	CEF_{log}	3600	12,062	**46.0%**	45.3%	45.3%	3601	8,843	47.9%	44.7%	45.3%	3600	37,767	93.7%	93.3%	93.3%
	CEF^{P}_{log}	3600	10,436	48.0%	45.3%	45.3%	3600	9,445	**44.5%**	44.7%	45.0%	3600	476	92.2%	92.5%	90.2%
2000,100	LF_{log}	3600	41,092	50.7%	51.2%	51.2%	3600	30,062	50.6%	50.8%	50.7%	3600	35,408	97.8%	100.0%	98.2%
	LF^{P}_{log}	3600	15,925	50.8%	51.1%	51.2%	3600	14,228	50.7%	50.8%	50.8%	3600	69,565	**94.8%**	95.5%	95.1%
	CEF_{log}	3601	5,139	48.8%	47.9%	48.4%	3600	4,909	48.5%	44.4%	45.2%	3600	25,840	97.0%	95.5%	95.6%
	CEF^{P}_{log}	3600	9,576	**47.8%**	48.3%	48.2%	3600	7,815	**44.6%**	45.1%	45.0%	3600	339	96.6%	95.1%	93.7%
5000,100	LF_{log}	3600	18,499	67.9%	68.6%	68.6%	3600	34,661	65.0%	69.5%	69.5%	3601	13,907	98.8%	100.0%	98.8%
	LF^{P}_{log}	3600	9,434	68.8%	68.6%	68.6%	3600	12,867	67.9%	69.5%	69.2%	3601	16,900	**96.9%**	96.7%	96.5%
	CEF_{log}	3600	5,092	51.4%	46.4%	47.1%	3600	3,305	48.0%	44.6%	45.6%	3600	11,678	97.0%	96.7%	96.7%
	CEF^{P}_{log}	3600	4,295	**46.7%**	47.2%	47.0%	3601	3,406	**45.2%**	45.7%	45.5%	3601	34	98.3%	96.4%	96.0%
10000,100	LF_{log}	3600	15,052	68.6%	69.0%	69.0%	3600	11,855	68.2%	69.2%	69.2%	3601	2,471	99.4%	100.0%	99.3%
	LF^{P}_{log}	3601	5,732	68.5%	69.0%	69.0%	3601	6,058	68.4%	69.2%	68.8%	3601	6,595	**97.8%**	98.0%	97.9%
	CEF_{log}	3600	1,873	50.5%	47.2%	47.7%	3600	1,010	48.2%	44.3%	45.1%	3601	475	99.4%	98.0%	99.0%
	CEF^{P}_{log}	3601	896	**47.5%**	47.9%	47.7%	3600	1,165	**44.8%**	45.1%	45.0%	3600	†	†	†	†

Bibliography

[1] Adams, W. P. and Forrester, R. J. (2005). A simple recipe for concise mixed 0-1 linearizations. *Operations Research Letters*, 33(1):55–61.

[2] Ahuja, R. K., Magnanti, T. L., and Orlin, J. B. (1993). *Network Flows: Theory, Algorithms, and Applications*. Pearson Education.

[3] Amaldi, E., Bosio, S., and Malucelli, F. (2012). Hyperbolic set covering problems with competing ground-set elements. *Mathematical Programming*, 134(2):323–348.

[4] Amaldi, E., Bosio, S., Malucelli, F., and Yuan, D. (2011). Solving nonlinear covering problems arising in wlan design. *Operations Research*, 59(1):173–187.

[5] Asuncion, A. and Newman, D. (2007). UCI machine learning repository.

[6] Atamtürk, A. and Gómez, A. (2018). Submodularity in conic quadratic mixed 0-1 optimization. *arXiv preprint arXiv:1705.05918*.

[7] Atamtürk, A. and Narayanan, V. (2008). Polymatroids and mean-risk minimization in discrete optimization. *Operations Research Letters*, 36(5):618–622.

[8] Ben-Tal, A. and Nemirovski, A. (2001). On polyhedral approximations of the second-order cone. *Mathematics of Operations Research*, 26(2):193–205.

