REGRET MODELS AND PREPROCESSING TECHNIQUES FOR COMBINATORIAL OPTIMIZATION UNDER UNCERTAINTY

SHI DONGJIAN

(B.Sc., NJU, China)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF MATHEMATICS

NATIONAL UNIVERSITY OF SINGAPORE

2013

To my parents

DECLARATION

I hereby declare that the thesis is my original work and it has been written by me in its entirety.

I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Shi Dongjian

June 2013

Acknowledgements

First and foremost, I would like to express my heartfelt gratitude to my Ph.D supervisor Professor Toh Kim Chuan for his support and encouragement, guidance and assistance in my studies and research work and especially, for his patience and advice on the improvement of my skills in both research and writing. His amazing depth of knowledge and tremendous expertise in optimization have greatly facilitated my research progress. His wisdom and attitude will always be a guide to me, and I feel very proud to be one of his Ph.D students.

I owe my deepest gratitude to Professor Karthik Natarajan. He was my first advisor who led me, hand in hand, to the world of the academic research. Even after he left NUS, he still continued to discuss the research questions with me almost every week. Karthik has never turned away from me in case I need any help. He has always shared his insightful ideas with me and encouraged me to do deep research, even though sometimes I lacked confidence in myself. Without his excellent mathematical knowledge and professional guidance, it would not have been possible to complete this doctoral thesis. I am greatly indebted to him.

I would like to give special thanks to Professor Sun Defeng who interviewed me five years ago and brought me to NUS. I feel very honored to have worked together with him as a tutor for the course Discrete Optimization for two semesters.

Many grateful thanks also go to Professor Zhao Gongyun for his introduction on mathematical programming, which I found to be the most basic and important optimization course I took in NUS. His excellent teaching style helped me to gain broad knowledge on numerical optimization and software.

I am also thankful to all my friends in Singapore for their kind help. Special thanks to Dr. Jiang Kaifeng, Dr. Miao Weimin, Dr. Ding Chao, Dr. Chen Caihua, Gong Zheng, Wu Bin, Li Xudong and Du Mengyu for their helpful discussions on many interesting optimization topics.

I would like to thank the Department of Mathematics, National University of Singapore for providing me excellent research conditions and a scholarship to complete my Ph.D study. I would also like to thank the Faculty of Science for providing me the financial support for attending the 21st International Symposium on Mathematical Programming, Berlin, Germany.

I am as ever, especially indebted to my parents, for their unconditional love and support all through my life. Last but not least, I would express my gratitude and love to my wife, Wang Xiaoyan, for her love and companionship during my five years Ph.D study period.

Shi Dongjian

June 2013

Contents

Summary

In this thesis, we consider probabilistic models for linear and quadratic combinatorial optimization problems under uncertainty. Firstly, we propose a new probabilistic model for minimizing the anticipated regret in combinatorial optimization problems with distributional uncertainty in the objective coefficients. The interval uncertainty representation of data is supplemented with information on the marginal distributions. As a decision criterion, we minimize the worst-case conditional value-at-risk of regret. For the class of combinatorial optimization problems with a compact convex hull representation, polynomial sized mixed integer linear programs (MILP) and mixed integer second order cone programs (MISOCP) are formulated. Secondly, for the subset selection problem of choosing K elements of maximum total weight out of a set of N elements, we show that the proposed probabilistic regret model is solvable in polynomial time under some specific distributional models. This extends the current known polynomial complexity result for minmax regret subset selection with range information only. A similar idea is used to find a polynomial time algorithm for the distributionally robust k-sum optimization problem. Finally, we develop a preprocessing technique to solve parametric quadratic unconstrained binary optimization problems where the uncertain parameter are described by probabilistic information.

List of Tables

List of Figures

Notations

- $\Re, \Re^N, \Re^{N\times N}$ denote the set of real numbers, N dimensional Euclidean space and $N \times N$ dimensional matrix space, respectively.
- Bold lower case letters such as $\boldsymbol{x}$ represents vectors and the upper case letters such as $\boldsymbol{A}$ denotes matrices.
- The tilde sign is used to denote random variables and random vectors, e.g., $\tilde{r}$, $\tilde{\boldsymbol{c}}$.
- For a real number x, x^+ denotes $\max\{x, 0\}$.
- $[N]$ denotes the set $\{1, 2, \ldots, N\}$, where N is a positive integer.
- $\|\cdot\|_2$ denotes the L_2 norm of a vector.
- $\succeq$ denotes the partial order partial relative to positive semidefinite cone, e.g., $\boldsymbol{A} \succeq 0$ means $\boldsymbol{A}$ is positive semidefinite.
- $rand(N, 1)$ denotes a function which returns an N-by-1 matrix containing pseudo random values drawn from the standard uniform distribution.
- $randn(N, 1)$ denotes a function which returns an N-by-1 matrix containing pseudo random values drawn from the standard normal distribution.

Chapter 1

Introduction

In this thesis, we focus on probabilistic models for combinatorial optimization with uncertainty. First, we consider the linear combinatorial optimization problem

$$\max_{\boldsymbol{x}\in\mathcal{X}} \tilde{\boldsymbol{c}}^T\boldsymbol{x}, \tag{1.1}$$

where $\mathcal{X} \subseteq \{0,1\}^N$. The uncertainty lies in the random objective coefficients $\tilde{\boldsymbol{c}}$. By assuming partial distributional information on $\tilde{\boldsymbol{c}}$, we propose a new probabilistic regret model that incorporates partial distributional information such as the mean and variance of the random coefficients.

Besides the linear combinatorial optimization problem, we also consider the quadratic unconstrained binary optimization (QUBO) problem

$$\max_{\boldsymbol{x}\in\{0,1\}^N} \boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \tilde{\boldsymbol{c}}^T\boldsymbol{x}, \tag{1.2}$$

where $\boldsymbol{Q}$ is a fixed $N \times N$ symmetric real matrix, and the parameter vector $\tilde{\boldsymbol{c}}$ is random. By assuming partial distributional information on $\tilde{\boldsymbol{c}}$, we propose a new preprocessing technique to solve a parametrical set of QUBO problems.

Structure of the chapter: In section 1.1, we introduce the motivation of the proposed probabilistic models and review the related literature. In section 1.2, we outline the organization and main contributions of this thesis.

1.1 Motivation and Literature Review

Data uncertainty is present in many real-world optimization problems. For example, we do not know the exact completion time of a job in a project management problem. Similarly, we do not know the precise time spent on a road if we want to travel to a destination. Uncertainty is incorporated into such optimization models with a goal of formulating this kind of problem to a tractable optimization problem which can be solved analytically or numerically in order to help the decision-maker to make good decisions.

Stochastic programming is a classical uncertainty model which was proposed in the 1950s by Dantzig [41]. It is a framework for modeling optimization problems that involve random uncertainty. In stochastic programming, the probabilistic distribution of the uncertain data is assumed to be known or can be estimated. The goal of this model is to find a policy that is feasible for all (or almost all) the possible data instances and minimizes or maximizes the expectation of a utility function of the decisions and the random variables. For example, the stochastic programming model for problem (1.1) is

$$\max_{\boldsymbol{x}\in\mathcal{X}} \mathbb{E}_P[U(\tilde{\boldsymbol{c}}^T\boldsymbol{x})],$$

where U is a utility function of the profit $\tilde{\boldsymbol{c}}^T\boldsymbol{x}$. Stochastic programming has been widely used in the applications of portfolio selection, project management and so on in the past few decades, and many efficient numerical methods have been addressed to deal with this model. While this model can deal with uncertain data with given distributions, there are some fundamental difficulties with it. First, it is often difficult to obtain the actual distributions of the uncertain parameters from data. Moreover, even if we know the distributions, it still can be computationally challenging to evaluate the expected utility.

When the parameters are uncertain and known to lie in a deterministic set, robust optimization is used to tackle the optimization problem. The origins of robust optimization date back to the establishment of modern decision theory in

the 1950s and the use of worst case analysis and Wald's maxmin model as a tool for the treatment of severe uncertainty [118, 119]. A simple robust optimization model for problem (1.1) is

$$\max_{\boldsymbol{x}\in\mathcal{X}} \min_{\boldsymbol{c}\in\Omega} \boldsymbol{c}^T\boldsymbol{x},$$

where Ω represent the set of possible scenario vectors for $\tilde{\boldsymbol{c}}$. Robust optimization became a field of its own in the 1970s with parallel developments in fields such as operations research, control theory, statistics, economics, and more [24, 112, 80, 46, 123, 19, 36]. In traditional robust optimization, only the worst case scenario is considered. Hence this model is often considered to be very conservative since it may lose additional information of the uncertain parameters.

To use additional probabilistic information of the random data, distributionally robust optimization models have been developed to make decisions when partial distributional information (e.g. mean , variance and so on) of the random data is given [58, 42]. The objective of this model is to maximize (or minimize) the expected utility (or disutility) for a worst case distribution with the given probabilistic information. For the random linear combinatorial optimization problem (1.1), by considering its equivalent minimization form $\min_{\boldsymbol{x}\in\mathcal{X}} -\tilde{\boldsymbol{c}}^T\boldsymbol{x}$, the distributionally robust optimization model is written as

$$\min_{\boldsymbol{x}\in\mathcal{X}} \sup_{P\in\mathbb{P}} \mathbb{E}_P[D(-\tilde{\boldsymbol{c}}^T\boldsymbol{x})],$$

where $\mathbb{P}$ is the set of all the possible distributions for the random vector $\tilde{\boldsymbol{c}}$ described by the given partial distributional information, and D is a disutility function of the cost $-\tilde{\boldsymbol{c}}^T\boldsymbol{x}$. Distributionally robust optimization can be viewed as being more conservative than stochastic programming and less conservative than robust optimization. Hence it can be an effective model to make good decisions when some partial distributional information of the uncertain data is given.

Besides the above models, another probabilistic model that will be considered in this thesis to is to find an optimal decision to minimize a risk measure of the random objective. For (1.1), the problem of minimizing the risk measure of the

random cost is as follows:

$$\min_{\boldsymbol{x}\in\mathcal{X}} \rho(-\tilde{\boldsymbol{c}}^T\boldsymbol{x}), \tag{1.3}$$

where ρ is a risk measure which is an increasing function of the cost $-\tilde{\boldsymbol{c}}^T\boldsymbol{x}$. We consider the model by choosing a proper ρ which has all the good properties of coherent risk measures. The definition of convex and coherent risk measures that is commonly used will be reviewed in the following subsection.

1.1.1 Convex and Coherent Risk Measures

In this subsection, we briefly review the definition of the convex and coherent risk measures. One of the basic tasks in finance is to quantify the risk associated with a given financial position, which is subject to uncertainty. Let Ω be a deterministic uncertainty set that captures all the possible realizations. Because of the uncertainty, the profit and loss of such a financial position is a random variable $\tilde{r}(\omega) : \Omega \to \Re$, where $\tilde{r}(\omega)$ is the loss of the position at the end of the trading period if the scenario $\omega \in \Omega$ is realized. The goal is to determine a real number $\rho(\tilde{r})$ which quantifies the risk and can be used as a decision criterion. For example, in the classical Markowitz model the portfolio return variance is used to be a quantification of the risk. In the last two decades, the theory of risk measures has been developed extensively. The following axiomatic approach to risk measures was initiated in the coherent case by Artzner et al. [8] and later independently extended to the class of convex risk measures by Föllmer and Schied [47], and Fritelli and Gianin [48].

Definition 1.1. *Consider a set X of random variables. A mapping $\rho : X \to \Re$ is called a convex risk measure if it satisfies the following conditions for all $\tilde{x}, \tilde{y} \in X$.*

1. *Monotonicity: If $\tilde{x} \leq \tilde{y}$, i.e. $\tilde{x}$ dominates $\tilde{y}$ for each outcome, then $\rho(\tilde{x}) \leq \rho(\tilde{y})$.*

2. *Translation invariance: If $c \in \Re$, then $\rho(\tilde{x} + c) = \rho(\tilde{x}) + c$.*

3. Convexity: If $\lambda \in [0,1]$, *then* $\rho(\lambda\tilde{x} + (1-\lambda)\tilde{y}) \leq \lambda\rho(\tilde{x}) + (1-\lambda)\rho(\tilde{y})$.

The convex risk measure ρ *is called a coherent risk measure if it satisfies the additional condition*

4. Positive homogeneity: If $\lambda \geq 0$, *then* $\rho(\lambda\tilde{x}) = \lambda\rho(\tilde{x})$.

A well-known example of coherent risk measures is the conditional value-at-risk (CVaR). Conditional value-at-risk is also referred to as average value-at-risk or expected shortfall in the risk management literature. We briefly review this concept here. Consider a random variable $\tilde{r}$ defined on a probability space $(\Pi, \mathcal{F}, Q)$, i.e. a real valued function $\tilde{r}(\omega) : \Pi \to \Re$, with finite second moment $\mathbb{E}[\tilde{r}^2] < \infty$. This ensures that the conditional value-at-risk is finite. For example, the finiteness of the second moment is guaranteed if the random variables are assumed to lie within a finite range. For a given $\alpha \in (0,1)$, the value-at-risk is defined as the lower α quantile of the random variable $\tilde{r}$:

$$\mathrm{VaR}_\alpha(\tilde{\mathrm{r}}) = \inf\left\{\mathrm{v} \mid \mathrm{Q}(\tilde{\mathrm{r}} \leq \mathrm{v}) \geq \alpha\right\}. \tag{1.4}$$

The definition of conditional value-at-risk is provided next.

Definition 1.2 (Rockafellar and Uryasev [103, 104], Acerbi and Tasche [1])**.** *For* $\alpha \in (0,1)$, *the conditional value-at-risk* (CVaR) *at level* α *of a random variable* $\tilde{r}(\omega) : \Pi \to \Re$ *is the average of the highest* $1-\alpha$ *of the outcomes:*

$$\mathrm{CVaR}_\alpha(\tilde{\mathrm{r}}) = \frac{1}{1-\alpha}\int_\alpha^1 \mathrm{VaR}_\beta(\tilde{\mathrm{r}})\mathrm{d}\beta. \tag{1.5}$$

An equivalent representation for CVaR *is:*

$$\mathrm{CVaR}_\alpha(\tilde{\mathrm{r}}) = \inf_{\mathrm{v}\in\Re}\left(\mathrm{v} + \frac{1}{1-\alpha}\mathbb{E}_\mathrm{Q}[\tilde{\mathrm{r}} - \mathrm{v}]^+\right). \tag{1.6}$$

From the above definition, we can easily check that $\rho(\tilde{r}) = \mathrm{CVaR}_\alpha(\tilde{\mathrm{r}})$ is an example of coherent risk measures which satisfies all the four axioms in Definition 1.1. Furthermore, CVaR is an attractive risk measure for stochastic optimization

since it is convexity preserving unlike the VaR measure. However the computation of CVaR might still be intractable (see Ben-Tal et. al. [15] for a detailed discussion on this). An instance when the computation of CVaR is tractable is for discrete distributions with a polynomial number of scenarios. Optimization with the CVaR measure has been used in portfolio optimization [103] and inventory control [3] among other stochastic optimization problems. Combinatorial optimization problems under the CVaR measure has been studied by So et. al. [114]:

$$\min_{\boldsymbol{x}\in\mathcal{X}} \mathrm{CVaR}_{\alpha}\left(-\tilde{\boldsymbol{c}}^{\mathrm{T}}\boldsymbol{x}\right). \tag{1.7}$$

The negative sign in Formulation (1.7) capture the feature that higher values of $\boldsymbol{c}^T\boldsymbol{x}$ are preferred to lower values. Using a sample average approximation method, So et. al. [114] propose approximation algorithms to solve (1.7) for covering, facility location and Steiner tree problems. In the distributional uncertainty representation, the concept of conditional value-at-risk is extended to the concept of worst-case conditional value-at-risk through the following definition.

Definition 1.3. *[Zhu and Fukushima [125], Natarajan et. al. [90]] Suppose the distribution of the random variable $\tilde{r}$ lies in a set $\mathbb{Q}$. For $\alpha \in (0,1)$, the worst-case conditional value-at-risk* (WCVaR) *at level α of a random variable $\tilde{r}$ with respect to $\mathbb{Q}$ is defined as:*

$$\mathrm{WCVaR}_{\alpha}(\tilde{r}) = \sup_{Q\in\mathbb{Q}} \inf_{v\in\Re} \left(v + \frac{1}{1-\alpha}\mathbb{E}_Q[\tilde{r}-v]^+ \right). \tag{1.8}$$

From an axiomatic perspective, WCVaR has also been shown to be a coherent risk measure under mild assumptions on the set of distributions (see the discussions in Zhu and Fukushima [125] and Natarajan et. al. [90]). WCVaR has been used as a risk measure in distributionally robust portfolio optimization [125, 90] and joint chance constrained optimization problems [35, 127]. Zhu and Fukushima [125] and Natarajan et. al. [90] also provide examples of sets of distributions $\mathbb{Q}$ where the position of sup and inf can be exchanged in formula (1.8). Since the objective is linear in the probability measure (possibly infinite-dimensional) over which it

is maximized and convex in the variable v over which it is minimized, the saddle point theorem from Rockafellar [105] is applicable. Applying Theorem 6 in [105] implies the following lemma:

Lemma 1.4. *Let $\alpha \in (0,1)$, and the distribution of the random variable $\tilde{r}$ lies in a set $\mathbb{Q}$. If $\mathbb{Q}$ is a convex set of the probability distributions defined on a closed convex support set $\Omega \subseteq \Re^n$, then*

$$\mathrm{WCVaR}_\alpha(\tilde{r}) = \inf_{v \in \Re} \left(v + \frac{1}{1-\alpha} \sup_{Q \in \mathbb{Q}} \mathbb{E}_Q[\tilde{r} - v]^+ \right). \tag{1.9}$$

The above lemma tell us that we can exchange the position of inf and sup in the definition of WCVaR. We use (1.9) to compute WCVaR for random variables with partial distributional information in the following sections. Throughtout this thesis, the distribution set we consider is always assumed to satisfy the condition in Lemma 1.4.

1.1.2 Minmax Regret and Distributional Models

The regret model was first proposed by Savage (1951) [107] to deal with optimization problems with uncertainty. In decision theory, regret is defined as the difference between the actual payoff and the payoff that would have been obtained if a different course of action had been chosen. The main difference between the regret model and cost (or profit) models is that we minimize the regret of the decision-maker in the regret model, while we optimize the cost (or profit) in the second class of models.

Let $Z(\boldsymbol{c})$ denote the optimal value to a linear combinatorial optimization problem over a feasible region $\mathcal{X} \subseteq \{0,1\}^N$ for a given objective coefficient vector $\boldsymbol{c}$:

$$Z(\boldsymbol{c}) = \max\{\boldsymbol{c}^T \boldsymbol{x} \mid \boldsymbol{x} \in \mathcal{X} \subseteq \{0,1\}^N\}. \tag{1.10}$$

Consider a decision-maker who needs to decide on a feasible solution $\boldsymbol{x} \in \mathcal{X}$ before knowing the actual value of the objective coefficients. This decision-maker experiences an ex-post regret of possibly not choosing the optimal solution, and the

value of his regret is given by:

$$R(\boldsymbol{x}, \boldsymbol{c}) = Z(\boldsymbol{c}) - \boldsymbol{c}^T\boldsymbol{x} = \max_{\boldsymbol{y}\in\mathcal{X}} \boldsymbol{c}^T\boldsymbol{y} - \boldsymbol{c}^T\boldsymbol{x}. \tag{1.11}$$

Let Ω represent the set of possible scenario vectors for $\boldsymbol{c}$. The maximum regret for the decision $\boldsymbol{x}$ corresponding to the uncertainty set Ω is:

$$\max_{\boldsymbol{c}\in\Omega} R(\boldsymbol{x}, \boldsymbol{c}). \tag{1.12}$$

Under a minmax regret approach, $\boldsymbol{x}$ is chosen such that it minimizes the maximum regret over all possible realizations of the objective coefficients, i.e.,

$$\min_{\boldsymbol{x}\in\mathcal{X}} \max_{\boldsymbol{c}\in\Omega} R(\boldsymbol{x}, \boldsymbol{c}). \tag{1.13}$$

One of the early references on the minmax regret model for combinatorial optimization problems is Kouvelis and Yu [83] which discusses the complexity of solving this class of problems. The computational complexity of the regret problem has been extensively studied under the following two representations of Ω [83, 9, 76, 77, 37].

(a) Scenario uncertainty: The vector $\boldsymbol{c}$ lies in a finite set of M possible discrete scenarios:

$$\Omega = \{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_M\}. \tag{1.14}$$

(b) Interval uncertainty: Each component c_i of the vector $\boldsymbol{c}$ takes a value between a lower bound $\underline{c}_i$ and upper bound $\overline{c}_i$. Let $\Omega_i = [\underline{c}_i, \overline{c}_i]$ for $i = 1, \ldots, N$. The uncertainty set is the Cartesian product of the sets of intervals:

$$\Omega = \Omega_1 \times \Omega_2 \times \ldots \times \Omega_N. \tag{1.15}$$

For the discrete scenario uncertainty, the minmax regret counterpart of problems such as the shortest path, minimum assignment and minimum spanning tree problems are NP-hard even when the scenario set contains only two scenarios (see Kouvelis and Yu [83]). This indicates the difficulty of solving regret problems to optimality since the original deterministic optimization problems are solvable in

polynomial time in these instances. These problems are weakly NP-hard for a constant number of scenarios while they become strongly NP-hard when the number of scenarios is non-constant.

In the interval uncertainty case, for deterministic combinatorial optimization problems with a compact convex hull representation, a mixed integer linear programming formulation for the minmax regret problem (1.13) was proposed by Yaman et. al. [121]. As in the scenario uncertainty case, the minmax regret counterpart is NP-hard under interval uncertainty for most classical polynomial time solvable combinatorial optimization problems. Averbakh and Lebedev [10] proved that the minmax regret shortest path and minmax regret minimum spanning tree problems are strongly NP-hard with interval uncertainty. Under the assumption that the deterministic problem is polynomial time solvable, a 2-approximation algorithm for minmax regret was designed by Kasperski and Zieliński [77]. Their algorithm is based on a mid-point scenario approach where the deterministic combinatorial optimization problem is solved with an objective coefficient vector $(\underline{\boldsymbol{c}}+\overline{\boldsymbol{c}})/2$. Kasperski and Zieliński [78] developed a fully polynomial time approximation scheme under the assumption that a pseudopolynomial algorithm is available for the deterministic problem. A special case where the minmax regret problem is solvable in polynomial time is the subset selection problem. The deterministic subset selection problem is: Given a set of elements $[N] := \{1, \ldots, N\}$ with weights $\{c_1, \ldots, c_N\}$, select a subset of K elements of maximum total weight. The deterministic problem can be solved by a simple sorting algorithm. With an interval uncertainty representation of the weights, Averbakh [9] designed a polynomial time algorithm to solve the minmax regret problem to optimality with a running time of $O(N \min(K, N-K)^2)$. Subsequently, Conde [37] designed a faster algorithm to solve this problem with running time $O(N \min(K, N-K))$.

A related model that has been analyzed in discrete optimization is the absolute robust approach (see Kouvelis and Yu [83] and Bertsimas and Sim [23]) where the decision-maker chooses a decision $\boldsymbol{x}$ that maximizes the minimum objective over

all possible realizations of the uncertainty:

$$\max_{\boldsymbol{x}\in\mathcal{X}} \min_{\boldsymbol{c}\in\Omega} \boldsymbol{c}^T\boldsymbol{x}. \tag{1.16}$$

Problem (1.16) is referred to as the absolute robust counterpart of the deterministic optimization problem. The formulation for the absolute robust counterpart should be contrasted with the minmax regret formulation which can be viewed as the relative robust counterpart of the deterministic optimization problem. For the discrete scenario uncertainty, the absolute robust counterpart of the shortest path problem is NP-hard as in the regret setting (see Kouvelis and Yu [83]). However for the interval uncertainty case, the absolute robust counterpart retains the complexity of the deterministic problem unlike the minmax regret counterpart. This follows from the observation that the worst case realization of the uncertainty in absolute terms is to set the objective coefficient vector to the lower bound $\underline{\boldsymbol{c}}$ irrespective of the solution $\boldsymbol{x}$. The minmax regret version in contrast is more difficult to solve since the worst case realization depends on the solution $\boldsymbol{x}$. However this also implies that the minmax regret solution is less conservative as it considers both the best and worst case. For illustration, consider the binary decision problem of deciding whether to invest or not in a single project with payoff c:

$$Z(c) = \max\{cy \mid y \in \{0,1\}\}.$$

The payoff is uncertain and takes a value in the range $c \in [\underline{c}, \overline{c}]$ where $\underline{c} < 0$ and $\overline{c} > 0$. The absolute robust solution is to not invest in the project since in the worst case the payoff is negative. On the other hand, the minmax regret solution is to invest in the project if $\overline{c} > -\underline{c}$ (the best payoff is more than the magnitude of the worst loss) and not invest in the project otherwise. Since the regret criterion evaluates the performance with respect to the best decision, it is not as conservative as the absolute robust solution. However the computation of the minmax regret solution is more difficult than the absolute robust solution.

In the minmax regret model, other than the supports of the random parameters, no information on the probability distribution is considered. Our goal is

to develop a model which incorporates probabilistic information and the decision-maker's attitude to regret. We use worst-case conditional value at risk (WCVaR) to incorporate the distributional information and the regret aversion attitude. The problem of interest is to minimize the WCVaR at probability level α of the regret for some random combinatorial optimization problems:

$$\min_{\boldsymbol{x}\in\mathcal{X}} \mathrm{WCVaR}_\alpha(R(\boldsymbol{x},\tilde{\boldsymbol{c}})). \tag{1.17}$$

By the definition of WCVaR and Lemma 1.4, the central problem (1.17) is written as

$$\min_{\boldsymbol{x}\in\mathcal{X},v\in\Re}\left\{v+\frac{1}{1-\alpha}\sup_{P\in\mathbb{P}}\mathbb{E}_P[R(\boldsymbol{x},\tilde{\boldsymbol{c}})-v]^+\right\}. \tag{1.18}$$

To generalize the interval uncertainty model supplemental marginal distributional information of the random vector $\tilde{\boldsymbol{c}}$ is assumed to be given. The random variables are however not assumed to be independent. Throughout this thesis, the following two models for the distribution set $\mathbb{P}$ are considered:

(a) **Marginal distribution model:** For each $i \in [N]$, the marginal probability distribution P_i of $\tilde{c}_i$ with support $\Omega_i = [\underline{c}_i, \overline{c}_i]$ is assumed to be given. Let $\mathbb{P}(P_1, \ldots, P_N)$ denote the set of joint distributions with the fixed marginals.

(b) **Marginal moment model:** For each $i \in [N]$, the probability distribution P_i of $\tilde{c}_i$ with support $\Omega_i = [\underline{c}_i, \overline{c}_i]$ is assumed to belong to a set of probability measures $\mathbb{P}_i$. The set $\mathbb{P}_i$ is defined through moment equality constraints on real-valued functions of the form $\mathbb{E}_{P_i}[f_{ik}(\tilde{c}_i)] = m_{ik}, k \in [K_i]$. If $f_{ik}(c_i) = c_i^k$, this reduces to knowing the first K_i moments of $\tilde{c}_i$. Let $\mathbb{P}(\mathbb{P}_1, \ldots, \mathbb{P}_N)$ denote the set of multivariate joint distributions compatible with the marginal probability distributions $P_i \in \mathbb{P}_i$. Throughout the paper, we assume that mild Slater type conditions hold on the moment information to guarantee that strong duality is applicable for moment problems. One such simple sufficient condition is that the moment vector is in the interior of the set of feasible moments (see Isii [72]). With the marginal moment specification, the multivariate moment space is the product of univariate moment spaces. Ensuring

that Slater type conditions hold in this case is relatively straightforward since it reduces to Slater conditions for univariate moment spaces. The reader is referred to Bertsimas et. al. [21] and Lasserre [84] for a detailed description on this topic.

The above two distributional models only capture the marginal information and they are commonly referred to as the Fréchet class of distributions in probability [40, 39]. In the thesis, we extend several existing results for the minmax regret model to the proposed probabilistic regret model under the Fréchet class of distributions. Moreover, some of the results obtained can be directly used to the problem of minimizing the WCVaR of cost:

$$\min_{\boldsymbol{x}\in\mathcal{X}} \mathrm{WCVaR}_{\alpha}(-\tilde{\boldsymbol{c}}^T\boldsymbol{x}). \tag{1.19}$$

Formulation (1.19) can be viewed as a regret minimization problem where the regret is defined with respect to an absolute benchmark of zero.

1.1.3 Quadratic Unconstrained Binary Optimization

Besides the linear combinatorial optimization with uncertainty, we also consider the quadratic unconstrained binary optimization problem. Define the quadratic function:

$$q(\boldsymbol{x};\boldsymbol{c},\boldsymbol{Q}) = \boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c}^T\boldsymbol{x}$$

and the corresponding quadratic unconstrained binary optimization:

$$\text{(QUBO)} \quad \max_{\boldsymbol{x}\in\{0,1\}^N} q(\boldsymbol{x};\boldsymbol{c},\boldsymbol{Q}), \tag{1.20}$$

where $\boldsymbol{Q}$ is a $N\times N$ real symmetric matrix (not necessarily negative semidefinite), and $\boldsymbol{c}\in\Re^N$.

Quadratic unconstrained binary optimization (QUBO) has applications in a number of diverse areas including computer-aided design (Boros and Hammer [31], Jünger et. al. [74]), solid-state physics (Barahona [12], Simone et. al. [113]), and

machine scheduling (Alidaee et. al. [5]). Several graph problems, such as the max-cut and the maximum clique problems can be reformulated as QUBO problems. As a result, QUBO is known to be NP-hard (see Garey and Johnson [51]). A variety of heuristics and exact methods that run in exponential time have been proposed to solve QUBO problems. When all the off-diagonal components of $\boldsymbol{Q}$ are nonnegative, QUBO is solvable in polynomial time (see Picard and Ratliff [97]). In this case, QUBO is equivalent to the following linear programming relaxation:

$$\begin{aligned} \max_{\boldsymbol{x},\boldsymbol{X}} \quad & \sum_{i=1}^{N}\sum_{j=1}^{N} Q_{ij}X_{ij} + \sum_{i=1}^{N} c_i x_i & \\ \text{s.t.} \quad & X_{ij} \le x_i, \quad X_{ij} \le x_j, & i,j \in [N], i \le j \\ & x_i \in [0,1], \quad X_{ij} \in [0,1], & i,j \in [N], i \le j. \end{aligned}$$

Two other instances of QUBO that are solvable in polynomial time are when: (a) The graph defined by $\boldsymbol{Q}$ is series-parallel (Barahona [11]) and, (b) $\boldsymbol{Q}$ is positive semidefinite and of fixed rank (Allemand et. al. [6]). For an in-depth discussion on polynomial time solvable instances of quadratic binary optimization problems, the reader is referred to the paper of Duan et. al. [45]. For general $\boldsymbol{Q}$ matrices, branch and bound algorithms to solve QUBO problems were proposed by Carter [34] and Pardalos and Rodgers [95]. Beasley [14] developed two heuristic algorithms based on tabu search and simulated annealing while Glover, Kochenberger and Alidaee [55] developed an adaptive memory search heuristic to solve binary quadratic programs. Helmberg and Rendl [69] combined a semidefinite relaxation with a cutting plane technique, and applied it in a branch and bound setting. More recently, second order cone programming has been used to solve QUBO problems (see Kim and Kojima [81], Muramatsu and Suzuki [89], Ghaddar et. al. [53]). Furthermore, the optimization software package CPLEX can efficiently solve problem (1.20) when the objective function in (1.20) is concave, that is the matrix $\boldsymbol{Q}$ is negative semidefinite.

In order to make the quadratic term in (1.20) concave, we make use of the fact that $\boldsymbol{x}^T \text{diag}(\boldsymbol{u})\boldsymbol{x} = \boldsymbol{u}^T\boldsymbol{x}$ for any $\boldsymbol{u} \in \Re^N$, if $x_i \in \{0,1\}$. A simple idea then is to

find a vector $\boldsymbol{u} \in \Re^N$, such that $\boldsymbol{Q} - \text{diag}(\boldsymbol{u})$ is negative semidefinite. Define

$$q_{\boldsymbol{u}}(\boldsymbol{x}; \boldsymbol{c}, \boldsymbol{Q}) = \boldsymbol{x}^T(\boldsymbol{Q} - \text{diag}(\boldsymbol{u}))\boldsymbol{x} + (\boldsymbol{c} + \boldsymbol{u})^T\boldsymbol{x}.$$

Then (1.20) is equivalent to the convex 0-1 quadratic programming problem

$$\max_{\boldsymbol{x}\in\{0,1\}^N} q_u(\boldsymbol{x}; \boldsymbol{c}, \boldsymbol{Q}). \tag{1.21}$$

We can use the mixed integer quadratic programming solver in CPLEX to solve (1.21) for any $\boldsymbol{u} \in \Re^N$ such that $\text{diag}(\boldsymbol{u}) - \boldsymbol{Q} \succeq 0$. However, the CPU time of solving (1.21) can be very different by choosing different $\boldsymbol{u}$. Then the goal is to find a good preprocessing parameter $\boldsymbol{u}$ such that $\text{diag}(\boldsymbol{u}) - \boldsymbol{Q} \succeq 0$.

Billionnet and Elloumi [25] proposed a Quadratic Convex Reformulation (QCR) method to find an "optimal" choice of the parameter $\boldsymbol{u}$ inspired by the semidefinite programming relaxations developed in Körner [82], Shor [111] and Poljak, Rendl and Wolkowicz [98]. In the QCR method of Billionnet and Elloumi [25], the "optimal" preprocessing parameter $\boldsymbol{u}$ was determined by a given matrix $\boldsymbol{Q}$ and a given vector $\boldsymbol{c}$. Notice that the purpose is to find a good parameter $\boldsymbol{u}$ such that $\text{diag}(\boldsymbol{u}) - \boldsymbol{Q} \succeq 0$. A straightforward question is that: when the matrix $\boldsymbol{Q}$ is fixed, and the vector $\boldsymbol{c}$ is random with scenarios lies in the set $\mathcal{C}$, can we still find a common preprocessing parameter $\boldsymbol{u}$ such that problem (1.21) is solved is a reasonable time for all vector $\boldsymbol{c} \in \mathcal{C}$?

In this thesis, we extend the QCR method to solve parametric quadratic unconstrained binary optimization problems:

$$\max_{\boldsymbol{x}\in\{0,1\}^N} \left\{ q(\boldsymbol{x}; \boldsymbol{c}, \boldsymbol{Q}) := \boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c}^T\boldsymbol{x} \right\}, \quad \forall \boldsymbol{c} \in \mathcal{C}, \tag{1.22}$$

where $\boldsymbol{Q}$ is a fixed $N \times N$ symmetric real matrix, and the parameter vector $\boldsymbol{c}$ varies in a set $\mathcal{C}$. We use a Penalized QCR method to find a good common preprocessing parameter $\boldsymbol{u}$ which is "optimal" in certain sense.

1.2 Organization and Contributions

The organization and contributions of this thesis are summarized as follows:

- In Chapter 2, a new probabilistic model for regret in combinatorial optimization is proposed, that is to minimize the WCVaR of regret (1.17). The proposed model incorporates limited probabilistic information on the uncertainty such as the knowledge of the mean, mean absolute deviation or standard deviation while also providing flexibility to model the decision-maker's attitude to regret. In special cases, the probabilistic regret criterion reduces to the traditional minmax regret criterion and the expected objective criterion respectively. To compare with the probabilistic regret model, the problem to minimize the WCVaR of cost is also considered in this chapter.

  We develop tractable formulations to compute the WCVaR of regret and cost for a fixed solution $\boldsymbol{x} \in \mathcal{X}$. The WCVaR of regret is shown to be computable in polynomial time if the deterministic optimization problem is solvable in polynomial time. This generalizes a known result for the interval uncertainty model, where the worst-case regret for a fixed solution $\boldsymbol{x} \in \mathcal{X}$ is known to be computable in polynomial time when the deterministic optimization problem is solvable in polynomial time.

  Then we show that the problem of minimizing the WCVaR of cost can be efficiently solved to optimality as the deterministic linear combinatorial optimization problem. However, since the minmax regret problem is NP-hard, the central problem to minimize the WCVaR of regret is at least NP-hard. To solve it to optimality, mixed integer linear program (MILP) and mixed integer second order cone program (MISOCP) approaches are developed when some partial distributional information for $\tilde{\boldsymbol{c}}$ is given.

- In Chapter 3, we focus on the probabilistic regret model for a problem called the subset selection problem. The polynomial complexity of the minmax regret counterpart of subsect selection in the interval uncertainty has been proved by Averbakh [9] and Conde [37]. We extend the polynomial time result for the minmax regret model to the probabilistic regret model (1.17) and design an efficient polynomial algorithm. The idea behind the algorithm

is furthermore used to solve the distributionally robust k-sum optimization problem.

- In Chapter 4, we generalize the QCR method for a single deterministic QUBO problem to the QUBO problem which has randomness in the linear term of the objective function. We develop a Penalized QCR method to solve this class of problems where the objective function in the dual semidefinite program for the deterministic problem is penalized with a separable term to account for the randomness in the objective. Our computational results indicate that the Penalized QCR method provides a useful preprocessing technique to solve random instances of quadratic unconstrained binary optimization problems.

- In Chapter 6, we finish this thesis with a final conclusion and an overview of possible future work.

Chapter 2

A Probabilistic Regret Model for Linear Combinatorial Optimization

In this chapter, we propose a new probabilistic model for minimizing the anticipated regret in combinatorial optimization problems with distributional uncertainty in the objective coefficients. As a decision criterion, we minimize the worst-case conditional value-at-risk of regret. The proposed model includes the interval data minmax regret as a special case. For the class of combinatorial optimization problems with a compact convex hull representation, a polynomial sized mixed integer linear program (MILP) is formulated when (a) the range and mean are known, and (b) the range, mean and mean absolute deviation are known while a mixed integer second order cone program (MISOCP) is formulated when (c) the range, mean and standard deviation are known.

Structure of the chapter: In Section 2.1, we provide a background on the minmax regret model and motivation for the probabilistic regret model. In Section 2.2, a new probabilistic model for minmax regret in combinatorial optimization is proposed. In Section 2.3, we develop a tractable formulation to compute the WCVaR of regret for a fixed solution $\boldsymbol{x} \in \mathcal{X}$, and show that the WCVaR of regret is computable in polynomial time if the deterministic optimization problem is solvable in polynomial time. In Section 2.4, we formulate conic mixed integer programs to

solve the proposed probabilistic regret model. In Section 2.5, numerical examples for the shortest path problem are provided.

2.1 Background and Motivation

Let $Z(\boldsymbol{c})$ denote the optimal value to a linear combinatorial optimization problem over a feasible region $\mathcal{X} \subseteq \{0,1\}^N$ for the objective coefficient vector $\boldsymbol{c}$:

$$Z(\boldsymbol{c}) = \max \left\{ \boldsymbol{c}^T \boldsymbol{y} \mid \boldsymbol{y} \in \mathcal{X} \subseteq \{0,1\}^N \right\}. \tag{2.1}$$

Assume the vector $\boldsymbol{c}$ is uncertain and let Ω represent a deterministic uncertainty set that captures all the possible realizations of the vector $\boldsymbol{c}$. The value of regret in absolute terms is given by:

$$R(\boldsymbol{x}, \boldsymbol{c}) = Z(\boldsymbol{c}) - \boldsymbol{c}^T \boldsymbol{x}. \tag{2.2}$$

The maximum value of regret for a decision $\boldsymbol{x}$ corresponding to the uncertainty set Ω is given as:

$$\max_{\boldsymbol{c} \in \Omega} R(\boldsymbol{x}, \boldsymbol{c}). \tag{2.3}$$

Savage [107] proposed the use of the following minmax regret model, where the decision $\boldsymbol{x}$ is chosen to minimize the maximum regret over all possible realizations of the uncertainty:

$$\min_{\boldsymbol{x} \in \mathcal{X}} \max_{\boldsymbol{c} \in \Omega} R(\boldsymbol{x}, \boldsymbol{c}). \tag{2.4}$$

The aim of this model is to perform as closely as possible to the optimal course. Since the minmax criterion applied here is to the regret rather than to the cost itself, it is not as pessimistic as the ordinary minmax (absolute robust) approach.

In this chapter, we always assume the vector $\boldsymbol{c}$ lies in an interval uncertainty set Ω, that is each component c_i of the vector $\boldsymbol{c}$ takes a value between a lower bound $\underline{c}_i$ and upper bound $\overline{c}_i$. Let $\Omega_i = [\underline{c}_i, \overline{c}_i]$ for $i = 1, \ldots, N$. The uncertainty set is the Cartesian product of the sets of intervals:

$$\Omega = \Omega_1 \times \Omega_2 \times \ldots \times \Omega_N.$$

In the above interval uncertainty case, for any $\boldsymbol{x} \in \mathcal{X}$, let $S^+_{\boldsymbol{x}}$ denote the scenario in which $c_i = \overline{c}_i$ if $x_i = 0$, and $c_i = \underline{c}_i$ if $x_i = 1$. It is straightforward to see that the scenario $S^+_{\boldsymbol{x}}$ is the worst-case scenario that maximizes the regret in (3.18) for a fixed $\boldsymbol{x} \in \mathcal{X}$. For a deterministic combinatorial optimization problem which is equivalent to its convex hull relaxation, this worst-case scenario can be used to develop compact MILP formulations for the minmax regret problem (2.4) (refer to Yaman et. al. [121] and Kasperski [76]).

The minmax regret models handle support information and assumes that the decision-maker uses the worst-case scenario (in terms of regret) to make the decision. However if additional probabilistic information is known or can be estimated from data, it is natural to incorporate this information into the regret model. To quantify the impact of probabilistic information on regret, consider the graph in Figure 2.1. In this graph, $c_1, c_2, \ldots, c_5$ are the possible traveling time on roads $1, 2, \ldots, 5$. There are three paths connecting node A to node D: $1-4$, $2-5$ and $1-3-5$. Consider a decision-maker who wants to go from node A to node D in the shortest possible time by choosing among the three paths. The mean μ_i and

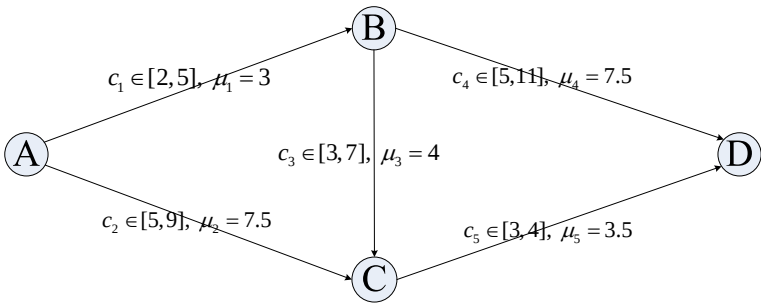


Figure 2.1: Find a Shortest Path from Node A to Node D

range $[\underline{c}_i, \overline{c}_i]$ for each edge i in Figure 2.1 denotes the average time and the range of possible times in hours to traverse the edge. The comparison of the different paths are shown in the following table:

Table 2.1: Comparison of paths

Criterion	Regret			Absolute Robust		Average
Path	$(c_1, c_2, c_3, c_4, c_5)$	Best Path	Max Regret	$(c_1, c_2, c_3, c_4, c_5)$	Max Time	Expected Time
$1-4$	$(5, 5, 3, 11, 3)$	$2-5$	8	$(5, 9, 7, 11, 4)$	16	10.5
$1-3-5$	$(5, 5, 7, 5, 4)$	$2-5$	7	$(5, 9, 7, 11, 4)$	16	10.5
$2-5$	$(2, 9, 3, 5, 4)$	$1-4$	6	$(5, 9, 7, 11, 4)$	13	11

In the minmax regret model, the optimal decision is the path $2-5$ with regret of 6 hours. However, on average this path takes 0.5 hours more than the other two paths. In terms of expected cost, the optimal decision is either of the paths $1-4$ or $1-3-5$. Note that only the range information is used in the minmax regret model, and only mean information is used to minimize the expected cost. Clearly, the choice of an "optimal" path is based on the decision criterion and the available data that guides the decision process. In this chapter, we propose a new probabilistic regret model in combinatorial optimization with uncertainty that incorporates partial distributional information such as the mean and variability of the random coefficients and provides flexibility in modeling the decision-maker's aversion to regret.

2.2 A Probabilistic Regret Model

Let $\tilde{\boldsymbol{c}}$ denote the random objective coefficient vector with a probability distribution P that is itself unknown. P is assumed to lie in the set of distributions $\mathbb{P}(\Omega)$ where Ω is the support of the random vector. In the simplest model, the decision-maker minimizes the anticipated regret in an expected sense:

$$\min_{\boldsymbol{x}\in\mathcal{X}} \sup_{P\in\mathbb{P}(\Omega)} \mathbb{E}_P[R(\boldsymbol{x}, \tilde{\boldsymbol{c}})]. \tag{2.5}$$

Model (2.5) includes two important subcases: (a) $\mathbb{P}(\Omega)$ is the set of all probability distributions with support Ω. In this case (2.5) reduces to the standard minmax regret model (2.4). And (b) The complete distribution is given with $\mathbb{P} = \{P\}$. In

this case (2.5) reduces to solving the deterministic optimization problem where the random objective is replaced with the mean vector $\boldsymbol{\mu}$, since

$$\underset{\boldsymbol{x}\in\mathcal{X}}{\operatorname{argmin}}\,\mathbb{E}_P[Z(\tilde{\boldsymbol{c}})-\tilde{\boldsymbol{c}}^T\boldsymbol{x}]=\underset{\boldsymbol{x}\in\mathcal{X}}{\operatorname{argmin}}\left(\mathbb{E}_P[Z(\tilde{\boldsymbol{c}})]-\boldsymbol{\mu}^T\boldsymbol{x}\right)=\underset{\boldsymbol{x}\in\mathcal{X}}{\operatorname{argmax}}\,\boldsymbol{\mu}^T\boldsymbol{x}.$$

Formulation (2.5) however does not capture the degree of regret aversion. Furthermore, as long as the mean vector is fixed, the optimal decision in (2.5) is the optimal solution to the deterministic problem with the mean objective. Thus the solution is insensitive to other distributional information such as variability. To address this, we propose use of the conditional value-at-risk measure that has been gaining popularity in the risk management literature. We propose the WCVaR of regret as a decision criterion in combinatorial optimization problems. By the definition 1.3 of WCVaR and Lemma 1.4, the central problem of interest to solve in this chapter is:

$$\min_{\boldsymbol{x}\in\mathcal{X}}\mathrm{WCVaR}_\alpha(R(\boldsymbol{x},\tilde{\boldsymbol{c}}))=\min_{\boldsymbol{x}\in\mathcal{X}}\inf_{v\in\Re}\left(v+\frac{1}{1-\alpha}\sup_{P\in\mathbb{P}(\Omega)}\mathbb{E}_P\left[R(\boldsymbol{x},\tilde{\boldsymbol{c}})-v\right]^+\right). \quad (2.6)$$

2.2.1 Differences between the Proposed Regret Model and the Existing Newsvendor Regret Model

As introduced in Chapter 1, we consider the distribution set $\mathbb{P}(\Omega)$ in the marginal distribution model and the marginal moment model. The moment representation of uncertainty in distributions has been used in the minmax regret newsvendor problem [124, 96]. A newsvendor needs to choose an order quantity q of a product before the exact value of demand is known by balancing the costs of under-ordering and over-ordering. The random demand is represented by $\tilde{d}$ with a probability distribution P. The unit selling price is p, the unit cost is c and the salvage value for any unsold product is 0. A risk neutral firm chooses its quantity to maximize its expected profit:

$$\max_{q\geq 0}\left(p\mathbb{E}_P[\min(q,\tilde{d})]-cq\right),$$

where $\min(q,\tilde{d})$ is the actual quantity of units sold which depends on the demand realization. In the minmax regret version of this problem studied in [124, 96], the

newsvendor chooses the order quantity where the demand distribution is not exactly known. The demand distribution is assumed to belong to a set of probability measures $P \in \mathbb{P}$ typically characterized with moment information. The objective is to minimize the maximum loss in profit from not knowing the full distribution:

$$\min_{q\geq 0} \max_{P\in\mathbb{P}} \left[\max_{s\geq 0} \left(p\mathbb{E}_P[\min(s,\tilde{d})] - cs \right) - \left(p\mathbb{E}_P[\min(q,\tilde{d})] - cq \right) \right].$$

Yue et. al. [124] solved this model analytically where only the mean and variance of demand are known. Roels and Perakis [96] generalized this model to incorporate additional moments and information on the shape of the demand. On the other hand, if the demand is known with certainty, the optimal order quantity is exactly the demand. The maximum profit would be $(p-c)\tilde{d}$ and the regret model as proposed in this chapter is:

$$\min_{q\geq 0} \inf_{v\in\Re} \left(v + \frac{1}{1-\alpha} \sup_{P\in\mathbb{P}(\Omega)} \mathbb{E}_P \left[(p-c)\tilde{d} - \left(p\min(q,\tilde{d}) - cq \right) - v \right]^+ \right),$$

where α is the parameter that captures aversion to regret. There are two major differences between the minmax regret newsvendor model in [124, 96] and the regret model proposed in this chapter. The first difference is that in [124, 96] the newsvendor minimizes the maximum ex-ante regret (with respect to distributions) of not knowing the right distribution, while in this chapter, the decision-maker minimizes the ex-post regret (with respect to cost coefficient realizations) of not knowing the right objective coefficients. The second difference is that the newsvendor problem deals with a single demand variable. However in the multi-dimensional case, the marginal model forms the natural extension and is a more tractable formulation.

2.2.2 Relation to the Standard Minmax Regret Model

The new probabilistic regret model can be related to the standard minmax regret model. In the marginal moment model, if only the range information of each random variable $\tilde{c}_i$ is given, then the WCVaR of regret reduces to the maximum regret. Consider the random vector whose distribution is a Dirac measure $\delta_{\tilde{\boldsymbol{c}}(\boldsymbol{x})}$

with $\hat{c}_i(\boldsymbol{x}) = \overline{c}_i(1-x_i) + \underline{c}_i x_i$ for $i \in [N]$. Then WCVaR of the regret satisfies:

$$\begin{aligned}\inf_{v\in\Re}\left(v + \frac{1}{1-\alpha}\sup_{P\in\mathbb{P}(\Omega)}\mathbb{E}_P[R(\boldsymbol{x},\tilde{\boldsymbol{c}})-v]^+\right) &\geq \inf_{v\in\Re}\left(v + \frac{1}{1-\alpha}\mathbb{E}_{\delta_{\hat{\boldsymbol{c}}}}[R(\boldsymbol{x},\tilde{\boldsymbol{c}})-v]^+\right)\\ &= \inf_{v\in\Re}\left(v + \frac{1}{1-\alpha}[R(\boldsymbol{x},\hat{\boldsymbol{c}})-v]^+\right)\\ &= R(\boldsymbol{x},\hat{\boldsymbol{c}})\\ &= \max_{\boldsymbol{c}\in\Omega} R(\boldsymbol{x},\boldsymbol{c}).\end{aligned}$$

The last equality is valid since $\hat{\boldsymbol{c}}(\boldsymbol{x})$ is the worst-case scenario for a given $\boldsymbol{x} \in \mathcal{X}$. Moreover, the WCVaR of the regret cannot be larger than the maximum value of regret. Hence, they are equal in this case. When $\alpha = 0$, problem (2.6) reduces to minimizing the worst-case expected regret,

$$\min_{\boldsymbol{x}\in\mathcal{X}}\sup_{P\in\mathbb{P}(\Omega)}\mathbb{E}_P[R(\boldsymbol{x},\tilde{\boldsymbol{c}})].$$

On the other hand, as α converges to 1, $\text{WCVaR}_\alpha(R(\boldsymbol{x},\tilde{\boldsymbol{c}}))$ converges to the maximum regret $\max_{\boldsymbol{c}\in\Omega} R(\boldsymbol{x},\boldsymbol{c})$, and problem (2.6) reduces to the traditional interval uncertainty minmax regret model. This implies that the problem of minimizing the WCVaR of the regret in this probabilistic model is NP-hard since the minmax regret problem is NP-hard [10]. The parameter α allows for the flexibility to vary the degree of regret aversion.