[9] Bertsimas, D., Brown, D. B., and Caramanis, C. (2011). Theory and applications of robust optimization. *SIAM Review*, 53(3):464–501.

[10] Bertsimas, D., Korolko, N., and Weinstein, A. M. (2019). Identifying exceptional responders in randomized trials: An optimization approach. *INFORMS Journal on Optimization*, pages 1–13.

[11] Bertsimas, D. and Mišić, V. V. (2016). Robust product line design. *Operations Research*, 65(1):19–37.

[12] Bertsimas, D. and Sim, M. (2003). Robust discrete optimization and network flows. *Mathematical Programming*, 98(1):49–71.

[13] Bertsimas, D. and Sim, M. (2004). The price of robustness. *Operations Research*, 52(1):35–53.

[14] Beyer, H.-G. and Sendhoff, B. (2007). Robust optimization–a comprehensive survey. *Computer methods in applied mechanics and engineering*, 196(33-34):3190–3218.

[15] Boros, E. and Hammer, P. (2002). Pseudo-boolean optimization. *Discrete Applied Mathematics*, 123(1):155–225.

[16] Borrero, J. S., Gillen, C., and Prokopyev, O. A. (2016). A simple technique to improve linearized reformulations of fractional (hyperbolic) 0–1 programming problems. *Operations Research Letters*, 44(4):479–486.

[17] Borrero, J. S., Gillen, C., and Prokopyev, O. A. (2017). Fractional 0–1 programming: applications and algorithms. *Journal of Global Optimization*, 69(1):255–282.

[18] Boyd, J. H. and Mellman, R. E. (1980). The effect of fuel economy standards on the us automotive market: an hedonic demand analysis. *Transportation Research Part A: General*, 14(5-6):367–378.

[19] Bront, J. J. M., Méndez-Díaz, I., and Vulcano, G. (2009). A column generation algorithm for choice-based network revenue management. *Operations Research*, 57(3):769–784.

[20] Buchheim, C. and Kurtz, J. (2018). Robust combinatorial optimization under convex and discrete cost uncertainty. *EURO Journal on Computational Optimization*, 6(3):211–238.

[21] Busygin, S., Prokopyev, O., and Pardalos, P. M. (2008). Biclustering in data mining. *Computers & Operations Research*, 35(9):2964–2987.

[22] Busygin, S., Prokopyev, O. A., and Pardalos, P. M. (2005). Feature selection for consistent biclustering via fractional 0–1 programming. *Journal of Combinatorial Optimization*, 10(1):7–21.

[23] Chandrashekar, G. and Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28.

[24] Chang, C.-T. (2001). On the polynomial mixed 0–1 fractional programming problems. *European Journal of Operational Research*, 131(1):224–227.

[25] Charnes, A. and Cooper, W. W. (1962). Programming with linear fractional functionals. *Naval Research Logistics*, 9(3-4):181–186.

[26] Cilia, N. D., De Stefano, C., Fontanella, F., and di Freca, A. S. (2019). A ranking-based feature selection approach for handwritten character recognition. *Pattern Recognition Letters*, 121:77–86.

[27] Dasdan, A., Irani, S. S., and Gupta, R. K. (1999). Efficient algorithms for optimum cycle mean and optimum cost to time ratio problems. In *Proceedings of the 36th Annual ACM/IEEE Design Automation Conference*, pages 37–42. ACM.

[28] Davis, J. M., Gallego, G., and Topaloglu, H. (2014). Assortment optimization under variants of the nested logit model. *Operations Research*, 62(2):250–273.

[29] Désir, A., Goyal, V., Jiang, B., Xie, T., and Zhang, J. (2019). A nonconvex min-max framework and its applications to robust assortment optimization. `https://faculty.insead.edu/antoine-desir`, Last accessed May 2019.

[30] Ding, C. and Peng, H. (2005). Minimum redundancy feature selection from microarray gene expression data. *Journal of Bioinformatics and Computational Biology*, 3(02):185–205.

[31] Dinkelbach, W. (1967). On nonlinear fractional programming. *Management Science*, 13(7):492–498.