If a decision $\boldsymbol{x}_1$ is preferred to decision $\boldsymbol{x}_2$ for each realization of the uncertainty, it is natural to conjecture that $\boldsymbol{x}_1$ is preferred to $\boldsymbol{x}_2$ in the regret model. The following lemma validates this monotonicity property for the chosen criterion.

Lemma 2.1. *For two decisions $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{X}$, if $\boldsymbol{x}_1$ dominates $\boldsymbol{x}_2$ in each realization of the uncertainty, i.e. $\boldsymbol{c}^T\boldsymbol{x}_1 \geq \boldsymbol{c}^T\boldsymbol{x}_2$ for all $\boldsymbol{c} \in \Omega$, then the decision $\boldsymbol{x}_1$ is preferred to $\boldsymbol{x}_2$, i.e.* $\text{WCVaR}_\alpha(R(\boldsymbol{x}_1,\tilde{\boldsymbol{c}})) \leq \text{WCVaR}_\alpha(R(\boldsymbol{x}_2,\tilde{\boldsymbol{c}}))$.

Proof. Since $\boldsymbol{c}^T\boldsymbol{x}_1 \geq \boldsymbol{c}^T\boldsymbol{x}_2$ for all $\boldsymbol{c} \in \Omega$,

$$R(\boldsymbol{x}_1,\boldsymbol{c}) = \max_{\boldsymbol{y}\in\mathcal{X}}\boldsymbol{c}^T\boldsymbol{y} - \boldsymbol{c}^T\boldsymbol{x}_1 \leq \max_{\boldsymbol{y}\in\mathcal{X}}\boldsymbol{c}^T\boldsymbol{y} - \boldsymbol{c}^T\boldsymbol{x}_2 = R(\boldsymbol{x}_2,\boldsymbol{c}), \quad \forall \boldsymbol{c}\in\Omega.$$

Thus $[R(\boldsymbol{x}_1,\boldsymbol{c})-v]^+ \leq [R(\boldsymbol{x}_2,\boldsymbol{c})-v]^+, \ \ \forall \boldsymbol{c}\in\Omega, v\in\Re$. Hence for any distribution $P \in \mathbb{P}$, $\mathbb{E}_P[R(\boldsymbol{x}_1,\tilde{\boldsymbol{c}})-v]^+ \leq \mathbb{E}_P[R(\boldsymbol{x}_2,\tilde{\boldsymbol{c}})-v]^+$. This implies that

$$\sup_{P\in\mathbb{P}}\mathbb{E}_P[R(\boldsymbol{x}_1,\tilde{\boldsymbol{c}})-v]^+ \leq \sup_{P\in\mathbb{P}}\mathbb{E}_P[R(\boldsymbol{x}_2,\tilde{\boldsymbol{c}})-v]^+.$$

Therefore,

$$\inf_{v\in\Re}\Big(v+\frac{1}{1-\alpha}\sup_{P\in\mathbb{P}}\mathbb{E}_P[R(\boldsymbol{x}_1,\tilde{\boldsymbol{c}})-v]^+\Big)\leq\inf_{v\in\Re}\Big(v+\frac{1}{1-\alpha}\sup_{P\in\mathbb{P}}\mathbb{E}_P[R(\boldsymbol{x}_2,\tilde{\boldsymbol{c}})-v]^+\Big),$$

that is $\text{WCVaR}_\alpha(R(\boldsymbol{x}_1,\tilde{\boldsymbol{c}}))\leq\text{WCVaR}_\alpha(R(\boldsymbol{x}_2,\tilde{\boldsymbol{c}}))$. □

2.3 Computation of the WCVaR of Regret and Cost

In this section, we compute the WCVaR of regret for a fixed $\boldsymbol{x}\in\mathcal{X}$ in the marginal distribution and marginal moment model. This is motivated by bounds in the Project Evaluation and Review Technique (PERT) networks that were proposed by Meilijson and Nadas [88] and later extended in the works of Klein Haneveld [66], Weiss [120], Birge and Maddox [30] and Bertsimas et. al. [21]. In a PERT network, let $[N]$ represent the set of activities. Each activity $i\in[N]$ is associated with a random activity time $\tilde{c}_i$ and marginal distribution P_i. Meilijson and Nadas [88] computed the worst-case expected project tardiness $\sup_{P\in\mathbb{P}(P_1,\ldots,P_N)}\mathbb{E}_P[Z(\tilde{\boldsymbol{c}})-v]^+$ where $Z(\boldsymbol{c})$ denotes the time to complete the project and v denotes a deadline for the project. Their approach can be summarized as follows. For all $\boldsymbol{d}\in\Re^N$ and $\boldsymbol{c}\in\Omega$:

$$\begin{aligned}[Z(\boldsymbol{c})-v]^+ &= \left[\max_{\boldsymbol{y}\in\mathcal{X}}(\boldsymbol{d}+\boldsymbol{c}-\boldsymbol{d})^T\boldsymbol{y}-v\right]^+\\ &\leq \left[\max_{\boldsymbol{y}\in\mathcal{X}}\boldsymbol{d}^T\boldsymbol{y}-v\right]^+ + \left[\max_{\boldsymbol{y}\in\mathcal{X}}(\boldsymbol{c}-\boldsymbol{d})^T\boldsymbol{y}\right]^+\\ &\leq [Z(\boldsymbol{d})-v]^+ + \sum_{i=1}^N[c_i-d_i]^+.\end{aligned}$$

Taking expectation with respect to a distribution $P\in\mathbb{P}(P_1,\ldots,P_N)$ and minimizing over $\boldsymbol{d}\in\Re^N$ gives the bound:

$$\mathbb{E}_P[Z(\tilde{\boldsymbol{c}})-v]^+\leq\inf_{\boldsymbol{d}\in\Re^N}\Big([Z(\boldsymbol{d})-v]^+ + \sum_{i=1}^N\mathbb{E}_{P_i}[\tilde{c}_i-d_i]^+\Big),\quad\forall P\in\mathbb{P}(P_1,\ldots,P_N).$$

Meilijson and Nadas [88] constructed a multivariate probability distribution that is consistent with the marginal distributions such that the upper bound is attained.

This leads to their main observation that the worst-case expected project tardiness is obtained by solving the following convex minimization problem:

$$\sup_{P\in\mathbb{P}(P_1,\ldots,P_N)} \mathbb{E}_P[Z(\tilde{\boldsymbol{c}})-v]^+ = \inf_{\boldsymbol{d}\in\Re^N}\Big([Z(\boldsymbol{d})-v]^+ + \sum_{i=1}^{N}\mathbb{E}_{P_i}[\tilde{c}_i-d_i]^+\Big). \tag{2.7}$$

With partial marginal distribution information, Klein Haneveld [66], Birge and Maddox [30] and Bertsimas et al. [21] extended the convex formulation of the worst-case expected project tardiness to:

$$\sup_{P\in\mathbb{P}(\mathbb{P}_1,\ldots,\mathbb{P}_N)} \mathbb{E}_P[Z(\tilde{\boldsymbol{c}})-v]^+ = \inf_{\boldsymbol{d}\in\Re^N}\Big([Z(\boldsymbol{d})-v]^+ + \sum_{i=1}^{N}\sup_{P_i\in\mathbb{P}_i}\mathbb{E}_{P_i}[\tilde{c}_i-d_i]^+\Big). \tag{2.8}$$

Klein Haneveld [66] estimated a project deadline v that balances the expected project tardiness with respect to the most unfavorable distribution and the cost of choosing the deadline for the project. This can be formulated as a two stage recourse problem:

$$\inf_{v\in\Re}\left(v+\frac{1}{1-\alpha}\sup_{P\in\mathbb{P}(\mathbb{P}_1,\ldots,\mathbb{P}_N)}\mathbb{E}_P\left[Z(\tilde{\boldsymbol{c}})-v\right]^+\right). \tag{2.9}$$

where $\alpha\in(0,1)$ is the tradeoff parameter between the two costs. Formulation (2.9) is clearly equivalent to estimating the worst-case conditional value-at-risk of the project completion time. We extend these results to the regret framework in the following section.

2.3.1 WCVaR of Regret

To compute the WCVaR of regret, we first consider the subproblem

$$\sup_{P\in\mathbb{P}(\Omega)}\mathbb{E}_P\left[Z(\tilde{\boldsymbol{c}})-\tilde{\boldsymbol{c}}^T\boldsymbol{x}-v\right]^+$$

in the central problem (2.6). The proof of Theorem 2.2 is inspired from proof techniques in Doan and Natarajan [44] and Natarajan et. al. [91].

Theorem 2.2. *For each $i\in[N]$, assume that the marginal distribution P_i of the continuously distributed random variable $\tilde{c}_i$ with support $\Omega_i=[\underline{c}_i,\overline{c}_i]$ is given. For*

$\boldsymbol{x} \in \mathcal{X} \subseteq \{0,1\}^N$ and $v \geq 0$, define

$$\phi(\boldsymbol{x}, v) := \sup_{P \in \mathbb{P}(P_1,\ldots,P_N)} \mathbb{E}_P \left[Z(\tilde{\boldsymbol{c}}) - \tilde{\boldsymbol{c}}^T \boldsymbol{x} - v\right]^+,$$

and

$$\bar{\phi}(\boldsymbol{x}, v) := \min_{\boldsymbol{d} \in \Omega} \Big(\left[Z(\boldsymbol{d}) - \boldsymbol{d}^T \boldsymbol{x} - v\right]^+ + (\boldsymbol{d} - \boldsymbol{\mu})^T \boldsymbol{x} + \sum_{i=1}^{N} \mathbb{E}_{P_i}[\tilde{c}_i - d_i]^+ \Big).$$

Then $\phi(\boldsymbol{x}, v) = \bar{\phi}(\boldsymbol{x}, v)$.

Proof. Define

$$\begin{aligned} \phi_0(\boldsymbol{x}, v) &:= \sup_{P \in \mathbb{P}(P_1,\ldots,P_N)} \mathbb{E}_P \left[\max\left(Z(\tilde{\boldsymbol{c}}), \tilde{\boldsymbol{c}}^T \boldsymbol{x} + v\right)\right] \\ \bar{\phi}_0(\boldsymbol{x}, v) &:= \min_{\boldsymbol{d} \in \Omega} \Big(\max\left(Z(\boldsymbol{d}), \boldsymbol{d}^T \boldsymbol{x} + v\right) + \sum_{i=1}^{N} \mathbb{E}_{P_i}[\tilde{c}_i - d_i]^+ \Big). \end{aligned}$$

Since $\max\left(Z(\boldsymbol{c}), \boldsymbol{c}^T \boldsymbol{x} + v\right) = [Z(\boldsymbol{c}) - \boldsymbol{c}^T \boldsymbol{x} - v]^+ + \boldsymbol{c}^T \boldsymbol{x} + v$, to prove $\phi(\boldsymbol{x}, v) = \bar{\phi}(\boldsymbol{x}, v)$ is equivalent to proving that $\phi_0(\boldsymbol{x}, v) = \bar{\phi}_0(\boldsymbol{x}, v)$.

Step 1: Prove that $\phi_0(\boldsymbol{x}, v) \leq \bar{\phi}_0(\boldsymbol{x}, v)$.

For any $\boldsymbol{c} \in \Omega = \Omega_1 \times \Omega_2 \times \ldots \times \Omega_N$, the following holds:

$$\begin{aligned} \max\left(Z(\boldsymbol{c}), \boldsymbol{c}^T \boldsymbol{x} + v\right) &= \max\left(\max_{\boldsymbol{y} \in \mathcal{X}} (\boldsymbol{c} - \boldsymbol{d} + \boldsymbol{d})^T \boldsymbol{y}, (\boldsymbol{c} - \boldsymbol{d} + \boldsymbol{d})^T \boldsymbol{x} + v \right) \\ &\leq \max\left(\max_{\boldsymbol{y} \in \mathcal{X}} \boldsymbol{d}^T \boldsymbol{y} + \max_{\boldsymbol{y} \in \mathcal{X}} (\boldsymbol{c} - \boldsymbol{d})^T \boldsymbol{y}, \boldsymbol{d}^T \boldsymbol{x} + v + (\boldsymbol{c} - \boldsymbol{d})^T \boldsymbol{x} \right) \\ &\leq \max\left(Z(\boldsymbol{d}) + \sum_{i=1}^{n} [c_i - d_i]^+, \boldsymbol{d}^T \boldsymbol{x} + v + \sum_{i=1}^{n} [c_i - d_i]^+ \right) \\ &= \max\left(Z(\boldsymbol{d}), \boldsymbol{d}^T \boldsymbol{x} + v\right) + \sum_{i=1}^{n} [c_i - d_i]^+. \end{aligned}$$

Taking expectation with respect to the probability measure $P \in \mathbb{P}(P_1, \ldots, P_N)$ and minimum with respect to $\boldsymbol{d} \in \Omega$, we get

$$\mathbb{E}_P \left[\max\left(Z(\tilde{\boldsymbol{c}}), \tilde{\boldsymbol{c}}^T \boldsymbol{x} + v\right)\right] \leq \bar{\phi}_0(\boldsymbol{x}, v), \quad \forall\ P \in \mathbb{P}(P_1, \ldots, P_N).$$

Taking supremum with respect to $P \in \mathbb{P}(P_1, \ldots, P_N)$, implies $\phi_0(\boldsymbol{x}, v) \leq \bar{\phi}_0(\boldsymbol{x})$.

Step 2: Prove that $\phi_0(\boldsymbol{x}, v) \geq \bar{\phi}_0(\boldsymbol{x}, v)$.

First, we claim that

$$\bar{\phi}_0(\boldsymbol{x}, v) = \min_{\boldsymbol{d} \in \Re^N} \Big(\max\big(Z(\boldsymbol{d}), \boldsymbol{d}^T\boldsymbol{x} + v\big) + \sum_{i=1}^{N} \mathbb{E}_{P_i}[\tilde{c}_i - d_i]^+ \Big). \tag{2.10}$$

Since for all $\boldsymbol{d} \in \Re^N \setminus \Omega$, we can choose $\boldsymbol{d}^* \in \Omega$:

$$d_i^* = \begin{cases} d_i, \text{ if } d_i \in [\underline{c}_i, \bar{c}_i], \\ \bar{c}_i, \text{ if } d_i > \bar{c}_i, \\ \underline{c}_i, \text{ if } d_i < \underline{c}_i. \end{cases}$$

such that the objective value will be lesser than or equal to the objective value at $\boldsymbol{d}$. The reason is that if $d_i > \bar{c}_i$, by setting $d_i^* = \bar{c}_i$, the second term of the objective function in (2.10) will not change while the first term will decrease or stay constant. If $d_i < \underline{c}_i$, by setting $d_i^* = \underline{c}_i$, the second term will decrease by $\underline{c}_i - d_i$, and the first term will increase by at most $\underline{c}_i - d_i$. Hence $\bar{\phi}_0(\boldsymbol{x}, v)$ can be expressed as:

$$\begin{aligned} \bar{\phi}_0(\boldsymbol{x}, v) \quad = \min_{\boldsymbol{d}, t} \quad & t + \sum_{i=1}^{N} \mathbb{E}_{P_i}[\tilde{c}_i - d_i]^+ \\ \text{s.t.} \quad & t \geq \boldsymbol{d}^T\boldsymbol{y}, \quad \forall\, \boldsymbol{y} \in \mathcal{X} \\ & t \geq \boldsymbol{d}^T\boldsymbol{x} + v. \end{aligned} \tag{2.11}$$

For a fixed $\boldsymbol{x} \in \mathcal{X}$, (2.11) is a convex programming problem in decision variables $\boldsymbol{d}$ and t. The Karush-Kuhn-Tucker (KKT) conditions for (2.11) are given as follows:

$$\lambda(\boldsymbol{y}) \geq 0, t \geq \boldsymbol{d}^T\boldsymbol{y}, \forall \boldsymbol{y} \in \mathcal{X}, \text{ and } s \geq 0, t \geq \boldsymbol{d}^T\boldsymbol{x} + v \tag{2.12a}$$

$$\sum_{\boldsymbol{y} \in \mathcal{X}} \lambda(\boldsymbol{y}) + s = 1 \tag{2.12b}$$

$$\lambda(\boldsymbol{y}) \Big(\max\big(Z(\boldsymbol{d}), \boldsymbol{d}^T\boldsymbol{x} + v\big) - \boldsymbol{d}^T\boldsymbol{y} \Big) = 0, \ \forall\, \boldsymbol{y} \in \mathcal{X} \tag{2.12c}$$

$$s \Big(\max\big(Z(\boldsymbol{d}), \boldsymbol{d}^T\boldsymbol{x} + v\big) - \boldsymbol{d}^T\boldsymbol{x} - v \Big) = 0 \tag{2.12d}$$

$$P(\tilde{c}_i \geq d_i) = \sum_{\boldsymbol{y} \in \mathcal{X} : y_i = 1} \lambda(\boldsymbol{y}) + s x_i. \tag{2.12e}$$

There exists an optimal $\boldsymbol{d}$ in the compact set Ω and optimal $t = \max\left(Z(\boldsymbol{d}), \boldsymbol{d}^T\boldsymbol{x} + v\right)$ to problem (2.11). Under the standard Slater's conditions for strong duality in convex optimization, there exist dual variables $s, \lambda(\boldsymbol{y})$ such that these optimal $\boldsymbol{d}, t, s, \lambda(\boldsymbol{y})$ satisfy the KKT conditions. For the rest of the proof, we let $\boldsymbol{d}, t, s, \lambda(\boldsymbol{y})$ denote the optimal solution that satisfy the KKT conditions. Let $f_i(\cdot)$ be the probability density function associated with P_i. We construct a distribution $\bar{P}$ as follows:

(a) Generate a random vector $\tilde{\boldsymbol{y}}$ which takes the value $\boldsymbol{y} \in \mathcal{X}$ with probability $\lambda(\boldsymbol{y})$ if $\boldsymbol{y} \neq \boldsymbol{x}$, and takes the value $\boldsymbol{x} \in \mathcal{X}$ with probability s. Note that $\lambda(\boldsymbol{x}) = 0$ from the KKT condition (2.12c).

(b) Define the set $I_1 = \{i \in [N] : \underline{c}_i < d_i < \bar{c}_i\}$ and $I_2 = [N] \setminus I_1$. For $i \in I_1$, generate the random variable $\tilde{c}_i$ with the conditional probability density function

$$\bar{f}_i(c_i|\tilde{\boldsymbol{y}} = \boldsymbol{y}) = \begin{cases} \frac{1}{P(\tilde{c}_i \geq d_i)}\mathbb{I}_{[d_i,\bar{c}_i]}(c_i)f_i(c_i) & \text{if } y_i = 1, \\ \frac{1}{P(\tilde{c}_i < d_i)}\mathbb{I}_{[\underline{c}_i,d_i)}(c_i)f_i(c_i) & \text{if } y_i = 0, \end{cases}$$

and for $i \in I_2$ generate the random variable $\tilde{c}_i$ with the conditional probability density function $\bar{f}_i(c_i|\tilde{\boldsymbol{y}} = \boldsymbol{y}) = f_i(c_i)$.

For $i \in I_2$, the probability density function for each $\tilde{c}_i$ under $\bar{P}$ is $\bar{f}_i(c_i) = f_i(c_i)$. For $i \in I_1$, the probability density function is:

$$\begin{aligned}
\bar{f}_i(c_i) =& \sum_{\boldsymbol{y}\in\mathcal{X}} \lambda(\boldsymbol{y})\bar{f}_i(c_i|\tilde{\boldsymbol{y}} = \boldsymbol{y}) + s \cdot \bar{f}_i(c_i|\tilde{\boldsymbol{y}} = \boldsymbol{x}) \\
=& \sum_{\boldsymbol{y}\in\mathcal{X}:y_i=1} \frac{\lambda(\boldsymbol{y})}{P(\tilde{c}_i \geq d_i)}\mathbb{I}_{[d_i,\bar{c}_i]}(c_i)f_i(c_i) + \frac{sx_i}{P(\tilde{c}_i \geq d_i)}\mathbb{I}_{[d_i,\bar{c}_i]}(c_i)f_i(c_i) \\
&+ \sum_{\boldsymbol{y}\in\mathcal{X}:y_i=0} \frac{\lambda(\boldsymbol{y})}{P(\tilde{c}_i < d_i)}\mathbb{I}_{[\underline{c}_i,d_i)}(c_i)f_i(c_i) + \frac{s(1-x_i)}{P(\tilde{c}_i < d_i)}\mathbb{I}_{[\underline{c}_i,d_i)}(c_i)f_i(c_i) \\
=& \mathbb{I}_{[d_i,\bar{c}_i]}(c_i)f_i(c_i) + \mathbb{I}_{[\underline{c}_i,d_i)}(c_i)f_i(c_i) \\
=& f_i(c_i).
\end{aligned}$$

The probability density function constructed hence belongs to $\mathbb{P}(P_1, \ldots, P_n)$. Therefore,

$$\begin{aligned}
\phi_0(\boldsymbol{x}, v) \geq & \mathbb{E}_{\bar{P}}\left[\max\left(Z(\tilde{\boldsymbol{c}}), \tilde{\boldsymbol{c}}^T\boldsymbol{x} + v\right)\right] \\
= & \sum_{\boldsymbol{y}\in\mathcal{X}} \lambda(\boldsymbol{y}) \mathbb{E}_{\bar{P}}\left[\max\left(Z(\tilde{\boldsymbol{c}}), \tilde{\boldsymbol{c}}^T\boldsymbol{x} + v\right) \mid \tilde{\boldsymbol{y}} = \boldsymbol{y}\right] \\
& + s\mathbb{E}_{\bar{P}}\left[\max\left(Z(\tilde{\boldsymbol{c}}), \tilde{\boldsymbol{c}}^T\boldsymbol{x} + v\right) \mid \tilde{\boldsymbol{y}} = \boldsymbol{x}\right] \\
\geq & \sum_{\boldsymbol{y}\in\mathcal{X}} \lambda(\boldsymbol{y}) \mathbb{E}_{\bar{P}}\left[Z(\tilde{\boldsymbol{c}}) \mid \tilde{\boldsymbol{y}} = \boldsymbol{y}\right] + s\mathbb{E}_{\bar{P}}\left[\tilde{\boldsymbol{c}}^T\boldsymbol{x} + v \mid \tilde{\boldsymbol{y}} = \boldsymbol{x}\right] \\
\geq & \sum_{\boldsymbol{y}\in\mathcal{X}} \lambda(\boldsymbol{y}) \mathbb{E}_{\bar{P}}\left[\tilde{\boldsymbol{c}}^T\boldsymbol{y} \mid \tilde{\boldsymbol{y}} = \boldsymbol{y}\right] + s\mathbb{E}_{\bar{P}}\left[\tilde{\boldsymbol{c}}^T\boldsymbol{x} + v \mid \tilde{\boldsymbol{y}} = \boldsymbol{x}\right] \\
= & \sum_{\boldsymbol{y}\in\mathcal{X}: y_i = 1} \lambda(\boldsymbol{y}) \Big(\sum_{i\in I_1} \int c_i \frac{1}{P(\tilde{c}_i \geq d_i)} \mathbb{I}_{[d_i, \bar{c}_i]}(c_i) f_i(c_i) dc_i + \sum_{i\in I_2} \int c_i f_i(c_i) dc_i \Big) \\
& + s\Big(\sum_{i\in I_1} \int c_i x_i \frac{1}{P(\tilde{c}_i \geq d_i)} \mathbb{I}_{[d_i, \bar{c}_i]}(c_i) f_i(c_i) dc_i + \sum_{i\in I_2} \int c_i x_i f_i(c_i) dc_i \Big) + sv \\
= & \sum_{i\in I_1} \int c_i \mathbb{I}_{[d_i, \bar{c}_i]}(c_i) f_i(c_i) dc_i + \sum_{i\in I_2} \int P(\tilde{c}_i \geq d_i) c_i f_i(c_i) dc_i + sv. \text{ (by (2.12e))}
\end{aligned}$$

Since $P(\tilde{c}_i \geq d_i) = 1$ or 0 for $i \in I_2$, hence

$$\int P(\tilde{c}_i \geq d_i) c_i f_i(c_i) dc_i = \int c_i \mathbb{I}_{[d_i, \bar{c}_i]}(c_i) f_i(c_i) dc_i, \forall i \in I_2.$$

Then, we obtain

$$\begin{aligned}
\phi_0(\boldsymbol{x}, v) \geq & \sum_{i=1}^{N} \int c_i \mathbb{I}_{[d_i, \bar{c}_i]}(c_i) f_i(c_i) dc_i + sv \\
= & \sum_{i=1}^{N} \int (c_i - d_i) \mathbb{I}_{[d_i, \bar{c}_i]}(c_i) f_i(c_i) dc_i + \sum_{i=1}^{N} d_i \int \mathbb{I}_{[d_i, \bar{c}_i]}(c_i) f_i(c_i) dc_i + sv \\
= & \sum_{i=1}^{N} \mathbb{E}_{P_i}[\tilde{c}_i - d_i]^+ + \sum_{i=1}^{N} d_i \Big(\sum_{\boldsymbol{y}\in\mathcal{X}: y_i = 1} \lambda(\boldsymbol{y}) + s x_i \Big) + sv \quad \text{(by (2.12e))} \\
= & \sum_{i=1}^{N} \mathbb{E}_{P_i}[\tilde{c}_i - d_i]^+ + \sum_{\boldsymbol{y}\in\mathcal{X}} \lambda(\boldsymbol{y}) \boldsymbol{d}^T\boldsymbol{y} + s(\boldsymbol{d}^T\boldsymbol{x} + v) \\
= & \sum_{i=1}^{N} \mathbb{E}_{P_i}[\tilde{c}_i - d_i]^+ + \sum_{\boldsymbol{y}\in\mathcal{X}} \lambda(\boldsymbol{y}) \max\left(Z(\boldsymbol{d}), \boldsymbol{d}^T\boldsymbol{x} + v\right) + s(\boldsymbol{d}^T\boldsymbol{x} + v) \text{ (by (2.12c))}
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=1}^{N} \mathbb{E}_{P_i}[\tilde{c}_i - d_i]^+ + (1-s)\max\left(Z(\boldsymbol{d}), \boldsymbol{d}^T\boldsymbol{x} + v\right) + s(\boldsymbol{d}^T\boldsymbol{x} + v) \text{ (by (2.12b))} \\
&= \sum_{i=1}^{N} \mathbb{E}_{P_i}[\tilde{c}_i - d_i]^+ + \max\left(Z(\boldsymbol{d}), \boldsymbol{d}^T\boldsymbol{x} + v\right) \quad \text{(by (2.12d))} \\
&= \bar{\phi}_0(\boldsymbol{x}, v).
\end{aligned}
$$

□

It is useful to contrast the regret bound in Theorem 2.2 with the earlier bound of Meilijson and Nadas [88] in (2.7). In Theorem 2.2, the worst-case joint distribution depends on the solution $\boldsymbol{x} \in \mathcal{X}$ and the scalar v. The worst-case joint distribution in Formulation (2.7) however depends on the scalar v only. The proof of Theorem 2.2 can be extended directly to discrete marginal distributions by replacing the integrals with summations and using linear programming duality. This result generalizes to the marginal moment model and piecewise linear convex functions as illustrated in the next theorem. The proof of Theorem 2.3 is inspired from proof techniques in Bertsimas et. al. [21] and Natarajan et. al. [91].

Theorem 2.3. *For $\boldsymbol{x} \in \mathcal{X} \subseteq \{0,1\}^N$, consider the marginal moment model:*

$$\mathbb{P}_i = \{P_i \mid \mathbb{E}_{P_i}[f_{ik}(\tilde{c}_i)] = m_{ik}, k \in [K_i], \mathbb{E}_{P_i}[\mathbb{I}_{[\underline{c}_i, \bar{c}_i]}(\tilde{c}_i)] = 1\},$$

where $\mathbb{I}_{[\underline{c}_i, \bar{c}_i]}(c_i) = 1$ if $\underline{c}_i \leq c_i \leq \bar{c}_i$ and 0 otherwise. Assume that the moments lie interior to the set of feasible moment vectors. Define

$$\phi(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{b}) := \sup_{P \in \mathbb{P}(\mathbb{P}_1, \ldots, \mathbb{P}_N)} \mathbb{E}_P\left[g\left(Z(\tilde{\boldsymbol{c}}) - \tilde{\boldsymbol{c}}^T\boldsymbol{x}\right)\right] \tag{2.13}$$

where $g(\cdot)$ is a non-decreasing piecewise linear convex function defined by

$$g(z) = \max_{j \in [J]} \left(a_j z + b_j\right),$$

with $0 \leq\ a_1 < a_2 < \ldots < a_J$. Let

$$
\begin{aligned}
&\bar{\phi}(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{b}) \qquad\qquad (2.14) \\
= &\min_{\boldsymbol{d}_1, \ldots, \boldsymbol{d}_J \in \Omega} \left\{ g\left(Z(\boldsymbol{d}_j) - \boldsymbol{d}_j^T\boldsymbol{x}\right) + \sum_{i=1}^{N} \sup_{P_i \in \mathbb{P}_i} \mathbb{E}_{P_i}\left[\max_{j \in [J]} a_j \left([\tilde{c}_i - d_{ji}]^+ - [\tilde{c}_i - d_{ji}]x_i\right)\right] \right\}.
\end{aligned}
$$

Then $\phi(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{b}) = \bar{\phi}(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{b})$.

Proof. Step 1: Prove that $\phi(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{b}) \leq \bar{\phi}(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{b})$.

For any $\boldsymbol{c} \in \Omega$, and $\boldsymbol{d}_1, \ldots, \boldsymbol{d}_J \in \Omega$, the following holds:

$$\begin{aligned} g(Z(\boldsymbol{c}) - \boldsymbol{c}^T\boldsymbol{x}) &= \max_{j\in[J]} \left[a_j \left(\max_{\boldsymbol{y}\in\mathcal{X}} (\boldsymbol{c} - \boldsymbol{d}_j + \boldsymbol{d}_j)^T\boldsymbol{y} - \boldsymbol{c}^T\boldsymbol{x} + \boldsymbol{d}_j^T\boldsymbol{x} - \boldsymbol{d}_j^T\boldsymbol{x} \right) + b_j \right] \\ &\leq \max_{j\in[J]} \left[a_j (\max_{\boldsymbol{y}\in\mathcal{X}} \boldsymbol{d}_j^T\boldsymbol{y} - \boldsymbol{d}_j^T\boldsymbol{x}) + b_j \right] \\ &\quad + \max_{j\in[J]} a_j \left[\max_{\boldsymbol{y}\in\mathcal{X}} (\boldsymbol{c} - \boldsymbol{d}_j)^T\boldsymbol{y} - (\boldsymbol{c} - \boldsymbol{d}_j)^T\boldsymbol{x} \right] \\ &\leq g(Z(\boldsymbol{d}_j) - {\boldsymbol{d}_j}^T\boldsymbol{x}) + \max_{j\in[J]} a_j \sum_{i=1}^{N} \left[(c_i - d_{ji})^+ - (c_i - d_{ji})x_i \right]. \end{aligned}$$

The first inequality is due to the subadditivity of $Z(\cdot)$, and the second one follows from the fact that $\max_{\boldsymbol{y}\in\mathcal{X}} (\boldsymbol{c} - \boldsymbol{d}_j)^T\boldsymbol{y} \leq \sum_{i=1}^{N} (c_i - d_{ji})^+$ and $a_j \geq 0$ for all $j \in [J]$. For any distribution P, taking expectation on both sides of the above inequality gives

$$\begin{aligned} & \mathbb{E}_P[g(Z(\boldsymbol{c}) - \boldsymbol{c}^T\boldsymbol{x})] \\ \leq\ & g(Z(\boldsymbol{d}_j) - {\boldsymbol{d}_j}^T\boldsymbol{x}) + \mathbb{E}_P \left(\max_{j\in[J]} a_j \sum_{i=1}^{N} [(\tilde{c}_i - d_{ji})^+ - (\tilde{c}_i - d_{ji})x_i] \right) \\ \leq\ & g(Z(\boldsymbol{d}_j) - {\boldsymbol{d}_j}^T\boldsymbol{x}) + \sum_{i=1}^{N} \mathbb{E}_{P_i} \left(\max_{j\in[J]} a_j [(\tilde{c}_i - d_{ji})^+ - (\tilde{c}_i - d_{ji})x_i] \right). \end{aligned}$$

Note that the last inequality follows from the fact that

$$\max_{j\in[J]} (a_j \sum_{i=1}^{N} [(\tilde{c}_i - d_{ji})^+ - (\tilde{c}_i - d_{ji})x_i]) \leq \sum_{i=1}^{N} \max_{j\in[J]} (a_j [(\tilde{c}_i - d_{ji})^+ - (\tilde{c}_i - d_{ji})x_i]).$$

The above inequality holds for any distribution $P \in \mathbb{P}(\mathbb{P}_1, \ldots, \mathbb{P}_N)$ and $\boldsymbol{d}_1, \ldots, \boldsymbol{d}_J \in \Omega$. Taking supremum with respect to $P \in \mathbb{P}(\mathbb{P}_1, \ldots, \mathbb{P}_N)$, and taking minimum with respect to $\boldsymbol{d}_1, \ldots, \boldsymbol{d}_J \in \Omega$, we get

$$\begin{aligned} \phi(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{b}) &= \sup_{P\in\mathbb{P}(\mathbb{P}_1,\ldots,\mathbb{P}_N)} \mathbb{E}_P[g(Z(\boldsymbol{c}) - \boldsymbol{c}^T\boldsymbol{x})] \\ &\leq \min_{\boldsymbol{d}_1,\ldots,\boldsymbol{d}_J\in\Omega} \left\{ \max_{j\in[J]} [a_j(Z(\boldsymbol{d}_j) - \boldsymbol{d}_j^T\boldsymbol{x}) + b_j] \right. \\ &\quad \left. + \sum_{i=1}^{N} \sup_{P_i\in\mathbb{P}_i} \mathbb{E}_{P_i} \left[\max_{j\in[J]} a_j \left([\tilde{c}_i - d_{ji}]^+ - [\tilde{c}_i - d_{ji}]x_i \right) \right] \right\} \\ &= \bar{\phi}(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{b}). \end{aligned}$$

Step 2: Prove that $\phi(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{b}) \geq \bar{\phi}(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{b})$.

Consider the dual problem of (2.13) and the dual of the supremum problem in (2.14). Since the moments lie interior to the set of feasible moment vectors, strong duality holds (see Isii [72]). Hence

$$\begin{aligned}
\phi(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{b}) = \min \Bigg\{ & y_{00} + \sum_{i=1}^{N} \sum_{k=1}^{K_i} y_{ik} m_{ik} \Bigg\} \\
& \text{s.t. } y_{00} + \sum_{i=1}^{N} \sum_{k=1}^{K_i} y_{ik} f_{ik}(c_i) - [a_j(\boldsymbol{c}^T \boldsymbol{y} - \boldsymbol{c}^T \boldsymbol{x}) + b_j] \geq 0, \qquad (2.15) \\
& \quad \forall \boldsymbol{c} \in \Omega, \boldsymbol{y} \in \mathcal{X}, j \in [J].
\end{aligned}$$

$$\begin{aligned}
\bar{\phi}(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{b}) = \min \Bigg\{ & \max_{j \in [J]} [a_j(Z(\boldsymbol{d}_j) - \boldsymbol{d}_j^T \boldsymbol{x}) + b_j] + \sum_{i=1}^{N} \bar{y}_{i0} + \sum_{i=1}^{N} \sum_{k=1}^{K_i} \bar{y}_{ik} m_{ik} \Bigg\} \\
& \text{s.t. } \bar{p}_{i1}(c_i) := \bar{y}_{i0} + \sum_{k=1}^{K_i} \bar{y}_{ik} f_{ik}(c_i) - a_j(c_i - d_{ji})(1 - x_i) \geq 0, \\
& \quad \forall c_i \in \Omega_i, i \in [N], j \in [J], \qquad (2.16) \\
& \quad \bar{p}_{i2}(c_i) := \bar{y}_{i0} + \sum_{k=1}^{K_i} \bar{y}_{ik} f_{ik}(c_i) + a_j(c_i - d_{ji}) x_i \geq 0, \\
& \quad \forall c_i \in \Omega_i, i \in [N], j \in [J].
\end{aligned}$$

Let $y_{00}^*, y_{ik}^*, k \in [K_i], i \in [N]$ be the optimal solution to (2.15). Now generate a feasible solution to (2.16) as follows. Set $\bar{y}_{ik} = y_{ik}^*, k \in [K_i], i \in [N]$. Having fixed $\bar{y}_{ik}$, choose $\bar{y}_{i0}$ and d_{ji} based on the value $x_i = 1$ or 0 in the following manner:

1. If $x_i = 1$, we choose $\bar{y}_{i0}^*$ to be the minimal value such that $\bar{p}_{i1}(c_i)$ is nonnegative over Ω_i. Namely, there exists some $c_i^* \in \Omega_i$ such that $\bar{p}_{i1}(c_i^*) = 0$. Then for all $j \in [J]$, choose d_{ji}^* to be the maximal value such that $\bar{p}_{i2}(c_i)$ is nonnegative over Ω_i. This value can be chosen such that $d_{ji}^* \in \Omega_i$. To verify this observe that since $x_i = 1$, $\bar{p}_{i2}(c_i) = \bar{p}_{i1}(c_i) + a_j(c_i - d_{ji})$. If $a_j = 0$ the result is obvious; if $a_j > 0$ then $d_{ji} = \underline{c}_i$ is feasible since $\bar{p}_{i1}(c_i) \geq 0, \forall c_i \in \Omega_i$. Hence $d_{ji}^* \geq \underline{c}_i$ since it is chosen as the maximal value such that $\bar{p}_{i2}(c_i)$ is nonnegative over Ω_i. Moreover, $d_{ji}^* \leq \overline{c}_i$ or else $p_{i2}(c_i^*) = a_j(c_i^* - d_{ji}) < 0$. Hence $d_{ji}^* \in \Omega_i$.

2. If $x_i = 0$, we choose $\bar{y}^*_{i0}$ to be the minimal value such that $\bar{p}_{i2}(c_i)$ is nonnegative over Ω_i. Then for all $j \in [J]$, choose d^*_{ji} to be the minimal value such that $\bar{p}_{i1}(c_i)$ is nonnegative over Ω_i. A similar argument to the previous case shows that d^*_{ji} can be restricted to the set Ω_i.

Now, for any $j \in [J]$, and any $\boldsymbol{y} \in \mathcal{X}$, from the constraints of (2.16), we obtain that

$$\bar{y}^*_{i0} + \sum_{k=1}^{K_i} y^*_{ik} f_{ik}(c_i) - a_j(c_i - d^*_{ji})(y_i - x_i) \geq 0, \quad \forall c_i \in \Omega_i, i \in [N]. \tag{2.17}$$

By the choice of $\bar{y}^*_{i0}$ and d^*_{ji}, the value $\bar{y}^*_{i0} + a_j d^*_{ji}(y_i - x_i)$ is the minimal value such that the above inequality holds over Ω_i for all $i \in [N]$. Take the summation of these n inequalities:

$$\sum_{i=1}^{N} \bar{y}^*_{i0} + \sum_{i=1}^{N}\sum_{k=1}^{K_i} y^*_{ik} f_{ik}(c_i) - a_j(\boldsymbol{c} - \boldsymbol{d}^*_j)^T(\boldsymbol{y} - \boldsymbol{x}) \geq 0, \forall \boldsymbol{c} \in \Omega, \boldsymbol{y} \in \mathcal{X}, j \in [J]. \tag{2.18}$$

Note that in general, given N univariate functions $\bar{p}_i(c_i) = \sum_{k=1}^{K_i} a_{ik} f_{ik}(c_i) + a_{i0}$ such that a_{i0} is the minimal value for $\bar{p}_i(c_i)$ to be nonnegative over Ω_i, the minimal value of a_{00} for the multivariate function $\bar{p}(\boldsymbol{c}) = \sum_{i=1}^{N}\sum_{k=1}^{K_i} a_{ik} f_{ik}(c_i) + a_{00}$ to be nonnegative over Ω is $\sum_{i=1}^{N} a_{i0}$. By setting $a_{i0} = \bar{y}^*_{i0} + a_j d^*_{ji}(y_i - x_i)$, $a_{00} = y^*_{00} - b_j$, and comparing (2.18) with the constraint of (2.15):

$$y^*_{00} + \sum_{i=1}^{N}\sum_{k=1}^{K_i} y^*_{ik} f_{ik}(c_i) - [a_j(\boldsymbol{c}^T\boldsymbol{y} - \boldsymbol{c}^T\boldsymbol{x}) + b_j] \geq 0, \quad \forall \boldsymbol{c} \in \Omega, \boldsymbol{y} \in \mathcal{X}, j \in [J].$$

This leads to the following result

$$y^*_{00} - b_j \geq \sum_{i=1}^{N} \bar{y}^*_{i0} + a_j \boldsymbol{d}^{*T}_j(\boldsymbol{y} - \boldsymbol{x}), \quad \forall \boldsymbol{y} \in \mathcal{X}, j \in [J],$$

which is equivalent to

$$y^*_{00} \geq \sum_{i=1}^{N} \bar{y}^*_{i0} + g(Z(\boldsymbol{d}^*_j) - \boldsymbol{d}^{*T}_j \boldsymbol{x}).$$

Therefore

$$\bar{\phi}(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{b}) \quad \leq \quad g(Z(\boldsymbol{d}^*_j) - \boldsymbol{d}^{*T}_j \boldsymbol{x}) + \sum_{i=1}^{N} \bar{y}^*_{i0} + \sum_{i=1}^{N}\sum_{k=1}^{K_i} y^*_{ik} m_{ik}$$

$$\leq \; y_{00}^* + \sum_{i=1}^{N} \sum_{k=1}^{K_i} y_{ik}^* m_{ik} \; = \; \phi(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{b}).$$

□

The next proposition provides an extension of the results in Meilijson and Nadas [88] and Bertsimas et al. [21] to the WCVaR of regret.

Proposition 2.4. *Consider the marginal distribution model with* $\mathbb{P}_i = \{P_i\}, i \in [N]$ *or the marginal moment model with* $\mathbb{P}_i = \{P_i : \mathbb{E}_{P_i}[f_{ik}(\tilde{c}_i)] = m_{ik}, k \in [K_i], \mathbb{E}_{P_i}[\mathbb{I}_{[\underline{c}_i, \bar{c}_i]}(\tilde{c}_i)] = 1\}, i \in [N]$. *For* $\boldsymbol{x} \in \mathcal{X} \subseteq \{0,1\}^N$, *the worst-case* CVaR *of regret can be computed as*

$$\text{WCVaR}_\alpha(R(\boldsymbol{x}, \tilde{\boldsymbol{c}})) = \min_{\boldsymbol{d} \in \Omega} \left(Z(\boldsymbol{d}) + \frac{\alpha}{1-\alpha} \boldsymbol{d}^T \boldsymbol{x} + \frac{1}{1-\alpha} \sum_{i=1}^{N} \sup_{P_i \in \mathbb{P}_i} \mathbb{E}_{P_i}([\tilde{c}_i - d_i]^+ - \tilde{c}_i x_i) \right). \tag{2.19}$$

Proof. From the definition of WCVaR in (1.3):

$$\text{WCVaR}_\alpha(R(\boldsymbol{x}, \tilde{\boldsymbol{c}})) = \inf_{v \in \Re} \left(v + \frac{1}{1-\alpha} \sup_{P \in \mathbb{P}(\mathbb{P}_1, \ldots, \mathbb{P}_N)} \mathbb{E}_P \left[Z(\tilde{\boldsymbol{c}}) - \tilde{\boldsymbol{c}}^T \boldsymbol{x} - v \right]^+ \right).$$

Applying Theorems 2.2 and 2.3, we have:

$$\begin{aligned} & \sup_{P \in \mathbb{P}(\mathbb{P}_1, \ldots, \mathbb{P}_N)} \mathbb{E}_P \left[Z(\tilde{\boldsymbol{c}}) - \tilde{\boldsymbol{c}}^T \boldsymbol{x} - v \right]^+ \qquad (2.20) \\ = \; & \min_{\boldsymbol{d} \in \Omega} \left(\left[Z(\boldsymbol{d}) - \boldsymbol{d}^T \boldsymbol{x} - v \right]^+ + \sum_{i=1}^{N} \sup_{P_i \in \mathbb{P}_i} \mathbb{E}_{P_i} \left([\tilde{c}_i - d_i]^+ - [\tilde{c}_i - d_i] x_i \right) \right). \end{aligned}$$

The worst-case CVaR of regret is thus computed as:

$$\min_{\boldsymbol{d} \in \Omega, v \in \Re} \left\{ v + \frac{1}{1-\alpha} [Z(\boldsymbol{d}) - \boldsymbol{d}^T \boldsymbol{x} - v]^+ + \frac{1}{1-\alpha} \sum_{i=1}^{N} \sup_{P_i \in \mathbb{P}_i} \mathbb{E}_{P_i}([\tilde{c}_i - d_i]^+ - [\tilde{c}_i - d_i] x_i) \right\}. \tag{2.21}$$

In formulation (2.21), the optimal decision variable is $v^* = Z(\boldsymbol{d}) - \boldsymbol{d}^T \boldsymbol{x}$ which results in the desired formulation. □

This formulation is appealing computationally since it exploits the marginal distributional representation of the uncertainty. The next result identifies conditions under which the WCVaR of regret is computable in polynomial time for a fixed solution $\boldsymbol{x} \in \mathcal{X}$.

Theorem 2.5. *Assume the following two conditions hold:*

(a) The deterministic combinatorial optimization is solvable in polynomial time, and

(b) For each $i \in [N]$, $G_i(d_i) := \sup_{P_i \in \mathbb{P}_i} \mathbb{E}_{P_i}([\tilde{c}_i - d_i]^+ - \tilde{c}_i x_i)$ and its subgradient with respect to d_i are computable in polynomial time for a fixed $d_i \in \Omega_i$ and x_i.

Then for a given solution $\boldsymbol{x} \in \mathcal{X}$, the worst-case CVaR *of regret under the marginal distribution or marginal moment models is computable in polynomial time.*

Proof. From Proposition 2.4, the WCVaR of regret is computed as:

$$\begin{aligned} \mathrm{WCVaR}_\alpha(R(\boldsymbol{x}, \tilde{\boldsymbol{c}})) = \min_{\boldsymbol{d},t,\boldsymbol{s}} \quad & \left(t + \frac{\alpha}{1-\alpha} \boldsymbol{d}^T \boldsymbol{x} + \frac{1}{1-\alpha} \sum_{i=1}^N s_i \right) \\ \text{s.t.} \quad & t \geq Z(\boldsymbol{d}), \\ & s_i \geq \sup_{P_i \in \mathbb{P}_i} \mathbb{E}_{P_i}([\tilde{c}_i - d_i]^+ - \tilde{c}_i x_i), \quad i \in [N], \quad (2.22) \\ & \boldsymbol{d} \in \Omega. \end{aligned}$$

Denote the feasible set of (2.22) by $\mathcal{K}$. We consider the separation problem of (2.22): given $(\boldsymbol{d}^*, t^*, \boldsymbol{s}^*)$, decide if $(\boldsymbol{d}^*, t^*, \boldsymbol{s}^*) \in \mathcal{K}$, and if not, find a hyperplane which separates $(\boldsymbol{d}^*, t^*, \boldsymbol{s}^*)$ from $\mathcal{K}$. Under assumption (a) and (b), we can check if $(\boldsymbol{d}^*, t^*, \boldsymbol{s}^*) \in \mathcal{K}$ in polynomial time, and if not we consider the following two situations.

1. If $t^* < Z(\boldsymbol{d}^*)$, we can find $\boldsymbol{y}^* \in \mathcal{X}$ such that $Z(\boldsymbol{d}^*) = \boldsymbol{d}^{*T} \boldsymbol{y}^*$ in polynomial time. It follows that the hyperplane $\{(\boldsymbol{d}, t, \boldsymbol{s}) : \boldsymbol{y}^{*T} \boldsymbol{d} = t\}$ separates $(\boldsymbol{d}^*, t^*, \boldsymbol{s}^*)$ from $\mathcal{K}$.

2. If $s_i^* < G_i(d_i^*)$ for some $i \in [N]$, then we can find the separating hyperplane in polynomial time, since the subgradient of $G_i(d_i)$ is computable in polynomial time. The remaining constraints $\boldsymbol{d} \in \Omega$ are $2N$ linear constraints that are easy to enforce. Hence, the separation problem of (2.22) can be solved in

polynomial time. It follows that the WCVaR of regret under the marginal model is computable in polynomial time.

□

Many combinatorial optimization problems satisfy the assumption (a) in Theorem 2.5. Examples include the longest path problem on a directed acyclic graph, spanning tree problems and assignment problems. Moreover, in the marginal distribution model and several instances of the marginal moment model, Assumption (b) in Theorem 2.5 is easy to verify. For both the continuous and discrete marginal distribution model, $G_i(d_i)$ is a convex function of d_i and a subgradient of the function is given by $-P(\tilde{c}_i \geq d_i)$. For the marginal moment model when (a) the range and mean are given, or (b) the range, mean and mean absolute deviation are given, $G_i(d_i)$ is a piecewise linear convex function that is efficiently computable (see Madansky [87] and Ben-Tal and Hochman [16]). If $P_i^* \in \mathbb{P}_i$ denotes the extremal distribution that attains the bound $\sup_{P_i \in \mathbb{P}_i} \mathbb{E}_{P_i}([\tilde{c}_i - d_i]^+ - \tilde{c}_i x_i)$ in these instances, then a subgradient of the function is given by $-P_i^*(\tilde{c}_i \geq d_i)$.

2.3.2 WCVaR of Cost

In this subsection, we apply the previous results to combinatorial optimization problems with an objective of minimizing the WCVaR of cost:

$$\min_{\boldsymbol{x} \in \mathcal{X}} \text{WCVaR}_\alpha \left(-\tilde{\boldsymbol{c}}^T \boldsymbol{x}\right). \tag{2.23}$$

Applying similar arguments as the regret framework, we obtain the following proposition for the WCVaR of cost under the marginal models.

Proposition 2.6. *Consider the marginal distribution model with* $\mathbb{P}_i = \{P_i\}, i \in [N]$ *or the marginal moment model with* $\mathbb{P}_i = \{P_i : \mathbb{E}_{P_i}[f_{ik}(\tilde{c}_i)] = m_{ik}, k \in [K_i], \mathbb{E}_{P_i}[\mathbb{I}_{[\underline{c}_i, \bar{c}_i]}(\tilde{c}_i)] = 1\}, i \in [N]$. *For* $\boldsymbol{x} \in \mathcal{X} \subseteq \{0,1\}^N$, *the* WCVaR *of cost can be computed as*

$$\text{WCVaR}_\alpha(-\tilde{\boldsymbol{c}}^T \boldsymbol{x}) = \min_{\boldsymbol{d} \in \Omega} \left\{ -\boldsymbol{d}^T \boldsymbol{x} + \frac{1}{1-\alpha} \sum_{i=1}^{N} \sup_{P_i \in \mathbb{P}_i} \mathbb{E}_{P_i}([d_i - \tilde{c}_i]^+) \right\}. \tag{2.24}$$

Now we consider the problem of minimizing the WCVaR of cost. Since the objective function in (2.24) is separable in d_i, it can be expressed as:

$$\mathrm{WCVaR}_\alpha(-\tilde{\boldsymbol{c}}^T\boldsymbol{x}) = \sum_{i=1}^{N} \hat{h}_i(x_i), \tag{2.25}$$

where the function $\hat{h}_i(x_i)$ is defined as:

$$\hat{h}_i(x_i) = \min_{d_i \in \Omega_i} \left\{ -d_i x_i + \frac{1}{1-\alpha} \sup_{P_i \in \mathbb{P}_i} \mathbb{E}_{P_i}([d_i - \tilde{c}_i]^+) \right\}.$$

Define the parameter h_i as the optimal value to a univariate convex programming problem:

$$h_i = \min_{d_i \in \Omega_i} \left\{ -d_i + \frac{1}{1-\alpha} \sup_{P_i \in \mathbb{P}_i} \mathbb{E}_{P_i}([d_i - \tilde{c}_i]^+) \right\}. \tag{2.26}$$

Then the worst-case CVaR of the cost can be expressed as

$$\mathrm{WCVaR}_\alpha(-\tilde{\boldsymbol{c}}^T\boldsymbol{x}) = \sum_{i=1}^{N} h_i x_i. \tag{2.27}$$

To see why this is true observe that if $x_i = 1$, then $\hat{h}_i(1) = h_i x_i$ and if $x_i = 0$, then $\hat{h}_i(0) = 0$. By using the above analysis, we can easily obtain the following theorem which gives the main method to solve the problem of minimizing the WCVaR of cost.