[32] Edmonds, J. (1970). Submodular functions, matroids, and certain polyhedra. *Combinatorial Structures and Their Applications*, pages 69–87.

[33] Elhedhli, S. (2005). Exact solution of a class of nonlinear knapsack problems. *Operations Research Letters*, 33(6):615–624.

[34] Fan, Y.-J. and Chaovalitwongse, W. A. (2010). Optimizing feature selection to improve medical diagnosis. *Annals of Operations Research*, 174(1):169–183.

[35] Feldman, J. and Topaloglu, H. (2015). Bounding optimal expected revenues for assortment optimization under mixtures of multinomial logits. *Production and Operations Management*, 24(10):1598–1620.

[36] Glover, F. and Woolsey, E. (1974). Converting the 0-1 polynomial programming problem to a 0-1 linear program. *Operations research*, 22(1):180–182.

[37] Gómez, A. and Prokopyev, O. A. (2018). A mixed-integer fractional optimization approach to best subset selection. `http://www.optimization-online.org/DB_HTML/2018/09/6795.html`, Last accessed August 2019.

[38] Gorissen, B. L. (2015). Robust fractional programming. *Journal of Optimization Theory and Applications*, 166(2):508–528.

[39] Gurobi (2018). Gurobi optimizer reference manual v. 8. `http://www.gurobi.com`.

[40] Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182.

[41] Hall, M. A. (1999). *Correlation-based feature selection for machine learning*. PhD thesis, University of Waikato Hamilton.

[42] Han, J., Lee, K., Lee, C., and Park, S. (2013). Exact algorithms for a bandwidth packing problem with queueing delay guarantees. *INFORMS Journal on Computing*, 25(3):585–596.

[43] Hansen, P., de Aragão, M. V. P., and Ribeiro, C. C. (1991). Hyperbolic 0-1 programming and query optimization in information retrieval. *Mathematical Programming*, 52(1-3):255–263.

[44] Hansen, P., Poggi de Aragão, M. V., and Ribeiro, C. C. (1990). Boolean query optimization and the 0-1 hyperbolic sum problem. *Annals of Mathematics and Artificial Intelligence*, 1(1):97–109.

[45] Huang, H., Xie, H.-B., Guo, J.-Y., and Chen, H.-J. (2012). Ant colony optimization-based feature selection method for surface electromyography signals classification. *Computers in Biology and Medicine*, 42(1):30–38.

[46] Ibaraki, T. (1983). Parametric approaches to fractional programs. *Mathematical Programming*, 26(3):345–362.

[47] IBM (2017). ILOG CPLEX Optimizer v. 12.7.1. `http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/`.

[48] Jeyakumar, V. and Li, G. (2011). Robust duality for fractional programming problems with constraint-wise data uncertainty. *Journal of Optimization Theory and Applications*, 151(2):292.

[49] Jeyakumar, V., Li, G., and Srisatkunarajah, S. (2013). Strong duality for robust minimax fractional programming problems. *European Journal of Operational Research*, 228(2):331–336.

[50] Jović, A., Brkić, K., and Bogunović, N. (2015). A review of feature selection methods with applications. In *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1200–1205. IEEE.

[51] Keane, M. P. and Wada, N. (2012). *Estimation of discrete choice models with many alternatives using random subsets of the full choice set: With an application to demand for frozen pizza*. Nuffield College.

[52] Kocheturov, A., Pardalos, P. M., and Karakitsiou, A. (2019). Massive datasets and machine learning for computational biomedicine: trends and challenges. *Annals of Operations Research*, 276(1-2):5–34.

[53] Lawler, E. L. (2001). *Combinatorial optimization: networks and matroids*. Courier Corporation.

[54] Li, H.-L. (1994). A global approach for general 0–1 fractional programming. *European Journal of Operational Research*, 73(3):590–596.

[55] Li, J., Cheng, K., Wang, S., Morstatter, F., Robert, T., Tang, J., and Liu, H. (2016). Feature selection: A data perspective. *arXiv:1601.07996*.

[56] Liu, H. and Motoda, H. (2012). *Feature selection for knowledge discovery and data mining*, volume 454. Springer Science & Business Media.