Theorem 2.7. *In the marginal distribution model with* $\mathbb{P}_i = \{P_i\}, i \in [N]$ *or the marginal moment model with* $\mathbb{P}_i = \{P_i : \mathbb{E}_{P_i}[f_{ik}(\tilde{c}_i)] = m_{ik}, k \in [K_i], \mathbb{E}_{P_i}[\mathbb{I}_{[\underline{c}_i, \bar{c}_i]}(\tilde{c}_i)] = 1\}, i \in [N]$, *the problem of minimizing the* WCVaR *of cost can be formulated as the deterministic combinatorial optimization problem:*

$$\min_{\boldsymbol{x} \in \mathcal{X}} \mathrm{WCVaR}_\alpha(-\tilde{\boldsymbol{c}}^T\boldsymbol{x}) = \min_{\boldsymbol{x} \in \mathcal{X}} \sum_{i=1}^{N} h_i x_i, \tag{2.28}$$

where h_i *is the optimal value of (2.26),i.e.*

$$h_i = \min_{d_i \in \Omega_i} \left\{ -d_i + \frac{1}{1-\alpha} \sup_{P_i \in \mathbb{P}_i} \mathbb{E}_{P_i}([d_i - \tilde{c}_i]^+) \right\}. \tag{2.29}$$

In (2.29), h_i can be easily obtained by solving a univariable convex optimization problem. Hence if the deterministic combinatorial optimization problem is

polynomially solvable, the problem of minimizing the WCVaR of cost is also polynomially solvable under the marginal distribution and marginal moment models. The problem of minimizing the WCVaR of cost is still tractable in this case. However, the problem of minimizing the WCVaR of regret is a difficult problem and NP-hard in general. In the next section, we provide conic mixed integer programs to minimize the WCVaR of regret.

2.4 Mixed Integer Programming Formulations

From Proposition 2.4, the problem of minimizing the WCVaR of regret is formulated as:

$$\min_{\boldsymbol{x}\in\mathcal{X}} \mathrm{WCVaR}_\alpha(R(\boldsymbol{x},\tilde{\boldsymbol{c}})) \;=\; \min_{\boldsymbol{x}\in\mathcal{X},\boldsymbol{d}\in\Omega} \left(Z(\boldsymbol{d}) + \frac{\alpha}{1-\alpha}\boldsymbol{d}^T\boldsymbol{x} + \frac{1}{1-\alpha}H(\boldsymbol{x},\boldsymbol{d}) \right), \tag{2.30}$$

where

$$H(\boldsymbol{x},\boldsymbol{d}) \;:=\; \sum_{i=1}^{N} \sup_{P_i\in\mathbb{P}_i} \mathbb{E}_{P_i}([\tilde{c}_i - d_i]^+ - \tilde{c}_i x_i). \tag{2.31}$$

Formulation (2.30) is a stochastic nonconvex mixed integer programming problem where the nonconvexity appears in the term $\boldsymbol{d}^T\boldsymbol{x}$. For bilinear terms, several linearization techniques have been proposed in the literature by Glover [54], Glover and Woolsey [56, 57], Sherali and Alameddine [110] and Adams and Sherali [2] among others. These alternative linearization techniques vary significantly in terms of their computational performance. We adopt the simplest linearization technique from Glover [54] to handle the bilinear terms where one set of variables is restricted to be binary. For all $i \in [N]$, and $\boldsymbol{x} \in \mathcal{X} \subseteq \{0,1\}^N$,

$$z_i = d_i x_i \Leftrightarrow \begin{cases} \overline{c}_i x_i \geq z_i \geq \underline{c}_i x_i \\ d_i - \underline{c}_i(1-x_i) \geq z_i \geq d_i - \overline{c}_i(1-x_i). \end{cases} \tag{2.32}$$

By applying the linearization technique, (2.30) is reformulated as the following stochastic convex mixed integer program:

$$\min_{\boldsymbol{x},\boldsymbol{d},\boldsymbol{z}} \quad \left(Z(\boldsymbol{d}) + \frac{\alpha}{1-\alpha}\sum_{i=1}^{N} z_i + \frac{1}{1-\alpha}H(\boldsymbol{x},\boldsymbol{d}) \right)$$

$$\begin{aligned} \text{s.t.} \quad & \overline{c}_i x_i \geq z_i \geq \underline{c}_i x_i, \quad i \in [N], \\ & d_i - \underline{c}_i(1 - x_i) \geq z_i \geq d_i - \overline{c}_i(1 - x_i), \quad i \in [N], \\ & \boldsymbol{d} \in \Omega, \quad \boldsymbol{x} \in \mathcal{X}. \end{aligned} \tag{2.33}$$

The objective function in (2.33) is convex with respect to $\boldsymbol{x}, \boldsymbol{d}, \boldsymbol{z}$ since convexity is preserved under the expectation and maximization operation.

Assume that the feasible region $\mathcal{X}$ is described in the compact form:

$$\mathcal{X} = \left\{ \boldsymbol{y} \in \{0,1\}^N \mid \boldsymbol{A}\boldsymbol{y} = \boldsymbol{b} \right\},$$

where $\boldsymbol{A}$ is a given integer matrix and $\boldsymbol{b}$ is a given integer vector. For the rest of this section, we assume that matrix $\boldsymbol{A}$ is totally unimodular, namely each square submatrix of $\boldsymbol{A}$ has determinant equal to 0, +1, or -1. Under this assumption the deterministic combinatorial optimization problem is solvable in polynomial time as a compact linear program (see Schrijver [109]):

$$Z(\boldsymbol{d}) = \max \left\{ \boldsymbol{d}^T \boldsymbol{y} \mid \ \boldsymbol{A}\boldsymbol{y} = \boldsymbol{b},\ 0 \leq y_i \leq 1, i \in [N] \right\}. \tag{2.34}$$

Many polynomially solvable 0-1 optimization problems fall under this category including subset selection, longest path on a directed acyclic graph and linear assignment problems. Let $(\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2)$ be the vectors of dual variables associated with the constraints of (2.34). The dual linear program of (2.34) is given by

$$Z(\boldsymbol{d}) = \min \left\{ \boldsymbol{b}^T \boldsymbol{\lambda}_1 + \boldsymbol{e}^T \boldsymbol{\lambda}_2 \mid \ \boldsymbol{A}^T \boldsymbol{\lambda}_1 + \boldsymbol{\lambda}_2 \geq \boldsymbol{d},\ \boldsymbol{\lambda}_2 \geq 0 \right\}, \tag{2.35}$$

where $\boldsymbol{e}$ is the vector of all ones. By solving the dual formulation of $Z(\boldsymbol{d})$ in (2.33), we get:

$$\min_{\boldsymbol{x},\boldsymbol{d},\boldsymbol{z},\boldsymbol{\lambda}_1,\boldsymbol{\lambda}_2} \quad \left(\boldsymbol{b}^T \boldsymbol{\lambda}_1 + \boldsymbol{e}^T \boldsymbol{\lambda}_2 + \frac{\alpha}{1-\alpha} \sum_{i=1}^{N} z_i + \frac{1}{1-\alpha} H(\boldsymbol{x}, \boldsymbol{d}) \right) \tag{2.36}$$

$$\text{s.t.} \quad \overline{c}_i x_i \geq z_i \geq \underline{c}_i x_i, \quad i \in [N], \tag{2.36a}$$

$$d_i - \underline{c}_i(1 - x_i) \geq z_i \geq d_i - \overline{c}_i(1 - x_i), \quad i \in [N], \tag{2.36b}$$

$$\boldsymbol{d} \in \Omega,\ \boldsymbol{A}^T \boldsymbol{\lambda}_1 + \boldsymbol{\lambda}_2 \geq \boldsymbol{d},\ \boldsymbol{\lambda}_2 \geq 0,\ \boldsymbol{x} \in \mathcal{X}. \tag{2.36c}$$

The constraints in problem (2.36) are all linear except for the integrality restrictions in the description of $\mathcal{X}$. To convert this to a conic mixed integer program, we apply standard conic programming methods to evaluate $H(\boldsymbol{x}, \boldsymbol{d})$ in the objective function.

2.4.1 Marginal Discrete Distribution Model

Assume that the marginal distributions of $\tilde{\boldsymbol{c}}$ are discrete:

$$\tilde{c}_i \sim c_{ij} \text{ with probability } p_{ij},\ j \in [J_i], i \in [N]$$

where $\sum_{j\in[J_i]} p_{ij} = 1$ and $\sum_{j\in[J_i]} c_{ij}p_{ij} = \mu_i$ for each $i \in [N]$. The input specification for the marginal discrete distribution model needs $J_1 + J_2 + \ldots + J_N$ probabilities which is typically much smaller than the size of the input needed to specify the joint distribution that needs up to $J_1 \times J_2 \times \ldots \times J_N$ probabilities. In this case:

$$H(\boldsymbol{x}, \boldsymbol{d}) \ = \ \sum_{i=1}^{N}\sum_{j=1}^{J_i}(c_{ij} - d_i)^+ p_{ij} - \boldsymbol{\mu}^T\boldsymbol{x} \ = \ \min_{t_{ij}\geq c_{ij}-d_i, t_{ij}\geq 0} \sum_{i=1}^{N}\sum_{j=1}^{J_i} t_{ij}p_{ij} - \boldsymbol{\mu}^T\boldsymbol{x},$$

The problem of minimizing WCVaR is thus formulated as the compact MILP:

$$\begin{aligned} \min_{\boldsymbol{x},\boldsymbol{d},\boldsymbol{z},\boldsymbol{\lambda}_1,\boldsymbol{\lambda}_2,\boldsymbol{t}} \quad & \left\{ \boldsymbol{b}^T\boldsymbol{\lambda}_1 + \boldsymbol{e}^T\boldsymbol{\lambda}_2 + \frac{\alpha}{1-\alpha}\sum_{i=1}^{N} z_i + \frac{1}{1-\alpha}\left(\sum_{i=1}^{N}\sum_{j=1}^{J_i} t_{ij}p_{ij} - \boldsymbol{\mu}^T\boldsymbol{x}\right)\right\} \\ \text{s.t.} \quad & t_{ij} \geq c_{ij} - d_i, t_{ij} \geq 0, j \in [J_i], i \in [N], \\ & (2.36a), (2.36b) \text{ and } (2.36c). \end{aligned} \tag{2.37}$$

2.4.2 Marginal Moment Model

In the standard representation of the marginal moment model, $H(\boldsymbol{x}, \boldsymbol{d})$ is evaluated through conic optimization. This is based on the well-known duality theory of moments and nonnegative polynomials for univariate models. The reader is referred to Nesterov [94] and Bertsimas and Popescu [22] for details. We restrict attention to instances of the marginal moment model where (2.36) can be solved as a MILP

or MISOCP. The advantage of these formulations is that the probabilistic regret model can be solved with standard off-the-shelf solvers such as CPLEX. The details are listed next:

(a) Range and Mean are Known:

Assume the interval range and mean of the random vector $\tilde{\boldsymbol{c}}$ are given:

$$\mathbb{P}_i = \{P_i : \mathbb{E}_{P_i}[\tilde{c}_i] = \mu_i, \mathbb{E}_{P_i}[\mathbb{I}_{[\underline{c}_i, \overline{c}_i]}(\tilde{c}_i)] = 1\}.$$

In this case, the optimal distribution to the problem $\sup_{P_i \in \mathbb{P}_i} \mathbb{E}_{P_i}\left[(\tilde{c}_i - d_i)^+ - \tilde{c}_i x_i\right]$ is known explicitly (see Madansky [87] and Ben-Tal and Hochman [16]):

$$\tilde{c}_i = \begin{cases} \overline{c}_i, \text{ with probability } \frac{\mu_i - \underline{c}_i}{\overline{c}_i - \underline{c}_i}, \\ \underline{c}_i, \text{ with probability } \frac{\overline{c}_i - \mu_i}{\overline{c}_i - \underline{c}_i}. \end{cases}$$

The worst-case marginal distribution is a two point distribution and can be treated as a special case of the discrete marginal distribution. The probabilistic regret model is solved with the MILP (2.37).

(b) Range, Mean and Mean Absolute Deviation are Known:

Assume the interval range, mean and the mean absolute deviation of the random vector $\tilde{\boldsymbol{c}}$ are given:

$$\mathbb{P}_i = \{P_i : \mathbb{E}_{P_i}(\tilde{c}_i) = \mu_i, \mathbb{E}_{P_i}(|\tilde{c}_i - \mu_i|) = \delta_i, \mathbb{E}_{P_i}[\mathbb{I}_{[\underline{c}_i, \overline{c}_i]}(\tilde{c}_i)] = 1\}.$$

For feasibility the mean absolute deviation satisfies $\delta_i \leq \frac{2(\overline{c}_i - \mu_i)(\mu_i - \underline{c}_i)}{\overline{c}_i - \underline{c}_i}$. The optimal distribution for $\sup_{P_i \in \mathbb{P}_i} \mathbb{E}_{P_i}\left[(\tilde{c}_i - d_i)^+ - \tilde{c}_i x_i\right]$ has been identified by Ben-Tal and Hochman [16]:

$$\tilde{c}_i = \begin{cases} \underline{c}_i, \text{ with probability } \frac{\delta_i}{2(\mu_i - \underline{c}_i)} =: p_i, \\ \overline{c}_i, \text{ with probability } \frac{\delta_i}{2(\overline{c}_i - \mu_i)} =: q_i, \\ \mu_i, \text{ with probability } 1 - p_i - q_i. \end{cases}$$

This is a three point distribution and the MILP reformulation (2.37) can be used.

(c) Range, Mean and Standard Deviation are Known:

Assume the range, mean and the standard deviation of the random vector $\tilde{\boldsymbol{c}}$ are given:

$$\mathbb{P}_i = \{P_i : \mathbb{E}_{P_i}(\tilde{c}_i) = \mu_i, \mathbb{E}_{P_i}(\tilde{c}_i^2) = \mu_i^2 + \sigma_i^2, \mathbb{E}_{P_i}[\mathbb{I}_{[\underline{c}_i, \overline{c}_i]}(\tilde{c}_i)] = 1\}.$$

By using duality theory, we have:

$$\begin{aligned}\sup_{P_i\in\mathbb{P}_i}\mathbb{E}_{P_i}\left[(\tilde{c}_i-d_i)^+-\tilde{c}_ix_i\right]=\min\ & y_{i0}+\mu_iy_{i1}+(\mu_i^2+\sigma_i^2)y_{i2}-\mu_ix_i \\ \text{s.t. }& y_{i0}+y_{i1}c_i+y_{i2}c_i^2-(c_i-d_i)\geq 0,\ \forall c_i\in[\underline{c}_i,\overline{c}_i], \\ & y_{i0}+y_{i1}c_i+y_{i2}c_i^2\geq 0,\quad \forall c_i\in[\underline{c}_i,\overline{c}_i].\end{aligned} \tag{2.38}$$

By applying the S-lemma to the constraints of the above problem, problem (2.38) can be formulated as

$$\begin{aligned}\min\quad & y_{i0}+\mu_iy_{i1}+(\mu_i^2+\sigma_i^2)y_{i2}-\mu_ix_i \\ \text{s.t.}\quad & \tau_{i1}\geq 0,\quad y_{i0}+d_i+\underline{c}_i\overline{c}_i\tau_{i1}\geq 0,\quad y_{i2}+\tau_{i1}\geq 0, \\ & \tau_{i2}\geq 0,\quad y_{i0}+\underline{c}_i\overline{c}_i\tau_{i2}\geq 0,\quad y_{i2}+\tau_{i2}\geq 0, \\ & \left\|\begin{matrix}y_{i1}-1-(\underline{c}_i+\overline{c}_i)\tau_{i1}\\ y_{i0}+d_i+(\underline{c}_i\overline{c}_i-1)\tau_{i1}-y_{i2}\end{matrix}\right\|_2\leq y_{i0}+d_i+(\underline{c}_i\overline{c}_i+1)\tau_{i1}+y_{i2}, \\ & \left\|\begin{matrix}y_{i1}-(\underline{c}_i+\overline{c}_i)\tau_{i2}\\ y_{i0}+(\underline{c}_i\overline{c}_i-1)\tau_{i2}-y_{i2}\end{matrix}\right\|_2\leq y_{i0}+(\underline{c}_i\overline{c}_i+1)\tau_{i2}+y_{i2}.\end{aligned} \tag{2.39}$$

The problem of minimizing WCVaR of regret can be formulated in this case as the mixed integer SOCP:

$$\begin{aligned}\min_{\boldsymbol{x},\boldsymbol{d},\boldsymbol{z},\boldsymbol{\lambda}_1,\boldsymbol{\lambda}_2\boldsymbol{y},\boldsymbol{\tau}}\ & \left\{\boldsymbol{b}^T\boldsymbol{\lambda}_1+\boldsymbol{e}^T\boldsymbol{\lambda}_2+\frac{1}{1-\alpha}\left(\sum_{i=1}^N\left[\alpha z_i+y_{i0}+\mu_iy_{i1}+(\mu_i^2+\sigma_i^2)y_{i2}\right]-\boldsymbol{\mu}^T\boldsymbol{x}\right)\right\} \\ \text{s.t.}\quad & \tau_{i1}\geq 0,\quad y_{i0}+d_i+\underline{c}_i\overline{c}_i\tau_{i1}\geq 0,\quad y_{i2}+\tau_{i1}\geq 0,\ \ \forall\, i\in[N], \\ & \tau_{i2}\geq 0,\quad y_{i0}+\underline{c}_i\overline{c}_i\tau_{i2}\geq 0,\quad y_{i2}+\tau_{i2}\geq 0,\ \ \forall\, i\in[N], \\ & \left\|\begin{matrix}y_{i1}-1-(\underline{c}_i+\overline{c}_i)\tau_{i1}\\ y_{i0}+d_i+(\underline{c}_i\overline{c}_i-1)\tau_{i1}-y_{i2}\end{matrix}\right\|_2\leq y_{i0}+d_i+(\underline{c}_i\overline{c}_i+1)\tau_{i1}+y_{i2},\ \ \forall\, i\in[N], \\ & \left\|\begin{matrix}y_{i1}-(\underline{c}_i+\overline{c}_i)\tau_{i2}\\ y_{i0}+(\underline{c}_i\overline{c}_i-1)\tau_{i2}-y_{i2}\end{matrix}\right\|_2\leq y_{i0}+(\underline{c}_i\overline{c}_i+1)\tau_{i2}+y_{i2},\ \ \forall\, i\in[N], \\ & (2.36a),(2.36b)\text{ and }(2.36c).\end{aligned} \tag{2.40}$$

The regret formulations identified in this section are compact size mixed integer conic programs and generalize to higher order moments using mixed integer semidefinite programs.

2.5 Numerical Examples

Consider a directed, acyclic network $G = (V, A)$ with a finite set of vertices V and a finite set of arcs A. Associated with each arc, is the duration (length) of that arc. The goal is to find the shortest path from a fixed source node to the sink node. When the arc lengths are deterministic, the shortest path problem can be solved efficiently. However, when the arc lengths are random, the definition of a "shortest path" has to be suitably modified.

Shortest paths under a stochastic setting is a well studied problem [75, 126, 10, 77, 101]. Some of the possible approaches to determine the "shortest path" in the stochastic framework are discussed next.

1. **Expected Shortest Path:** The classical approach chooses the path with the shortest length in an expected sense.

2. **Most Likely Path:** Kamburowski [75] defined the optimality index of a path to be the probability that it is the shortest path. The "shortest path" in this case is defined as the path with the greatest optimality index and is termed as the most likely path. Unlike the expected shortest path, computing the most likely path is highly challenging even for moderate size networks.

3. **Absolute Robust Path:** An absolute robust path is defined as the path that is the shortest under the worst-case scenario. In the interval uncertainty model, this path is found by solving the shortest path problem on the graph when the arc lengths are replaced by the largest length for each arc.

4. **Minmax Regret Path:** In recent years, the shortest path with the minmax regret criterion has been proposed as an alternative decision criterion. In the interval uncertainty case, Zieliński [126] showed that the minmax regret shortest path problem is NP-hard even when the graph is restricted to be directed, acyclic and planar with vertex degrees at most three. Mixed integer linear programs to solve the interval uncertainty minmax regret path have

been developed in Yaman et. al. [121].

5. **Minimum WCVaR Cost Path:** Choose the path by minimizing the WCVaR of the cost.

6. **Minimum WCVaR Regret Path:** Choose the path by minimizing the WCVaR of regret.

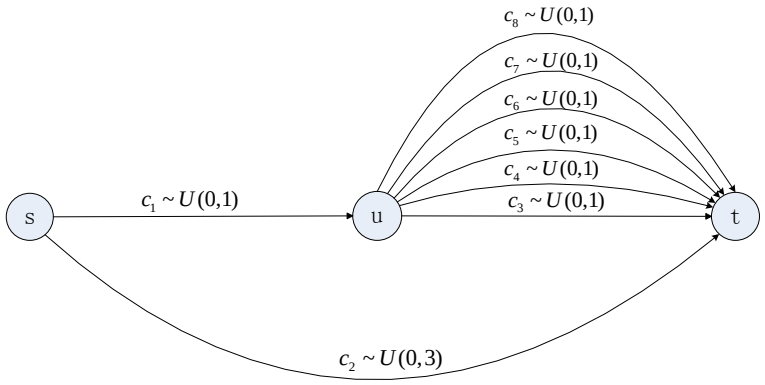


Figure 2.2: Network for Example 2.1

Example 2.1. *In figure 2.2, arc length* $\tilde{c}_2 \sim$ *uniform*$(0,3)$*, and the other arc length* $\tilde{c}_i \sim$ *uniform*$(0,1)$*,* $i \neq 2$*. The goal is to find a shortest path from s to t. This example is from Reich and Lopes [101].*

The choices of paths passing through the intermediate node u have expected length 1, worst-case length 2, and maximum regret 2, while the path consisting of $\tilde{c}_2$ has expected length 1.5, worst-case length 3, and maximum regret 3. In the sense of (1) Expected shortest path, (3) Absolute robust path, (4) Minmax regret path, and (5) Minimum WCVaR cost Path, the "shortest path" is any path passing through the intermediate node u. In the sense of (2) Most likely path, the "shortest path" consists of $\tilde{c}_2$ (see Reich and Lopes[101]). To solve the probabilistic regret model, we use only the marginal moment information. Consider the following three cases (a) known range and mean, (b) known range, mean and mean absolute deviation and (c) known range, mean and variance. In all the three cases, by choosing the probability level $\alpha \in [0, 0.99)$ the optimal decision (6) Min WCVaR Regret Path

is always one of the paths passing through the intermediate node u, which is the same as the decision of (1), (3), (4) and (5). This result is in agreement with the intuition that while the path consisting of arc $\tilde{c}_2$ is the most likely shortest path, in terms of worst-case value and regret it is not the best one.

Example 2.2. *Reconsider the example shown in Figure 2.1 in Section 1 with a network that consists of four nodes and five arcs. All the lengths of the arcs are known to lie in interval ranges with the means and standard deviations of the lengths given.*

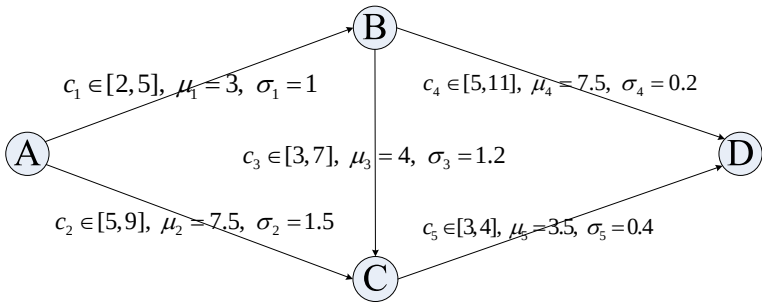


Figure 2.3: Network for Example 2.2

The network in Example 2.2 is the Wheatstone bridge network with the objective of finding the shortest path from node A to node D. The solutions identified from the Expected shortest path, Absolute robust shortest path, Minmax regret path, Minimum WCVaR cost path and Minimum WCVaR regret path are provided in Table 2.2.

The expected shortest path uses only the mean information, and the absolute robust and minmax regret approaches use only range information. However, the model of minimizing the WCVaR of cost and regret can deal with more probabilistic information. As the probability level α is varied, the minimum WCVaR regret decision changes. This is consistent with the observation that α captures the decision-maker's aversion to regret where a larger α implies higher aversion to regret. If the decision-maker is regret neutral, by setting $\alpha = 0$, the method reduces to the expected shortest path method where the mean is specified for each

Table 2.2: The stochastic "shortest path"

Methods	"Shortest path"	Information
Expected shortest path	$1-4$ or $1-3-5$	Mean
Absolute robust path	$2-5$	Range
Minmax regret path	$2-5$	Range
Min WCVaR cost path	$2-5$ if $0.5001 \leq \alpha < 1$ $1-3-5$ if $0 < \alpha \leq 0.5000$	Range and mean
Min WCVaR cost path	$2-5$ if $0.8261 \leq \alpha < 1$ $1-4$ if $0 < \alpha \leq 0.8260$	Range, mean and standard deviation
Min WCVaR regret path	$2-5$ if $0.6667 \leq \alpha < 1$ $1-3-5$ if $0 < \alpha \leq 0.6666$	Range and mean
Min WCVaR regret path	$2-5$ if $0.6883 \leq \alpha < 1$ $1-4$ if $0 < \alpha \leq 0.6882$	Range, mean and standard deviation

arc. Moreover, the choice of the solution is sensitive to the probability information available. If we only use the range and mean information, path 1-3-5 always dominates path 2-5 if $\alpha \leq 0.6666$, although they have the same expected traveling time. This should correspond to our intuition since the range of the edge c_4 is significantly larger than the range of edge c_3 and c_5, and there are more edges in path 1-3-5 which can spread more risk than path 1-4. However, when the standard deviation information is also involved, the standard deviation of edge c_4 is much smaller than the standard deviation of c_3 and c_5, which means that the risk of c_4 is smaller. Hence in this case, it is intuitive to expect that path 1-4 dominates path 1-3-5 as indicated in Table 2.2. The optimal decision of minimizing the WCVaR of cost is similar to the decision of minimizing the WCVaR of regret. In this example, this can be partly explained by the observation that the absolute robust path and minmax regret path are the same. When α is close to 0, Min WCVaR cost path and Min WCVaR regret path reduce to the expected shortest path, and when α goes to 1, the solutions reduce to the absolute robust path and minmax regret path which are the same in this example.

Example 2.3. *The previous two examples used small-sized networks. We now create a fictitious network in the form of a square grid graph with width and height both equal to H as in Figure 2.4. There are H^2 nodes and $2H(H-1)$ arcs in the*

graph. The start node is at the left bottom corner and the end node is at the right upper corner. Each arc on the graph proceeds either towards the right node or the upper node.

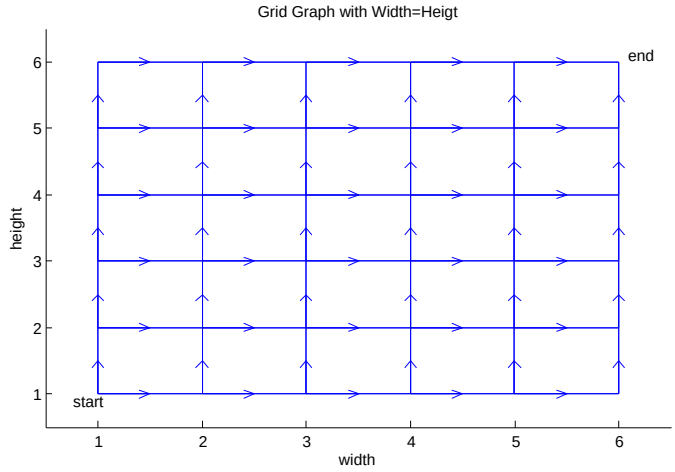


Figure 2.4: Grid Graph with $H = 6$

We evaluate the CPU times needed to minimize the WCVaR of regret and cost with randomly generated data. In this experiment, the interval range for each arc length $[\underline{c}_i, \overline{c}_i]$ is randomly generated with $\underline{c}_i = \min\{a_i, b_i\}$, $\overline{c}_i = \max\{a_i, b_i\}$, where a_i, b_i are chosen from the uniform distribution $U[1, 10]$. The mean is randomly generated as $\mu_i \sim U[\underline{c}_i, \overline{c}_i]$. Define $\overline{\delta}_i = 2\frac{(\overline{c}_i - \mu_i)(\mu_i - \underline{c}_i)}{\overline{c}_i - \underline{c}_i}$ as the largest mean absolute deviation when the mean and range of $\tilde{c}_i$ are given. Let the mean absolute deviation of $\tilde{c}_i$ be randomly generated by $\delta_i \sim U[0, \overline{\delta}_i]$. We report the CPU time taken to minimize the WCVaR of regret and cost for the following two cases of the marginal moment model: (a) range $[\underline{c}_i, \overline{c}_i]$ and mean μ_i are given and (b) range $[\underline{c}_i, \overline{c}_i]$, mean μ_i and mean absolute deviation δ_i are given. The results are shown in Table 2.3.

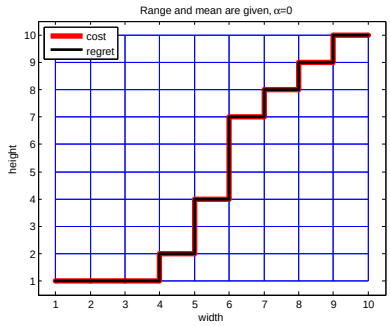
The computational studies were implemented in Matlab R2012a on an Intel Core 2 Duo CPU 2.8GHz laptop with 4 GB of RAM. In Table 2.3, the CPU time (in the format of seconds) is the average execution time for 10 randomly generated instances, and "$**$" indicates that the instances ran out of memory. The CPU time taken to minimize the WCVaR of cost is very small (< 0.03 seconds), since this problem is solvable as a linear programming problem (see Theorem 2.7). To

Table 2.3: Average CPU time to minimize the WCVaR of cost and regret, $\alpha = 0.8$

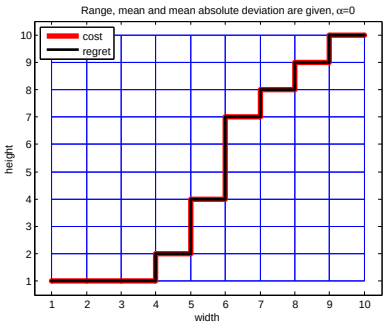
H	Nodes	Arcs	$\min_{\boldsymbol{x}\in\mathcal{X}} \text{WCVaR}_\alpha(-\tilde{\boldsymbol{c}}^T\boldsymbol{x})$		$\min_{\boldsymbol{x}\in\mathcal{X}} \text{WCVaR}_\alpha(Z(\tilde{\boldsymbol{c}}) - \tilde{\boldsymbol{c}}^T\boldsymbol{x})$	
			$[\underline{c}_i, \overline{c}_i], \mu_i$ are given	$[\underline{c}_i, \overline{c}_i], \mu_i, \delta_i$ are given	$[\underline{c}_i, \overline{c}_i], \mu_i$ are given	$[\underline{c}_i, \overline{c}_i], \mu_i, \delta_i$ are given
10	100	180	3.10e-03	0.00e+00	2.95e-01	3.62e-01
12	144	264	1.50e-03	4.70e-03	6.52e-01	8.11e-01
14	196	364	3.00e-03	3.20e-03	6.24e-01	9.63e-01
16	256	480	9.60e-03	3.20e-03	9.95e-01	1.07e+00
18	324	612	1.09e-02	1.60e-03	1.29e+00	1.86e+00
20	400	760	8.00e-03	9.30e-03	1.24e+00	1.96e+00
21	441	840	9.60e-03	1.26e-02	1.86e+00	2.41e+00
22	484	924	1.11e-02	1.40e-02	2.74e+00	3.68e+00
23	529	1012	1.59e-02	1.39e-02	3.31e+00	4.46e+00
24	576	1104	1.57e-02	1.40e-02	3.34e+00	**
25	625	1200	2.05e-02	1.69e-02	3.68e+00	**
26	676	1300	1.99e-02	2.07e-02	**	**

minimize the WCVaR of regret, we use CPLEX to solve the binary integer linear programming problem. When (a) range and mean information are given, we can solve the regret problem to optimality for $H = 25$ (i.e. 625 nodes and 1200 edges) in around 4 seconds; when (b) range, mean and mean absolute deviation are given, we can solve the regret problem to optimality for $H = 23$ (i.e. 529 nodes and 1012 edges) in around 5 seconds.

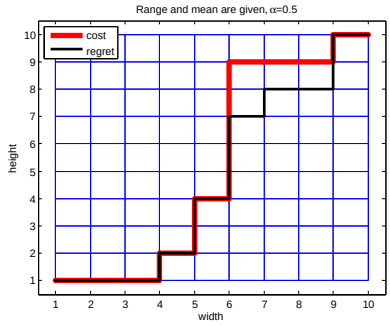
Next we compare the optimal paths obtained from minimizing the WCVaR of cost and regret at different probability levels α. Let $H = 10$, and assume $\underline{c}_i, \overline{c}_i, \mu_i, \delta_i,\ i = 1, 2, \ldots, 2H(H-1)$ are given. The optimal paths that minimize the WCVaR of cost and regret with (a) the range and mean information, and (b) the range, mean and mean absolute deviation information are provided in Figure 2.5. From Figure 2.5, we see that when $\alpha = 0$, the optimal paths that minimize the WCVaR of cost and regret are the same regardless of whether (a) range and mean information are given or (b) range, mean and mean absolute deviation are given. In this case, the two models reduce to the deterministic shortest path problem where every edge length equals to its mean. When α is close to 1, the Min-WCVaR-cost path approaches the absolute robust path, and the Min-WCVaR-regret path approaches the minmax regret path. For intermediate values



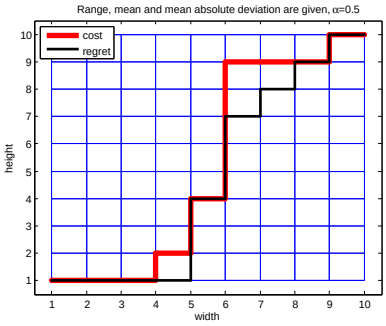
(a) $[\underline{c}_i, \overline{c}_i]$ and μ_i are given, $\alpha = 0$



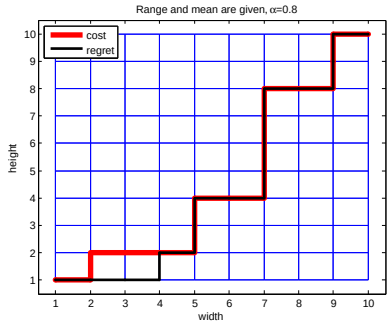
(b) $[\underline{c}_i, \overline{c}_i]$, μ_i and δ_i are given, $\alpha = 0$



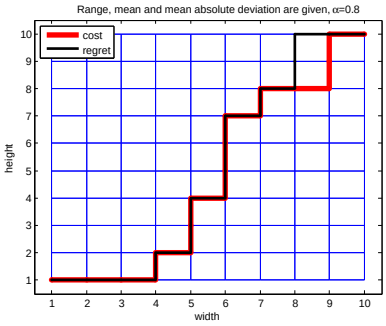
(c) $[\underline{c}_i, \overline{c}_i]$ and μ_i are given, $\alpha = 0.5$



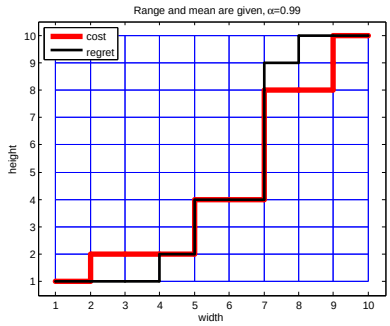
(d) $[\underline{c}_i, \overline{c}_i]$, μ_i and δ_i are given, $\alpha = 0.5$



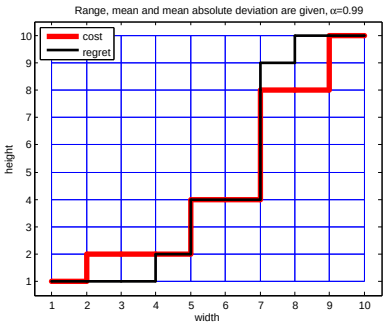
(e) $[\underline{c}_i, \overline{c}_i]$ and μ_i are given, $\alpha = 0.8$



(f) $[\underline{c}_i, \overline{c}_i]$, μ_i and δ_i are given, $\alpha = 0.8$



(g) $[\underline{c}_i, \overline{c}_i]$ and μ_i are given, $\alpha = 0.99$



(h) $[\underline{c}_i, \overline{c}_i]$, μ_i and δ_i are given, $\alpha = 0.99$

Figure 2.5: Optimal paths that minimize the WCVaR of cost and regret

of α, the Min-WCVaR-cost path and the Min-WCVaR-regret path are different. The optimal paths also differ based on whether information on the mean absolute deviation is available or not.

Chapter 3

Polynomially Solvable Instances

In this chapter, we design a polynomial time algorithm to solve the problem of minimizing the the WCVaR of regret for the subset selection problem. This extends a known polynomial complexity result for the minmax regret subset selection problem with range information only. The idea in the design of this algorithm is also used to the distributionally robust k-sum optimization problem.

Structure of the chapter: In Section 3.1, we review the existing polynomial time algorithms for the minmax regret subset selection problem. In Section 3.2, a polynomial time algorithm for the problem of minimizing the WCVaR of regret for the subset selection problem is developed. In Section 3.3, some numerical results are given to compare the proposed polynomial algorithm with the commercial solver CPLEX for solving the corresponding mixed integer linear programming problem. In Section 3.4, we generalize this polynomial time algorithm to the distributionally robust k-sum optimization problem with uncertainty.

3.1 Polynomial Time Algorithm of the Minmax Regret Subsect Selection Problem

In this section, we review the polynomial time algorithm to solve the minmax regret subset selection problem. The deterministic subset selection problem is to choose a subset of K items of maximum total weight. Let c_i denote the weight of item i, $i \in [N]$, the problem is

$$Z(\boldsymbol{c}) = \max_{\boldsymbol{x} \in \mathcal{X}} \boldsymbol{c}^T \boldsymbol{x}, \tag{3.1}$$

where

$$\mathcal{X} = \{\boldsymbol{x} \in \{0,1\}^N \mid \sum_{i=1}^{N} x_i = K\}. \tag{3.2}$$

This problem can also be viewed as a special case of 0-1 knapsack in which the capacities of all items are equal to 1. The optimal solution to the deterministic subset selection problem can be obtained in $O(N)$ time by first selecting K-th largest weighted item and by selecting then $K-1$ items of the weight greater than or equal to the weight of the K-th largest weighted item. Assume the parameter vector $\boldsymbol{c}$ is uncertain and lies in an interval uncertainty set Ω, that is $\Omega = \Omega_1 \times \Omega_2 \times \cdots \times \Omega_N$, where $\Omega_i = [\underline{c}_i, \overline{c}_i]$ for all $i \in [N]$. The minmax regret subset selection problem is

$$\min_{\boldsymbol{x} \in \mathcal{X}} \max_{\boldsymbol{c} \in \Omega} (Z(\boldsymbol{c}) - \boldsymbol{c}^T \boldsymbol{x}), \tag{3.3}$$

where $\mathcal{X}$ is defined as in (3.2).

Up to present, the minmax regret subset selection problem is one of the few polynomially solvable minmax regret combinatorial optimization problems. With the interval uncertainty representation of the weights, Averbakh [9] first designed a polynomial algorithm to solve the minmax regret subsection problem to optimality with a running time of $O(N \min(K, N-K)^2)$. Subsequently, Conde [37] designed an improved algorithm to solve this problem with running time $O(N \min(K, N-K))$. Next, we discuss the polynomial algorithm with faster running time, which is due to Conde [37].

First the number of selected items K can be assumed to be no larger than $N/2$ without loss of generality. Otherwise the problem can be transformed in $O(N)$ time to an equivalent problem by selecting $N-K$ items from N with weight $-c_i$, $i \in [N]$. Since in the interval uncertainty case, the worst-case scenario is

$$c_i = \underline{c}_i x_i + \overline{c}_i(1-x_i), \quad i \in [N]$$

the minmax regret problem (3.3) can be formulated as

$$\min_{\boldsymbol{x}\in\mathcal{X}} \left\{ \max_{\boldsymbol{y}\in\mathcal{X}} \sum_{i=1}^{N} \left([\underline{c}_i x_i + \overline{c}_i(1-x_i)]y_i - \underline{c}_i x_i \right) \right\}. \tag{3.4}$$

The subproblem $\max_{\boldsymbol{y}\in\mathcal{X}} \sum_{i=1}^{N} [\underline{c}_i x_i + \overline{c}_i(1-x_i)]y_i$ is equivalent to its LP relaxation

$$\max \left\{ \sum_{i=1}^{N} [\underline{c}_i x_i + \overline{c}_i(1-x_i)]y_i : \sum_{i=1}^{N} y_i = K, 0 \le y_i \le 1, i \in [N] \right\}. \tag{3.5}$$

By considering the dual problem of (3.5), the minmax regret problem (3.3) can be formulated as the following mixed integer linear programming problem

$$\begin{aligned} \min \quad & \sum_{i=1}^{N} \lambda_i + K\lambda_0 - \sum_{i=1}^{N} \underline{c}_i x_i \\ \text{s.t.} \quad & \lambda_i \ge \underline{c}_i x_i + \overline{c}_i(1-x_i) - \lambda_0, \ i \in [N], \\ & \lambda_i \ge 0, \ i \in [N], \\ & \boldsymbol{x} \in \mathcal{X}. \end{aligned} \tag{3.6}$$

Observe that the objective of (3.6) is an increasing function of all $\lambda_i, i \in [N]$. Thus every optimal solution to (3.6) must satisfy

$$\lambda_i = [\underline{c}_i x_i + \overline{c}_i(1-x_i) - \lambda_0]^+, \quad i \in [N].$$

Hence (3.6) is equivalent to

$$\min_{\lambda_0\in\Re, \boldsymbol{x}\in\mathcal{X}} \left\{ \sum_{i=1}^{N} [\underline{c}_i x_i + \overline{c}_i(1-x_i) - \lambda_0]^+ + K\lambda_0 - \sum_{i=1}^{N} \underline{c}_i x_i. \right\} \tag{3.7}$$

Conde [37] proved that the optimal λ_0 can be chosen from a set which has $2K+1$ elements.

Proposition 3.1 (Conde [37]). *Consider the sorted sequences of the bounds of the interval weights*

$$\underline{c}_{[1]} \geq \underline{c}_{[2]} \geq \cdots \geq \underline{c}_{[N]}$$
$$\overline{c}_{[1]} \geq \overline{c}_{[2]} \geq \cdots \geq \overline{c}_{[N]},$$

the objective function of (3.7) attains minimum for

$$\lambda_0 \in \Lambda := \{\underline{c}_{[1]}, \ldots, \underline{c}_{[K]}\} \cup \{\overline{c}_{[K]}, \ldots, \overline{c}_{[2K]}\}.$$

Let us now additionally transform problem (3.7). Since $K = \sum_{i=1}^{N} x_i$, (3.7) can be written as

$$\begin{aligned} & \min_{\lambda_0 \in \Lambda, \boldsymbol{x} \in \mathcal{X}} \ \sum_{i=1}^{N} \Big([\underline{c}_i x_i + \overline{c}_i (1 - x_i) - \lambda_0]^+ + (\lambda_0 - \underline{c}_i) x_i \Big) \\ = & \min_{\lambda_0 \in \Lambda, \boldsymbol{x} \in \mathcal{X}} \ \sum_{i=1}^{N} \Big[(\underline{c}_i - \lambda_0)^+ x_i - (\underline{c}_i - \lambda_0) x_i + (\overline{c}_i - \lambda_0)^+ (1 - x_i) \Big] \\ = & \min_{\lambda_0 \in \Lambda, \boldsymbol{x} \in \mathcal{X}} \ \sum_{i=1}^{N} \Big[[(\lambda_0 - \underline{c}_i)^+ - (\overline{c}_i - \lambda_0)^+] x_i + (\overline{c}_i - \lambda_0)^+ \Big]. \end{aligned}$$

Let $a_i(\lambda_0) = (\lambda_0 - \underline{c}_i)^+ - (\overline{c}_i - \lambda_0)^+$ and $b_i(\lambda_0) = (\overline{c}_i - \lambda_0)^+$ for all $i \in [N]$. Finally, problem (3.7) can be written as

$$\min_{\lambda_0 \in \Lambda} \{ f(\lambda_0) := \min_{\boldsymbol{x} \in \mathcal{X}} \sum_{i=1}^{N} [a_i(\lambda_0) x_i + b_i(\lambda_0)] \}. \tag{3.8}$$

Based on the above formulation, the algorithm in Conde [37] is quite straightforward. For every λ_0 in the set $\Lambda := \{\underline{c}_{[1]}, \ldots, \underline{c}_{[K]}\} \cup \{\overline{c}_{[K]}, \ldots, \overline{c}_{[2K]}\}$, we solve a deterministic subset selection problem $\min_{\boldsymbol{x} \in \mathcal{X}} \sum_{i=1}^{N} a_i(\lambda_0) x_i$. Then choose the smallest one from $\{f(\lambda_0) \mid \lambda_0 \in \Lambda\}$. The complexity of this algorithm is $O(NK)$. In the above algorithm, we assumed $K \leq N/2$. If $K > N/2$, we should first transform the problem to an equivalent subset selection problem by selecting $N - K$ items. The complexity of this transformation is $O(N)$. Finally, a polynomial algorithm with complexity $O(N \min(K, N-K))$ for the minmax regret subset selection problem is obtained.

3.2 Polynomial Solvability for the Probabilistic Regret Model in Subset Selection

In this section, we identify a polynomial time algorithm to solve the probabilistic regret model for subset selection. Assume that the weight vector $\tilde{\boldsymbol{c}}$ for a set of items $\{1,\ldots,N\}$ is random. The marginal distribution of each $\tilde{c}_i$ is given as P_i. In the deterministic subset selection problem, the objective is to choose a subset of K items of maximum total weight. In the probabilistic regret model, the objective is to minimize the WCVaR of regret. This problem is formulated as

$$\min_{x\in\mathcal{X}} \mathrm{WCVaR}_\alpha\left(Z(\tilde{\boldsymbol{c}}) - \tilde{\boldsymbol{c}}^T\boldsymbol{x}\right), \tag{3.9}$$

where the feasible region is:

$$\mathcal{X} = \left\{ x \in \{0,1\}^N : \sum_{i=1}^N x_i = K \right\}.$$

For the subset selection problem, $Z(\cdot)$ is computed as the optimal objective value to the linear program:

$$Z(\boldsymbol{c}) = \max\left\{\boldsymbol{c}^T\boldsymbol{y} \mid \boldsymbol{e}^T\boldsymbol{y} = K, 0 \le \boldsymbol{y} \le \boldsymbol{e}\right\}.$$

Strong duality of linear programming implies that it can be reformulated as:

$$Z(\boldsymbol{c}) = \min\left\{\boldsymbol{e}^T\boldsymbol{\lambda} + K\lambda_0 \mid \boldsymbol{\lambda} \ge \boldsymbol{c} - \lambda_0\boldsymbol{e}, \boldsymbol{\lambda} \ge 0\right\} = \min_{\lambda_0} \sum_{i=1}^N (c_i - \lambda_0)^+ + K\lambda_0.$$

When the marginal distributions P_i, $i \in [N]$ for the random vector $\tilde{\boldsymbol{c}}$ are given, using Proposition 2.4, the probabilistic regret model (3.9) for subset selection is formulated as:

$$\min_{\lambda_0,\boldsymbol{x}\in\mathcal{X},\boldsymbol{d}\in\Omega} \left(\sum_{i=1}^N [d_i - \lambda_0]^+ + K\lambda_0 + \frac{\alpha}{1-\alpha}\boldsymbol{d}^T\boldsymbol{x} - \frac{1}{1-\alpha}\boldsymbol{\mu}^T\boldsymbol{x} + \frac{1}{1-\alpha}\sum_{i=1}^N \mathbb{E}_{P_i}[\tilde{c}_i - d_i]^+ \right). \tag{3.10}$$

Observe that for a fixed λ_0, the objective function of (3.10) is separable in d_i. Define

$$F_i(d_i, x_i, \lambda_0) = [d_i - \lambda_0]^+ + \frac{\alpha}{1-\alpha} d_i x_i + \frac{1}{1-\alpha}\mathbb{E}_{P_i}[\tilde{c}_i - d_i]^+ - \frac{1}{1-\alpha}\mu_i x_i.$$

Then problem (3.10) is expressed as:

$$\min_{\lambda_0, \boldsymbol{x}\in\mathcal{X}, \boldsymbol{d}\in\Omega} \sum_{i=1}^{N} F_i(d_i, x_i, \lambda_0) + K\lambda_0. \tag{3.11}$$

For fixed λ_0 and x_i, $F_i(d_i, x_i, \lambda_0)$ is a convex function of d_i. Denote a minimizer of this function as $d_i^*(x_i, \lambda_0) = \operatorname{argmin}_{d_i\in\Omega_i} F_i(d_i, x_i, \lambda_0)$. Define the minimizers:

$$a_i(\lambda_0) = \operatorname*{argmin}_{d_i\in\Omega_i} F_i(d_i, 1, \lambda_0), \qquad b_i(\lambda_0) = \operatorname*{argmin}_{d_i\in\Omega_i} F_i(d_i, 0, \lambda_0).$$

Since $x_i \in \{0, 1\}$, this implies:

$$d_i^*(x_i, \lambda_0) = a_i(\lambda_0)x_i + b_i(\lambda_0)(1 - x_i).$$

For simplicity, we will denote $a_i(\lambda_0), b_i(\lambda_0)$ and $d_i^*(x_i, \lambda_0)$ by a_i, b_i and d_i^* by dropping the explicit dependence on the parameters. By substituting in the expression for d_i^* with the observation that $x_i \in \{0, 1\}$, we have

$$\begin{aligned}
&F_i(d_i^*, x_i, \lambda_0)\\
=&(a_i - \lambda_0)^+ x_i + (b_i - \lambda_0)^+(1 - x_i) + \frac{\alpha}{1-\alpha} a_i x_i\\
&+ \frac{1}{1-\alpha}\mathbb{E}_{P_i}\Big[(\tilde{c}_i - a_i)^+ x_i + (\tilde{c}_i - b_i)^+(1 - x_i)\Big] - \frac{1}{1-\alpha}\mu_i x_i\\
=&\Big((a_i - \lambda_0)^+ - (b_i - \lambda_0)^+ + \frac{\alpha}{1-\alpha} a_i + \frac{1}{1-\alpha}\mathbb{E}_{P_i}[(\tilde{c}_i - a_i)^+ - (\tilde{c}_i - b_i)^+] - \frac{1}{1-\alpha}\mu_i\Big) x_i\\
&+ (b_i - \lambda_0)^+ + \frac{1}{1-\alpha}\mathbb{E}_{P_i}[\tilde{c}_i - b_i]^+.
\end{aligned}$$

Define an N dimensional vector $\boldsymbol{h}(\lambda_0)$ and a scalar $h_0(\lambda_0)$ with

$$\begin{aligned}
h_i(\lambda_0) =&(a_i - \lambda_0)^+ - (b_i - \lambda_0)^+ + \frac{\alpha}{1-\alpha} a_i + \frac{1}{1-\alpha}\mathbb{E}_{P_i}[(\tilde{c}_i - a_i)^+ - (\tilde{c}_i - b_i)^+]\\
&- \frac{1}{1-\alpha}\mu_i, \quad i \in [N],\\
h_0(\lambda_0) =&\sum_{i=1}^{N}(b_i - \lambda_0)^+ + \frac{1}{1-\alpha}\sum_{i=1}^{N}\mathbb{E}_{P_i}[\tilde{c}_i - b_i]^+ + K\lambda_0.
\end{aligned}$$

Problem (3.10) is thus reformulated as:

$$\min_{\lambda_0}\min_{\boldsymbol{x}\in\mathcal{X}} \boldsymbol{h}(\lambda_0)^T\boldsymbol{x} + h_0(\lambda_0). \tag{3.12}$$

For a fixed λ_0, the inner optimization problem of picking K smallest items from N can be done efficiently in $O(N)$ time. The next proposition shows that for discrete marginal distributions, the search for the optimal value of λ_0 can be restricted to a finite set.