[57] Lovász, L. (1983). Submodular functions and convexity. In *Mathematical Programming The State of the Art*, pages 235–257. Springer.

[58] MacKay, D. J. and Mac Kay, D. J. (2003). *Information theory, inference and learning algorithms*. Cambridge university press.

[59] McCormick, S. T. and Ervolina, T. R. (1994). Computing maximum mean cuts. *Discrete Applied Mathematics*, 52(1):53–70.

[60] Megiddo, N. (1979). Combinatorial optimization with rational objective functions. *Mathematics of Operations Research*, 4(4):414–424.

[61] Mehmanchi, E., Gillen, C. P., Gómez, A., and Prokopyev, O. A. (2019a). On robust fractional 0-1 programming. *INFORMS Journal on Optimization*. `http://doi.org/10.1287/ijoo.2019.0025`.

[62] Mehmanchi, E., Gómez, A., and Prokopyev, O. A. (2019b). Fractional 0–1 programs: Links between mixed-integer linear and conic quadratic formulations. *Journal of Global Optimization*, 75(2):273–339. `https://doi.org/10.1007/s10898-019-00817-7`.

[63] Méndez-Díaz, I., Miranda-Bront, J. J., Vulcano, G., and Zabala, P. (2014). A branch-and-cut algorithm for the latent-class logit assortment problem. *Discrete Applied Mathematics*, 164:246–263.

[64] Miller, A. (2002). *Subset selection in regression*. Chapman and Hall/CRC.

[65] Moeini, M. (2015). The maximum ratio clique problem: a continuous optimization approach and some new results. In *Modelling, Computation and Optimization in Information Systems and Management Sciences*, pages 215–227. Springer.

[66] Nemhauser, G. L. and Wolsey, L. A. (1988). Integer programming and combinatorial optimization. *Wiley, Chichester. GL Nemhauser, MWP Savelsbergh, GS Sigismondi (1992). Constraint Classification for Mixed Integer Programming Formulations. COAL Bulletin*, 20:8–12.

[67] Nguyen, H., Franke, K., and Petrovic, S. (2009). Optimizing a class of feature selection measures. In *NIPS 2009 Workshop on Discrete Optimization in Machine Learning: Submodularity, Sparsity & Polyhedra (DISCML), Vancouver, Canada*.

[68] Nguyen, H., Franke, K., and Petrovic, S. (2010a). Improving effectiveness of intrusion detection by correlation feature selection. In *Availability, Reliability, and Security, 2010. ARES'10 International Conference on*, pages 17–24. IEEE.

[69] Nguyen, H. T., Franke, K., and Petrovic, S. (2010b). Towards a generic feature-selection measure for intrusion detection. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 1529–1532. IEEE.

[70] Nguyen, H. T., Franke, K., and Petrović, S. (2011). A new ensemble-feature-selection framework for intrusion detection. In *2011 11th International Conference on Intelligent Systems Design and Applications*, pages 213–218. IEEE.

[71] Palubeckis, G. (2004). Multistart tabu search strategies for the unconstrained binary quadratic optimization problem. *Annals of Operations Research*, 131(1-4):259–282.

[72] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python . *Journal of Machine Learning Research*, 12:2825–2830.

[73] Peng, H., Long, F., and Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, 27(8):1226–1238.

[74] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1992). Numerical recipes in C++. *The art of scientific computing*, 2:1002.

[75] Prokopyev, O. A. (2008). Fractional zero-one programming. In *Encyclopedia of Optimization*, pages 1091–1094. Springer.

[76] Prokopyev, O. A., Huang, H.-X., and Pardalos, P. M. (2005a). On complexity of unconstrained hyperbolic 0–1 programming problems. *Operations Research Letters*, 33(3):312–318.

[77] Prokopyev, O. A., Meneses, C., Oliveira, C. A., and Pardalos, P. M. (2005b). On multiple-ratio hyperbolic 0–1 programming problems. *Pacific Journal of Optimization*, 1(2):327–345.

[78] Python Software Foundation (2019). Python Language Reference v. 3.7.3. `https://www.python.org/`.