Proposition 3.2. *Assume that the marginal distribution of $\tilde{c}_i$ is discrete and*

$$\tilde{c}_i \sim c_{ij} \text{ with probability } p_{ij}, \quad j \in [J_i], i \in [N].$$

The objective function of (3.12) attains its minimum in the finite set:

$$\lambda_0 \in \{c_{ij} \mid j \in [J_i], i \in [N]\}.$$

Proof. For discrete marginal distributions, problem (3.10) is formulated as:

$$\min_{\lambda_0, \boldsymbol{x}\in\mathcal{X}, \boldsymbol{d}\in\Omega} \left(\sum_{i=1}^{N} [d_i - \lambda_0]^+ + K\lambda_0 + \frac{\alpha}{1-\alpha} \boldsymbol{d}^T \boldsymbol{x} + \frac{1}{1-\alpha} \Big[\sum_{i=1}^{N} \sum_{j=1}^{J_i} (c_{ij} - d_i)^+ p_{ij} - \boldsymbol{\mu}^T \boldsymbol{x} \Big] \right). \tag{3.13}$$

For a fixed vector $\boldsymbol{d}$, sort the components of the vector such that $d^{(1)} \geq d^{(2)} \geq \ldots d^{(N)}$. Let $\lambda_0^* = d^{(K)}$ be the K-th largest component of $\boldsymbol{d}$. Then

$$\begin{aligned} \min_{\lambda_0} \left(\sum_{i=1}^{N} [d_i - \lambda_0]^+ + K\lambda_0 \right) &= \max \left\{ \boldsymbol{d}^T \boldsymbol{y} : \sum_{i=1}^{N} y_i = K, 0 \leq y_i \leq 1, i \in [N] \right\} \\ &= \sum_{i=1}^{K} d^{(i)} \\ &= \sum_{i=1}^{N} [d_i - \lambda_0^*]^+ + K\lambda_0^*, \end{aligned}$$

where the first equality comes from linear programming duality. Hence the minimizer λ_0 can be chosen as the K-th largest component of $\boldsymbol{d}$. We claim that for each $i \in [N]$, the i-th component of all the optimal $\boldsymbol{d}$ can be chosen in the set $\{c_{ij} \mid j \in [J_i]\}$. To prove this claim, the problem of minimizing the WCVaR of regret is formulated as:

$$\min_{\boldsymbol{x}\in\mathcal{X}} \min_{\boldsymbol{d}\in\Omega} \max_{\boldsymbol{y}\in \mathrm{conv}(\mathcal{X})} \left(\boldsymbol{d}^T \boldsymbol{y} + \frac{\alpha}{1-\alpha} \boldsymbol{d}^T \boldsymbol{x} - \frac{1}{1-\alpha} \boldsymbol{\mu}^T \boldsymbol{x} + \frac{1}{1-\alpha} \sum_{i=1}^{N} \sum_{j=1}^{J_i} p_{ij} [c_{ij} - d_i]^+ \right)$$

$$= \min_{\boldsymbol{x}\in\mathcal{X}} \max_{\boldsymbol{y}\in\mathrm{conv}(\mathcal{X})} \min_{\boldsymbol{d}\in\Omega} \left(\boldsymbol{d}^T\boldsymbol{y} + \frac{\alpha}{1-\alpha}\boldsymbol{d}^T\boldsymbol{x} - \frac{1}{1-\alpha}\boldsymbol{\mu}^T\boldsymbol{x} + \frac{1}{1-\alpha}\sum_{i=1}^{N}\sum_{j=1}^{J_i} p_{ij}[c_{ij}-d_i]^+ \right)$$

$$= \min_{\boldsymbol{x}\in\mathcal{X}} \max_{\boldsymbol{y}\in\mathrm{conv}(\mathcal{X})} \sum_{i=1}^{N} \left[\min_{d_i\in\Omega_i} \left(d_i \left(y_i + \frac{\alpha}{1-\alpha}x_i \right) + \frac{1}{1-\alpha}\sum_{j=1}^{J_i} p_{ij}[c_{ij}-d_i]^+ \right) - \frac{1}{1-\alpha}\mu_i x_i \right], \tag{3.14}$$

where $\mathrm{conv}(\mathcal{X})$ is the convex hull of the set $\mathcal{X}$. For fixed $\boldsymbol{x}$ and $\boldsymbol{y}$, the function $d_i(y_i + \frac{\alpha}{1-\alpha}x_i) + \frac{1}{1-\alpha}\sum_{j=1}^{J_i} p_{ij}[c_{ij}-d_i]^+$ is a piecewise linear function in d_i, and its minimum value over $d_i \in \Omega_i$ occurs at one of the break points $\{c_{ij} \mid j \in [J_i]\}$. Since the optimal λ_0 is the K-th largest component of the optimal $\boldsymbol{d}$, the result holds. □

By combining Proposition 3.2 and formulation (3.12), we provide a polynomial time algorithm to minimize the WCVaR of regret for the subset selection problem.

The algorithm is described as follows:

Algorithm 1: Minimization of the WCVaR of regret for subset selection.

Input: K, probability level α, discrete marginal distribution

$$c_{ij}, p_{ij}, j \in [J_i], i \in [N].$$

Output: Optimal decision $\boldsymbol{x}$, the minimum WCVaR of regret obj.

1 Sort $\{c_{ij}\}_{j\in[J_i],i\in[N]}$ as an increasing sequence in the set Λ.

2 Delete the repeated numbers in Λ to get a new set Λ_0.

3 $\boldsymbol{x} = 0$, $obj = \infty$

4 **for** $\lambda_0 \in \Lambda_0$ **do**

5 | **for** $i = 1, \ldots, N$ **do**

6 | | $a_i = \operatorname{argmin}_{d_i\in\Omega_i} F(d_i, 1, \lambda_0)$, $b_i = \operatorname{argmin}_{d_i\in\Omega_i} F(d_i, 0, \lambda_0)$,

7 | | $h_i = (a_i - \lambda_0)^+ - (b_i - \lambda_0)^+ + \frac{\alpha}{1-\alpha} a_i + \frac{1}{1-\alpha}\sum_{j=1}^{J_i}[(c_{ij} - a_i)^+ - (c_{ij} - b_i)^+]p_{ij} - \frac{1}{1-\alpha}\mu_i$,

8 | **end**

9 | $h_0 = \sum_{i=1}^{N}(b_i - \lambda_0)^+ + \frac{1}{1-\alpha}\sum_{i=1}^{N}\sum_{j=1}^{J_i}(c_{ij} - b_i)^+ p_{ij} + K\lambda_0$.

10 | $\boldsymbol{y} = \operatorname{argmin}_{\boldsymbol{x}\in\mathcal{X}} \boldsymbol{h}^T\boldsymbol{x}$, $val = \boldsymbol{h}^T\boldsymbol{y} + h_0$.

11 | **if** $val < obj$ **then**

12 | | $\boldsymbol{x} = \boldsymbol{y}$, $obj = val$.

13 | **end**

14 **end**

Proposition 3.3. *The running time of Algorithm 1 is $O(N^2 J_{max}^2)$ where $J_{max} = \max_{i\in[N]} J_i$. The algorithm solves formulation (3.9) to optimality.*

Proof. Sorting in step 1 can be done in $O(NJ_{max}\log(NJ_{max}))$. The function $F(d_i, 1, \lambda_0)$ is a piecewise linear function with respect to d_i. To get the optimal d_i, the values of $F(d_i, 1, \lambda_0)$ are evaluated at the break points c_{ij}, $j \in [J_i]$ and λ_0. The complexity of computing a_i is thus $O(J_i)$. Likewise for b_i. The complexity of computing the vector $\boldsymbol{h}(\lambda_0)$ and the scalar $h_0(\lambda_0)$ in steps 5 to 9 is thus $O(NJ_{max})$. For subset selection, the complexity of finding $\operatorname{argmin}_{\boldsymbol{x}\in\mathcal{X}} \boldsymbol{h}^T\boldsymbol{x}$ is $O(N)$. Moreover, $|\Lambda_0| \leq NJ_{max}$, hence the total computational complexity for Algorithm 1 is

$O(N^2 J_{max}^2)$. □

In the marginal moment model, if (a) the mean and range are given, the worst-case marginal distribution is a two-point discrete distribution; and (b) the mean, range and mean absolute deviation are given, the worst-case distribution is a three-point discrete distribution. These worst-case marginal distributions are fixed and can hence be treated as a special case of the discrete marginal distribution model. Thus in these two cases, Algorithm 1 solves the problem to optimality which brings us to the following result.

Theorem 3.4. *The problem of minimizing the* WCVaR *of regret for the subset selection problem is solvable in polynomial time with complexity* $O(N^2)$ *when (a) the range and mean, and (b) the range, mean and mean absolute deviation are given.*

This extends the polynomial complexity result when only the range is given (see Averbakh [9] and Conde [37]). Algorithm 1 is related to the earlier algorithms of Averbakh [9] and Conde [37] for the range case. When only the $[\underline{c}_i, \overline{c}_i], i \in [N]$ of each $\tilde{c}_i$ is known, the problem of minimizing the WCVaR of the regret reduces to the interval uncertainty minmax regret problem. In this case, the worst-case marginal distribution is the Dirac measure $\delta_{\hat{\boldsymbol{c}}(\boldsymbol{x})}$, where $\hat{c}_i(\boldsymbol{x}) = \underline{c}_i x_i + \overline{c}_i(1 - x_i)$. It is easy to check that the variables in Algorithm 1 are then $a_i = \underline{c}_i$, $b_i = \overline{c}_i$, $h_i = [\underline{c}_i - \lambda_0]^+ - [\overline{c}_i - \lambda_0]^+ - \underline{c}_i, i \in [N]$, and $h_0 = \sum_{i=1}^N [\overline{c}_i - \lambda_0]^+ + K\lambda_0$. The running time of Algorithm 1 is $O(N^2)$ in this case. Since the optimal λ_0 is the Kth largest value of the optimal $d_i^*(x_i) = \underline{c}_i(x_i) + \overline{c}_i(x_i)$, the feasible set Λ_0 can be further reduced to a smaller set with cardinality $2K+1$ (see the discussion in Conde [37]). Furthermore, if $K > N/2$ the problem can be transformed in $O(N)$ time to an equivalent problem with $K' \leq N/2$ (see Averbakh [9]). Hence by reducing the size of the feasible set Λ_0, the complexity of Algorithm 1 can be reduced to $O(N \min(K, N-K)\ )$ when only the range of $\tilde{c}_i$ is given, $i \in [N]$. Algorithm 1 is thus a generalization of the algorithms for the minmax regret subset selection

problem.

3.3 Numerical Examples

In this section, we compare the proposed polynomial algorithm with the commercial solver CPLEX to solve the problem of minimizing the WCVaR of regret for the subset selection problem.

Example 3.1. *In this experiment, the interval range for each item* $[\underline{c}_i, \overline{c}_i]$ *are randomly generated with* $\underline{c}_i = \min\{a_i, b_i\}$, $\overline{c}_i = \max\{a_i, b_i\}$, *with* a_i, b_i *generated from the uniform distribution* $U[0, 100]$. *The mean for each item is randomly generated as* $\mu_i \sim U[\underline{c}_i, \overline{c}_i]$. *Define* $\bar{\delta}_i = 2\frac{(\overline{c}_i - \mu_i)(\mu_i - \underline{c}_i)}{\overline{c}_i - \underline{c}_i}$ *as the largest mean absolute deviation when the mean and range of* $\tilde{c}_i$ *are given. Let the mean absolute deviation of* $\tilde{c}_i$ *be randomly generated by* $\delta_i \sim U[0, \bar{\delta}_i]$. *We test* **Algorithm 1** *for the following two cases of the marginal moment model: (a) range* $[\underline{c}_i, \overline{c}_i]$ *and mean* μ_i *are given and (b) range* $[\underline{c}_i, \overline{c}_i]$, *mean* μ_i *and mean absolute deviation* δ_i *are given.*

The algorithm was implemented in Matlab R2012a on an Intel Core 2 Duo CPU 2.8G Hz laptop with 4 GB of RAM. To compare the efficiency of Algorithm 1 with CPLEX's MIP solver (version 12.4), randomly generated instances were tested for different α's and K's. For the CPLEX optional parameters, the default values are used. We compare the CPU times of the two methods in the following tables. First, we fix the value of α and K, and compare the CPU time for different N. Then, we fix the value of the dimension N, and tested the sensitivity of the running time of Algorithm 1 to the parameters α and K. In the tables, the CPU time (in seconds) taken by Algorithm 1 to solve (3.10) and CPLEX's MIP solver to solve (2.37) are denoted by "time Alg1" and "time Cplex", respectively. The CPU time in the tables was the average execution time of 10 randomly generated instances. The instances with "$**$" indicates that CPLEX ran out of memory. From Table 3.1, it is clear that the CPU time taken by Algorithm 1 is significantly smaller than that taken by CPLEX's MIP solver. Even for extremely large values

of N, Algorithm 1 was able to solve the problem to optimality in a reasonable amount of time (see Table 3.2). From Figure 3.1, the running time of CPLEX increases dramatically as certain values of K. This has to be expected because the number of feasible combinations increases when K is close to $N/2$. The running time of CPLEX also increases as α increases. However, the running of Algorithm 1 is quite stable and insensitive to the parameters K and α. Other examples with different dimensions have been tested, and similar results holds. Hence, Algorithm 1 is very robust and efficient.

Table 3.1: Computational results for $\alpha = 0.3, K = 0.4N$.

	(a) $[\underline{c}_i, \overline{c}_i], \mu_i$ are given		(b) $[\underline{c}_i, \overline{c}_i], \mu_i, \delta_i$ are given	
N	time Alg1	time Cplex	time Alg1	time Cplex
50	7.80e-003	2.06e-001	1.23e-002	1.89e-001
100	1.72e-002	2.76e-001	3.73e-002	2.42e-001
200	5.65e-002	5.76e-001	9.52e-002	3.42e-001
400	1.58e-001	1.53e+000	2.98e-001	7.04e-001
800	5.23e-001	**	9.98e-001	**

Table 3.2: CPU time of Algorithm 1 for solving large instances ($\alpha = 0.9, K = 0.3N$).

N	(a) $[\underline{c}_i, \overline{c}_i], \mu_i$ are given	(b) $[\underline{c}_i, \overline{c}_i], \mu_i, \delta_i$ are given
5000	1.79e+001	3.55e+001
10000	7.02e+001	1.40e+002
20000	2.56e+002	5.25e+002
40000	1.02e+003	2.28e+003
80000	4.91e+003	1.07e+004

3.4 Distributionally Robust k-sum Optimization

The k-sum optimization problem is the combinatorial problem of finding a solution such that the sum of the k largest weighted elements of the solution is as small as possible,

$$\min_{\boldsymbol{x}\in\mathcal{X}} \max_{\boldsymbol{y}\in\mathcal{Y}} \Big\{ \sum_{i=1}^{N} c_i x_i y_i \Big\} \tag{3.15}$$

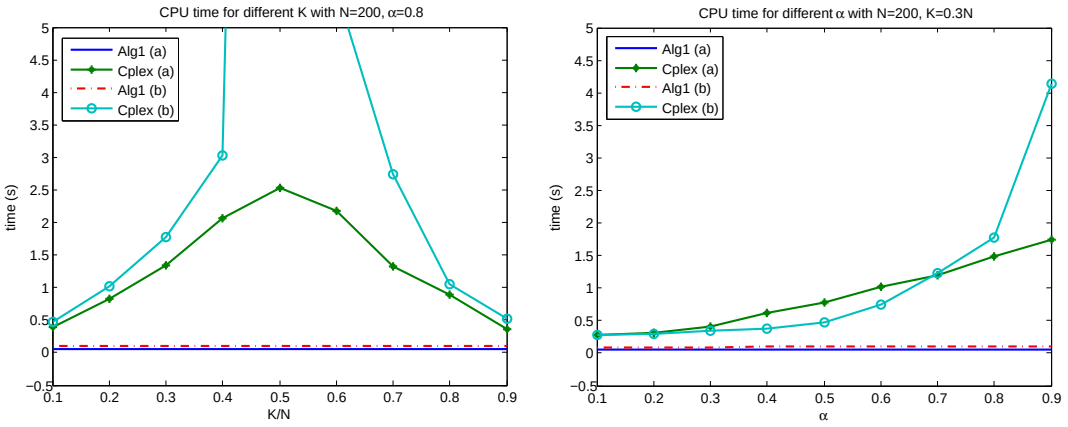


Figure 3.1: Sensitivity to the parameters K and α

where $\mathcal{X} \subseteq \{0,1\}^N$ is the feasible set of the optimization problem and $\mathcal{Y} = \{\boldsymbol{y} : \sum_{i=1}^N y_i = k, y_i \in \{0,1\}, \forall i \in [N]\}$. Notice that when $k = N$, (3.15) reduces to the standard linear sum optimization problem $\min_{\boldsymbol{x}\in\mathcal{X}} \boldsymbol{c}^T\boldsymbol{x}$. When $k = 1$, (3.15) reduces to the bottleneck optimization problem

$$\min_{\boldsymbol{x}\in\mathcal{X}} \max_{i\in[N]} \{c_i x_i\}. \tag{3.16}$$

Several efficient algorithms for the bottleneck and k-sum optimization problems have been developed. Gupta and Punnen [63] showed that the k-sum problem can be solved by solving $O(N)$ linear sum problems. Hence the k-sum optimization problem can be solved in polynomial time whenever the associated linear sum problem can be solved in polynomial time. Furthermore, Punnen and Aneja [100] showed that if the linear sum problem is solved by a polynomial time ϵ-approximation scheme then the k-sum problem can also be solved by a polynomial time ϵ-approximation scheme. As a special case of the k-sum optimization problem, the bottleneck optimization problem has been studied by several authors. A bottleneck location problem was considered in Hsu and Nemhauser [70], and an efficient algorithm was proposed. Gabow and Tarjan [49] developed two polynomial time for the bottleneck spanning tree problem in a directed graph and the the bottleneck maximum cardinality matching problem. With data uncertainty, stochastic bottleneck assignment, transportation and spanning tree problems have

been considered in [122, 52, 71]. With partial distributional information under the marginal distribution and marginal moment models, we consider the distributionally robust k-sum optimization problem in this section.

Assume the cost vector $\boldsymbol{c}$ is random and denoted by $\tilde{\boldsymbol{c}}$, and its distribution lies in a distribution set $\mathbb{P}$. The distributionally robust k-sum optimization problem is as follows:

$$Z^* = \min_{\boldsymbol{x}\in\mathcal{X}} \sup_{P\in\mathbb{P}} \mathbb{E}_P\Big[\max_{\boldsymbol{y}\in\mathcal{Y}} \sum_{i=1}^{N} \tilde{c}_i x_i y_i\Big]. \tag{3.17}$$

If the marginal distributions of $\tilde{\boldsymbol{c}}$ are given, let $\mathbb{P}(P_1,\ldots,P_N)$ denote the set of joint distributions with the fixed marginals. Then problem (3.17) can be written as

$$\begin{aligned} Z^* &= \min_{\boldsymbol{x}\in\mathcal{X}} \sup_{P\in\mathbb{P}(P_1,\ldots,P_N)} \mathbb{E}_P\Big[\max_{\boldsymbol{y}\in\mathcal{Y}} \sum_{i=1}^{N} \tilde{c}_i x_i y_i\Big] \\ &= \min_{\boldsymbol{x}\in\mathcal{X}} \min_{\boldsymbol{d}\in\Re^N} \Big(\max_{\boldsymbol{y}\in\mathcal{Y}} \boldsymbol{d}^T\boldsymbol{y} + \sum_{i=1}^{N} \mathbb{E}_{P_i}[\tilde{c}_i x_i - d_i]^+\Big) \\ &= \min_{\boldsymbol{x}\in\mathcal{X}} \min_{\boldsymbol{d}\in\Re^N,\lambda_0\in\Re} \Big(\sum_{i=1}^{N}[d_i-\lambda_0]^+ + k\lambda_0 + \sum_{i=1}^{N} \mathbb{E}_{P_i}[\tilde{c}_i x_i - d_i]^+\Big) \\ &= \min_{\boldsymbol{x}\in\mathcal{X}} \min_{\lambda_0} \Big(k\lambda_0 + \sum_{i=1}^{N} \mathbb{E}_{P_i}[\tilde{c}_i x_i - \lambda_0]^+\Big). \end{aligned} \tag{3.18}$$

In the above formula, the optimal λ_0 is the kth largest value of d_i. For $d_i \le \lambda_0$, we can increase these d_i to the value λ_0, and for $d_i \ge \lambda_0$, we can decrease these d_i to λ_0. By this modification, the objective value will not increase. Hence the last equality of (3.18) holds.

If $k = N$, the optimal λ_0 in (3.18) can be chosen as small as possible, and the distributionally robust k-sum optimization problem reduces to

$$\min_{\boldsymbol{x}\in\mathcal{X}} \sup_{P\in\mathbb{P}(P_1,\ldots,P_N)} \mathbb{E}_P[\tilde{\boldsymbol{c}}^T\boldsymbol{x}] = \min_{\boldsymbol{x}\in\mathcal{X}} \boldsymbol{\mu}^T\boldsymbol{x},$$

where $\mu_i = \mathbb{E}_{P_i}(\tilde{c}_i), i \in [N]$, which is a deterministic linear sum optimization problem. We focus on $1 \le k < N$. As $\lambda_0 \to \infty$, the objective value $\to \infty$, and as $\lambda_0 \to -\infty$ the objective value $\to \infty$ if $k < N$. Hence the optimal λ_0 exists. Next

we show that if the given marginal distributions are discrete, then the optimal λ_0 can be restricted to a finite set.

Proposition 3.5. *Assume that the marginal distribution of $\tilde{c}_i$ is discrete and*

$$\tilde{c}_i \sim c_{ij} \text{ with probability } p_{ij}, \quad j \in [J_i], i \in [N].$$

the objective function for (3.18) attains its minimum in the finite set:

$$\lambda_0 \in \{0\} \cup \{c_{ij} \mid j \in [J_i], i \in [N]\}.$$

Proof. In the discrete marginal distribution case, problem (3.18) can be written as

$$\begin{aligned} Z^* &= \min_{\boldsymbol{x}\in\mathcal{X}} \min_{\lambda_0} \Big(k\lambda_0 + \sum_{i=1}^{N}\sum_{j=1}^{J_i} [c_{ij}x_i - \lambda_0]^+ p_{ij} \Big) \\ &= \min_{\boldsymbol{x}\in\mathcal{X}} \min_{\lambda_0} \Big(k\lambda_0 + \sum_{i=1}^{N}\sum_{j=1}^{J_i} \big\{ [c_{ij} - \lambda_0]^+ p_{ij}x_i + [-\lambda_0]^+ p_{ij}(1-x_i) \big\} \Big) \end{aligned} \tag{3.19}$$

For fixed $\boldsymbol{x} \in \mathcal{X}$ the objective function in (3.19) is a piece-wise linear convex function in λ_0. Since the objective value goes to infinity as λ_0 goes to infinity or negative infinity, hence its minimum value occurs at one of the break points $\{0\} \cup \{c_{ij} \mid j \in [J_i], i \in [N]\}$. □

Let $\Lambda_0 = \{0\} \cup \{c_{ij} \mid j \in [J_i], i \in [N]\}$, and

$$h_i(\lambda_0) = \sum_{j=1}^{J_i} ([c_{ij} - \lambda_0]^+ - [-\lambda_0]^+) p_{ij}, \quad i \in [N], \tag{3.20}$$

$$h_0(\lambda_0) = N[-\lambda_0]^+ + k\lambda_0. \tag{3.21}$$

Then problem (3.19) can be written as

$$\min_{\lambda_0\in\Lambda_0} \min_{\boldsymbol{x}\in\mathcal{X}} \sum_{i=1}^{N} h(\lambda_0)x_i + h_0(\lambda_0). \tag{3.22}$$

Hence we can solve the distributionally robust k-sum optimization problem as follows: For all $\lambda_0 \in \Lambda_0$, solve the deterministic linear sum optimization problem $g(\lambda_0) := \min_{\boldsymbol{x}\in\mathcal{X}} \sum_{i=1}^{N} h_i(\lambda_0)x_i$ where $h_i(\lambda_0)$ is defined as in (3.20). Then select the smallest value from $\{g(\lambda_0) + h_0(\lambda_0) \mid \lambda_0 \in \Lambda_0\}$, where $h_0(\lambda_0)$ is defined as in (3.21). Therefore, we have the following theorem:

Theorem 3.6. *If the deterministic linear sum optimization problem* $\min_{\boldsymbol{x}\in\mathcal{X}} \boldsymbol{c}^T\boldsymbol{x}$ *is polynomially solvable, then under the discrete marginal distribution model, the distributionally robust k-sum optimization problem*

$$\min_{\boldsymbol{x}\in\mathcal{X}} \sup_{P\in\mathbb{P}(P_1,\ldots,P_N)} \mathbb{E}_P \left[\max\left\{ \sum_{i=1}^{N} \tilde{c}_i x_i y_i \mid \sum_{i=1}^{N} y_i = k, y_i \in \{0,1\}, \forall i \in [N] \right\} \right] \quad (3.23)$$

is also polynomially solvable.