[79] Radzik, T. (2013). Fractional combinatorial optimization. In *Handbook of Combinatorial Optimization*, pages 1311–1355. Springer.

[80] Rusmevichientong, P., Shen, Z.-J. M., and Shmoys, D. B. (2010). Dynamic assortment optimization with a multinomial logit choice model and capacity constraint. *Operations Research*, 58(6):1666–1680.

[81] Rusmevichientong, P. and Topaloglu, H. (2012). Robust assortment optimization in revenue management under the multinomial logit choice model. *Operations Research*, 60(4):865–882.

[82] Saeys, Y., Inza, I., and Larrañaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517.

[83] Saipe, A. L. (1975). Solving a (0, 1) hyperbolic program by branch and bound. *Naval Research Logistics*, 22(3):497–515.

[84] Schaible, S. and Ibaraki, T. (1983). Fractional programming. *European Journal of Operational Research*, 12(4):325–338.

[85] Şen, A., Atamtürk, A., and Kaminsky, P. (2018). A conic integer optimization approach to the constrained assortment problem under the mixed multinomial logit model. *Operations Research*, 66(4):994–1003.

[86] Sethuraman, S. and Butenko, S. (2015). The maximum ratio clique problem. *Computational Management Science*, 12(1):197–218.

[87] Shigeno, M., Saruwatari, Y., and Matsui, T. (1995). An algorithm for fractional assignment problems. *Discrete Applied Mathematics*, 56(2-3):333–343.

[88] Stancu-Minasian, I. (2006). A sixth bibliography of fractional programming. *Optimization*, 55(4):405–428.

[89] Subramanian, S. and Sherali, H. D. (2010). A fractional programming approach for retail category price optimization. *Journal of Global Optimization*, 48(2):263–277.

[90] Talluri, K. and Van Ryzin, G. (2004). Revenue management under a general discrete choice model of consumer behavior. *Management Science*, 50(1):15–33.

[91] Tang, J., Alelyani, S., and Liu, H. (2014). Feature selection for classification: A review. *Data Classification: Algorithms and Applications*, page 37.

[92] Tawarmalani, M., Ahmed, S., and Sahinidis, N. V. (2002). Global optimization of 0-1 hyperbolic programs. *Journal of Global Optimization*, 24(4):385–416.

[93] Trapp, A., Prokopyev, O. A., and Busygin, S. (2010). Finding checkerboard patterns via fractional 0–1 programming. *Journal of Combinatorial Optimization*, 20(1):1–26.

[94] Trapp, A. C. and Konrad, R. A. (2015). Finding diverse optima and near-optima to binary integer programs. *IIE Transactions*, 47(11):1300–1312.

[95] Ursulenko, O., Butenko, S., and Prokopyev, O. A. (2013). A global optimization algorithm for solving the minimum multiple ratio spanning tree problem. *Journal of Global Optimization*, 56(3):1029–1043.

[96] Vielma, J. P., Ahmed, S., and Nemhauser, G. L. (2008). A lifted linear programming branch-and-bound algorithm for mixed-integer conic quadratic programs. *INFORMS Journal on Computing*, 20(3):438–450.

[97] Vielma, J. P., Dunning, I., Huchette, J., and Lubin, M. (2017). Extended formulations in mixed integer conic quadratic programming. *Mathematical Programming Computation*, 9(3):369–418.

[98] Viola, M., Sangiovanni, M., Toraldo, G., and Guarracino, M. R. (2017). A generalized eigenvalues classifier with embedded feature selection. *Optimization Letters*, 11(2):299–311.

[99] Williams, H. P. (1974). *Experiments in the formulation of integer programming problems*. Springer.

[100] Wu, T.-H. (1997). A note on a global approach for general 0–1 fractional programming. *European Journal of Operational Research*, 101(1):220–223.

[101] Yu, L. and Liu, H. (2003). Feature selection for high-dimensional data: A fast correlation-based filter solution. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pages 856–863.

[102] Yuan, H., Xu, W., Li, Q., and Lau, R. (2018). Topic sentiment mining for sales performance prediction in e-commerce. *Annals of Operations Research*, 270(1-2):553–576.