Remark 3.1. *In the marginal moment model, if (a) range and mean are given, or (b) range mean and mean absolute deviation are given, the worst case marginal distributions are two points or three points discrete distributions, respectively. Hence, the result of Theorem 3.6 also holds for the distributionally robust k-sum optimization problem in these two marginal moment models.*

Remark 3.2. *The deterministic k-sum optimization problem (3.15) can be viewed as a special case of the distributionally robust k-sum optimization problem with univariate support discrete marginal distributions. Hence if the linear combinatorial optimization problem* $\min_{\boldsymbol{x}\in\mathcal{X}} \boldsymbol{c}^T\boldsymbol{x}$ *is polynomially solvable, the k-sum optimization problem (3.15) is also polynomially solvable.*

Chapter 4

A Preprocessing Method for Random Quadratic Unconstrained Binary Optimization

The Quadratic Convex Reformulation (QCR) method proposed by Billionnet and Elloumi (2007) can be used to solve quadratic unconstrained binary optimization problems using a preprocessing technique. In this method, the semidefinite relaxation is used to reformulate it to a convex binary quadratic program which is solved using mixed integer quadratic programming solvers. We extend this method to random quadratic unconstrained binary optimization problems, and develop a Penalized QCR method where the objective function in the semidefinite program is penalized with a separable term to account for the randomness in the objective.

Structure of the chapter:

1. In Section 4.1, we review the quadratic convex reformulation method to solve the deterministic quadratic unconstrained binary optimization (QUBO) problems. Then we propose the central problem of the chapter which is to solve a set of QUBO problems where uncertainty lies in the linear term of the objective function.

2. In Section 4.2, we develop an equivalent but computationally implementable reformulation to find the tight upper bound β^* that exploits the structure of the Fréchet class of distributions. This reformulation is used in developing the Penalized QCR method.

3. In Section 4.3, we develop a SDP relaxation using the continuous relaxation of the reformulation to find a weaker upper bound on the optimal expected value. The SDP relaxation has a natural interpretation as a Penalized Quadratic Convex Reformulation for QUBO problems with a random objective for the Fréchet class of distributions. Using this semidefinite program, we identify an "optimal" preprocessing vector $\boldsymbol{u}$ for this class of random QUBO problems.

4. In Section 4.4, we provide an extensive comparison between different approaches to solve QUBO problems with random objective coefficients. We demonstrate that for problems with up to 100 variables, the Penalized QCR method developed in this paper has computational advantages over alternate preprocessing approaches in terms of computational times and the quality of the bounds.

4.1 Introduction

Consider the quadratic function:

$$q(\boldsymbol{x}; \boldsymbol{c}, \boldsymbol{Q}) = \boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} + \boldsymbol{c}^T \boldsymbol{x}$$

and the corresponding quadratic unconstrained binary optimization:

$$\text{(QUBO)} \quad \beta(\boldsymbol{c}, \boldsymbol{Q}) = \max_{\boldsymbol{x} \in \{0,1\}^N} q(\boldsymbol{x}; \boldsymbol{c}, \boldsymbol{Q}), \tag{4.1}$$

where $\boldsymbol{Q}$ is a $N \times N$ real symmetric matrix (not necessarily negative semidefinite), and $\boldsymbol{c} \in \Re^N$.

For a binary variable $x_i \in \{0,1\}$, we have $x_i^2 = x_i$ and hence $\boldsymbol{x}^T \text{diag}(\boldsymbol{u})\boldsymbol{x} = \boldsymbol{u}^T\boldsymbol{x}$ for any $\boldsymbol{u} \in \Re^N$, where $\text{diag}(\boldsymbol{u})$ is the diagonal matrix obtained from the vector $\boldsymbol{u}$. A simple perturbation idea is to choose a vector $\boldsymbol{u} \in \Re^N$, such that $\boldsymbol{Q} - \text{diag}(\boldsymbol{u})$ is negative semidefinite. Define:

$$q_{\boldsymbol{u}}(\boldsymbol{x}; \boldsymbol{c}, \boldsymbol{Q}) = \boldsymbol{x}^T \left(\boldsymbol{Q} - \text{diag}(\boldsymbol{u})\right) \boldsymbol{x} + (\boldsymbol{c} + \boldsymbol{u})^T \boldsymbol{x}, \tag{4.2}$$

and the associated quadratic unconstrained binary maximization problem with a concave quadratic objective:

$$\beta(\boldsymbol{u}; \boldsymbol{c}, \boldsymbol{Q}) \quad = \quad \max_{\boldsymbol{x} \in \{0,1\}^N} q_{\boldsymbol{u}}(\boldsymbol{x}; \boldsymbol{c}, \boldsymbol{Q}). \tag{4.3}$$

Then, $\beta(\boldsymbol{c}, \boldsymbol{Q}) = \beta(\boldsymbol{u}; \boldsymbol{c}, \boldsymbol{Q})$. Since the objective function in (4.3) is concave, it is possible to use off-the-shelf mixed integer quadratic programming solvers such as CPLEX to solve it. This gives an exact solution method to solve the QUBO problem where in the preprocessing step, a perturbation vector $\boldsymbol{u}$ is chosen such that $\boldsymbol{Q} - \text{diag}(\boldsymbol{u})$ is negative semidefinite and then in the solution step, the convex binary quadratic programming problem is solved.

The simplest possible choice of the perturbation vector $\boldsymbol{u}$ is to use:

$$\boldsymbol{u}_{\text{eig}} = \lambda_{\max}(\boldsymbol{Q})\boldsymbol{e}, \tag{4.4}$$

where $\lambda_{\max}(\boldsymbol{Q})$ is the largest eigenvalue of the matrix $\boldsymbol{Q}$ and $\boldsymbol{e}$ is an N dimensional vector with all entries equal to 1. Clearly, $\boldsymbol{Q} - \text{diag}(\boldsymbol{u}_{\text{eig}})$ is negative semidefinite and the function $q_{\boldsymbol{u}_{\text{eig}}}(\boldsymbol{x}; \boldsymbol{c}, \boldsymbol{Q})$ is concave with respect to the decision variables. Such an eigenvalue based preprocessing method was first proposed by Hammer and Rubin [65].

The work most closely related to the method developed in this chapter is the Quadratic Convex Reformulation (QCR) method proposed by Billionnet and Elloumi [25] that we discuss in detail next.

4.1.1 Quadratic Convex Reformulation

One approach to solve the QUBO problem is QCR method proposed by Billionnet and Elloumi [25]. Their method is inspired from the semidefinite programming relaxations for discrete optimization problems developed in Körner [82], Shor [111] and Poljak, Rendl and Wolkowicz [98] among others. Billionnet and Elloumi [25] developed a preprocessing phase where the "optimal" choice of the parameter vector $\boldsymbol{u}$ was found by solving a semidefinite program (SDP). Their approach is based on evaluating the upper bound $\bar{\beta}(\boldsymbol{u}; \boldsymbol{c}, \boldsymbol{Q})$ of the optimal value of the QUBO problem obtained by solving the convex relaxation of problem (4.3):

$$\bar{\beta}(\boldsymbol{u}; \boldsymbol{c}, \boldsymbol{Q}) = \max_{\boldsymbol{x} \in [0,1]^N} q_{\boldsymbol{u}}(\boldsymbol{x}; \boldsymbol{c}, \boldsymbol{Q}). \tag{4.5}$$

The "optimal" vector $\boldsymbol{u}_{\text{opt}}$ is chosen such that it minimizes the upper bound $\bar{\beta}(\boldsymbol{u}; \boldsymbol{c}, \boldsymbol{Q})$ subject to the constraint $\text{diag}(\boldsymbol{u}) - \boldsymbol{Q} \succeq 0$. Let

$$\boldsymbol{u}_{\text{opt}} = \text{argmin} \left\{ \bar{\beta}(\boldsymbol{u}; \boldsymbol{c}, \boldsymbol{Q}) \mid \text{diag}(\boldsymbol{u}) - \boldsymbol{Q} \succeq 0 \right\}. \tag{4.6}$$

Using standard duality arguments, it was shown in [25] that $\boldsymbol{u}_{\text{opt}}$ is the optimal $\boldsymbol{u}$ vector in the following SDP:

$$\begin{aligned} \bar{\beta}(\boldsymbol{u}_{\text{opt}}; \boldsymbol{c}, \boldsymbol{Q}) = \quad & \min_{r, \boldsymbol{u}} \quad r \\ & \text{s.t.} \quad \begin{bmatrix} r & -(\boldsymbol{c} + \boldsymbol{u})^T/2 \\ -(\boldsymbol{c} + \boldsymbol{u})/2 & \text{diag}(\boldsymbol{u}) - \boldsymbol{Q} \end{bmatrix} \succeq 0, \end{aligned} \tag{4.7}$$

where the positive semidefiniteness constraint is for a matrix of size $(N+1) \times (N+1)$. The semidefinite program (4.7) is the dual to the classic semidefinite programming relaxation of the QUBO problem:

$$\begin{aligned} \bar{\beta}(\boldsymbol{u}_{\text{opt}}; \boldsymbol{c}, \boldsymbol{Q}) = \quad & \max_{\boldsymbol{x}, \boldsymbol{X}} \quad \sum_{i=1}^{N} \sum_{j=1}^{N} Q_{ij} X_{ij} + \sum_{i=1}^{N} c_i x_i \\ & \text{s.t.} \quad \begin{bmatrix} 1 & \boldsymbol{x}^T \\ \boldsymbol{x} & \boldsymbol{X} \end{bmatrix} \succeq 0 \\ & \qquad X_{ii} = x_i, \quad i \in [N], \end{aligned} \tag{4.8}$$

where $\boldsymbol{u}_{\text{opt}}$ is the optimal dual variables to the equality constraints $X_{ii} = x_i,\ i \in [N]$. The SDP relaxation in (4.8) has been considered by several authors including Körner [82], Shor [111] and Poljak, Rendl and Wolkowicz [98] among others. Billionnet and Elloumi [25] proposed the use of this semidefinite program as a preprocessing phase to find the optimal perturbation vector before applying an exact branch and bound method to solve the QUBO problem based on solving convex relaxations. In the numerical experiments, they showed that the relative gap between the optimum value of the QUBO problem and the continuous relaxation $\bar{\beta}(\boldsymbol{u}_{\text{opt}}; \boldsymbol{c}, \boldsymbol{Q})$ is about half the relative gap between the optimum value and $\bar{\beta}(\boldsymbol{u}_{\text{eig}}; \boldsymbol{c}, \boldsymbol{Q})$. Solving the QUBO problem with the CPLEX solver is also faster using the SDP based preprocessing step as compared to the eigenvalue based preprocessing step. Subsequently, Billionnet et. al. ([29]) extended the QCR method to 0-1 quadratic programming problem with linear constraints and to more general mixed-integer programs in [28, 27, 26]. Galli and Letchford [50] extended this approach to mixed-integer quadratically constrained quadratic programs.

4.1.2 The Main Problem

In this chapter, we extend the QCR method to solve parametric quadratic unconstrained binary optimization problems:

$$\max_{\boldsymbol{x}\in\{0,1\}^N} \left\{ q(\boldsymbol{x}; \boldsymbol{c}, \boldsymbol{Q}) := \boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} + \boldsymbol{c}^T \boldsymbol{x} \right\}, \quad \forall \boldsymbol{c} \in \mathcal{C}, \tag{4.9}$$

where $\boldsymbol{Q}$ is a fixed $N \times N$ symmetric real matrix, and the parameter vector $\boldsymbol{c}$ varies in a set $\mathcal{C}$. Our main purpose is to find a common preprocessing vector $\boldsymbol{u}$ such that $\text{diag}(\boldsymbol{u}) - \boldsymbol{Q} \succeq 0$, and the preprocessing vector is "optimal" in some sense. Moreover, using this common preprocessing vector, we hope that we can efficiently compute the solutions of the QUBO problems. To find such a preprocessing vector $\boldsymbol{u}$ for all $\boldsymbol{c} \in \mathcal{C}$, we assume that the parameter vector $\tilde{\boldsymbol{c}}$ is random with a probability distribution denoted by P. The expected optimal objective value for the QUBO

problem (4.9) averaged over the possible realizations of $\tilde{\boldsymbol{c}}$ is expressed as:

$$\mathbb{E}_P\left[\beta(\tilde{\boldsymbol{c}}, \boldsymbol{Q})\right] = \int_{\mathcal{C}} \max_{\boldsymbol{x}\in\{0,1\}^N} q(\boldsymbol{x}; \boldsymbol{c}, \boldsymbol{Q}) dP(\boldsymbol{c}). \tag{4.10}$$

Evaluating $\mathbb{E}_P\left[\beta(\tilde{\boldsymbol{c}}, \boldsymbol{Q})\right]$ is clearly challenging since we need to solve a set of NP-hard QUBO problems for each realization of $\boldsymbol{c}$.

To facilitate the analysis, we assume that the probability distribution P for the random vector $\tilde{\boldsymbol{c}}$ is not completely specified. Rather, the joint distribution P lies in a Fréchet class of multivariate joint distributions that consists of all multivariate joint distributions with fixed marginal distributions P_i for each component $\tilde{c}_i$; for more details, see for example [40]. We denote the Fréchet class of distributions as $\mathbb{P}(P_1, \ldots, P_N)$. Distributions in the Fréchet class differ with respect to the dependency structures between the fixed marginal distributions. Since the probability distribution is incompletely specified, we focus on the extremal multivariate joint distribution of the random parameter vector $\tilde{\boldsymbol{c}}$ that maximizes the expected optimal objective value of the quadratic unconstrained binary optimization problem over all distributions in the Fréchet class. The problem of interest is defined as:

$$\begin{aligned} \beta^* &= \sup_{P\in\mathbb{P}(P_1,\ldots,P_N)} \mathbb{E}_P\left[\beta(\tilde{\boldsymbol{c}}, \boldsymbol{Q})\right] \\ &= \sup_{P\in\mathbb{P}(P_1,\ldots,P_N)} \int_{\mathcal{C}} \max_{\boldsymbol{x}\in\{0,1\}^N} q(\boldsymbol{x}; \boldsymbol{c}, \boldsymbol{Q}) dP(\boldsymbol{c}). \end{aligned} \tag{4.11}$$

4.2 A Tight Upper Bound on the Expected Optimal Value

In this section, we develop a reformulation for (4.11) to evaluate the tight upper bound on the expected optimal value of QUBO problem. Our approach is based on the results in Meilijson and Nadas [88] who developed a convex majorization approach to compute the tightest upper bound on the expected length of a critical path in a project network for the Fréchet class of distributions. Weiss [120] generalized this bound to linear combinatorial optimization problems such as the shortest path, maximum flow, and the reliability problem. The main result in these papers

is outlined as

$$\sup_{P\in\mathbb{P}(P_1,\ldots,P_N)} \mathbb{E}_P\left[\max_{\boldsymbol{x}\in\mathcal{X}\subseteq\{0,1\}^N} \tilde{\boldsymbol{c}}^T\boldsymbol{x}\right] = \min_{\boldsymbol{d}\in\Re^N}\left(\max_{\boldsymbol{x}\in\mathcal{X}\subseteq\{0,1\}^N} \boldsymbol{d}^T\boldsymbol{x} + \sum_{i=1}^N \mathbb{E}_{P_i}[\tilde{c}_i - d_i]^+\right). \tag{4.12}$$

Extensions of this approach to limited information on the marginal distributions have been proposed in Klein Haneveld [66], Birge and Maddox [30], Bertsimas, Natarajan and Teo [24, 20] and Natarajan, Song and Teo [91] among others. Using a similar approach, we develop a reformulation for the expected optimal objective value of QUBO problems for the Fréchet class of distributions in the next proposition.

Proposition 4.1. *For each $i \in [N]$, assume that the marginal distribution P_i of the continuously distributed random variable $\tilde{c}_i$ is given. Define*

$$\beta^* = \sup_{P\in\mathbb{P}(P_1,\ldots,P_N)} \mathbb{E}_P\left[\max_{\boldsymbol{x}\in\{0,1\}^N} \left(\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \tilde{\boldsymbol{c}}^T\boldsymbol{x}\right)\right], \tag{4.13}$$

and

$$\beta^{**} = \min_{\boldsymbol{d}\in\Re^N}\left(\max_{\boldsymbol{x}\in\{0,1\}^N} \left(\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{d}^T\boldsymbol{x}\right) + \sum_{i=1}^N \mathbb{E}_{P_i}[\tilde{c}_i - d_i]^+\right). \tag{4.14}$$

Then the optimal objective values of the two formulations are equal, $\beta^* = \beta^{**}$.

Proof. For any $\boldsymbol{c}$ and $\boldsymbol{d} \in \Re^N$, the following holds:

$$\begin{aligned}\max_{\boldsymbol{x}\in\{0,1\}^N} \left(\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c}^T\boldsymbol{x}\right) &= \max_{\boldsymbol{x}\in\{0,1\}^N} \left(\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{d}^T\boldsymbol{x} + (\boldsymbol{c}-\boldsymbol{d})^T\boldsymbol{x}\right)\\ &\leq \max_{\boldsymbol{x}\in\{0,1\}^N} \left(\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{d}^T\boldsymbol{x}\right) + \sum_{i=1}^N [c_i - d_i]^+.\end{aligned}$$

Taking expectation with respect to the probability measure $P \in \mathbb{P}(P_1, \ldots, P_N)$ and the minimum with respect to $\boldsymbol{d} \in \Re^N$, we obtain

$$\mathbb{E}_P\left[\max_{\boldsymbol{x}\in\{0,1\}^N} \left(\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c}^T\boldsymbol{x}\right)\right] \leq \beta^{**}, \quad \forall P \in \mathbb{P}(P_1, \ldots, P_N).$$

Taking supremum with respect to $P \in \mathbb{P}(P_1, \ldots, P_N)$, implies $\beta^* \leq \beta^{**}$.

Next we show $\beta^{**} \leq \beta^*$. Notice that β^{**} can be evaluated as the optimal objective to the following convex programming problem with decision variables $\boldsymbol{d}$

and t:

$$\begin{aligned} \beta^{**} &= \min_{\boldsymbol{d},t} \quad t + \sum_{i=1}^{N} \mathbb{E}_{P_i}[\tilde{c}_i - d_i]^+ \\ &\quad \text{s.t.} \quad t \geq \boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} + \boldsymbol{d}^T \boldsymbol{x}, \quad \forall \boldsymbol{x} \in \{0,1\}^N. \end{aligned} \tag{4.15}$$

The Karush-Kuhn-Tucker (KKT) conditions for (4.15) are:

$$\lambda(\boldsymbol{x}) \geq 0, t \geq \boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} + \boldsymbol{d}^T \boldsymbol{x}, \quad \forall \boldsymbol{x} \in \{0,1\}^N, \tag{4.16a}$$

$$\sum_{\boldsymbol{x} \in \{0,1\}^N} \lambda(\boldsymbol{x}) = 1, \tag{4.16b}$$

$$\lambda(\boldsymbol{x})(t - \boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} - \boldsymbol{d}^T \boldsymbol{x}) = 0, \quad \forall \boldsymbol{x} \in \{0,1\}^N, \tag{4.16c}$$

$$P(\tilde{c}_i \geq d_i) = \sum_{\boldsymbol{x} \in \{0,1\}^N : x_i = 1} \lambda(\boldsymbol{x}). \tag{4.16d}$$

The Slater's condition for (4.15) is satisfied. Hence there exist dual variables $\lambda(\boldsymbol{x})$ and primal variables $\boldsymbol{d}$, t satisfying the KKT conditions. In the rest of the proof, we let $\boldsymbol{d}, t, \lambda(\boldsymbol{x})$ denote the solutions to the KKT conditions (4.16a)-(4.16d). Let $f_i(\cdot)$ be the probability density function associated with P_i. Next we construct a distribution $\bar{P}$ as follows.

(a) Generate a random vector $\tilde{\boldsymbol{x}}$ which takes the value $\boldsymbol{x} \in \{0,1\}^N$ with probability $\lambda(\boldsymbol{x})$.

(b) Define the set $I_1 = \{i \in [N] : 0 < P(\tilde{c}_i \geq d_i) < 1\}$ and $I_2 = [N] \setminus I_1$. For $i \in I_1$, generate the random variable $\tilde{c}_i$ with the conditional probability density function

$$\bar{f}_i(c_i \mid \tilde{\boldsymbol{x}} = \boldsymbol{x}) = \begin{cases} \dfrac{1}{P(\tilde{c}_i \geq d_i)} \mathbb{I}_{[d_i, \infty)}(c_i) f_i(c_i), & \text{if } x_i = 1, \\ \dfrac{1}{P(\tilde{c}_i < d_i)} \mathbb{I}_{(-\infty, d_i)}(c_i) f_i(c_i), & \text{if } x_i = 0. \end{cases}$$

For $i \in I_2$, generate the random variable $\tilde{c}_i$ with the conditional probability density function $\bar{f}_i(c_i \mid \tilde{\boldsymbol{x}} = \boldsymbol{x}) = f_i(c_i)$.

For $i \in I_1$, the marginal probability density function under $\bar{P}$ is

$$\bar{f}_i(c_i) = \sum_{\boldsymbol{x} \in \{0,1\}^N} \lambda(\boldsymbol{x}) \bar{f}_i(c_i \mid \tilde{\boldsymbol{x}} = \boldsymbol{x})$$

$$
\begin{aligned}
&= \sum_{\boldsymbol{x}\in\{0,1\}^N:x_i=1} \lambda(\boldsymbol{x}) \frac{1}{P(\tilde{c}_i \geq d_i)} \mathbb{I}_{[d_i,\infty)}(c_i) f_i(c_i) \\
&\quad + \sum_{\boldsymbol{x}\in\{0,1\}^N:x_i=0} \lambda(\boldsymbol{x}) \frac{1}{P(\tilde{c}_i < d_i)} \mathbb{I}_{(-\infty,d_i)}(c_i) f_i(c_i) \\
&= \mathbb{I}_{[d_i,\infty)}(c_i) f_i(c_i) + \mathbb{I}_{[-\infty,d_i)}(c_i) f_i(c_i) \quad \text{(by (4.16}d\text{))} \\
&= f_i(c_i).
\end{aligned}
$$

For $i \in I_2$, it is easy to see that $\bar{f}_i(c_i) = f_i(c_i)$. Hence, the constructed probability distribution $\bar{P} \in \mathbb{P}(P_1, \ldots, P_N)$. Therefore

$$
\begin{aligned}
\beta^* &\geq \mathbb{E}_{\bar{P}} \left[\max_{\boldsymbol{y}\in\{0,1\}^N} \boldsymbol{y}^T \boldsymbol{Q} \boldsymbol{y} + \tilde{\boldsymbol{c}}^T \boldsymbol{y} \right] \\
&\geq \sum_{\boldsymbol{x}\in\{0,1\}^N} \lambda(\boldsymbol{x}) \mathbb{E}_{\bar{P}} \left[\max_{\boldsymbol{y}\in\{0,1\}^N} \boldsymbol{y}^T \boldsymbol{Q} \boldsymbol{y} + \tilde{\boldsymbol{c}}^T \boldsymbol{y} \mid \tilde{\boldsymbol{x}} = \boldsymbol{x} \right] \\
&\geq \sum_{\boldsymbol{x}\in\{0,1\}^N} \lambda(\boldsymbol{x}) \mathbb{E}_{\bar{P}} \left[\boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} + \tilde{\boldsymbol{c}}^T \boldsymbol{x} \mid \tilde{\boldsymbol{x}} = \boldsymbol{x} \right] \\
&= \sum_{\boldsymbol{x}\in\{0,1\}^N} \lambda(\boldsymbol{x}) \boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} \\
&\quad + \sum_{\boldsymbol{x}\in\{0,1\}^N:x_i=1} \lambda(\boldsymbol{x}) \left(\sum_{i\in I_1} \int c_i \frac{1}{P(\tilde{c}_i \geq d_i)} \mathbb{I}_{[d_i,\infty)}(c_i) f_i(c_i) dc_i + \sum_{i\in I_2} \int c_i f_i(c_i) dc_i \right) \\
&= \sum_{\boldsymbol{x}\in\{0,1\}^N} \lambda(\boldsymbol{x}) \boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} + \sum_{i\in I_1} \int c_i \mathbb{I}_{[d_i,\infty)}(c_i) f_i(c_i) dc_i + \sum_{i\in I_2} \int c_i P(\tilde{c}_i \geq d_i) f_i(c_i) dc_i.
\end{aligned}
$$

Since $P(\tilde{c}_i \geq d_i) = 1$ or 0 for $i \in I_2$, hence

$$
\int c_i P(\tilde{c}_i \geq d_i) f_i(c_i) dc_i = \int c_i \mathbb{I}_{[d_i,\infty)}(c_i) f_i(c_i) dc_i, \forall i \in I_2.
$$

As a result

$$
\begin{aligned}
\beta^* &\geq \sum_{\boldsymbol{x}\in\{0,1\}^N} \lambda(\boldsymbol{x}) \boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} + \sum_{i=1}^N \int c_i \mathbb{I}_{[d_i,\infty)}(c_i) f_i(c_i) dc_i \\
&= \sum_{\boldsymbol{x}\in\{0,1\}^N} \lambda(\boldsymbol{x}) \boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} + \sum_{i=1}^N d_i \int \mathbb{I}_{[d_i,\infty)}(c_i) f_i(c_i) dc_i \\
&\quad + \sum_{i=1}^N \int (c_i - d_i) \mathbb{I}_{[d_i,\infty)}(c_i) f_i(c_i) dc_i \\
&= \sum_{\boldsymbol{x}\in\{0,1\}^N} \lambda(\boldsymbol{x}) \boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} + \sum_{i=1}^N d_i \sum_{\boldsymbol{x}\in\{0,1\}^N} \lambda(\boldsymbol{x}) x_i + \sum_{i=1}^N \mathbb{E}_{P_i}[\tilde{c}_i - d_i]^+ \quad \text{(by (4.16}d\text{))}
\end{aligned}
$$

$$
\begin{aligned}
&= \sum_{\boldsymbol{x}\in\{0,1\}^N} \lambda(\boldsymbol{x})(\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{d}^T\boldsymbol{x}) + \sum_{i=1}^{N} \mathbb{E}_i[\tilde{c}_i - d_i]^+ \\
&= \sum_{\boldsymbol{x}\in\{0,1\}^N} \lambda(\boldsymbol{x})t + \sum_{i=1}^{N} \mathbb{E}_i[\tilde{c}_i - d_i]^+ \quad \text{(by (4.16}c\text{))} \\
&= t + \sum_{i=1}^{N} \mathbb{E}_i[\tilde{c}_i - d_i]^+ \quad \text{(by (4.16}b\text{))} \\
&= \beta^{**}.
\end{aligned}
$$

□

Formulation (4.14) exploits the marginal specification of the joint distribution to provide a convex formulation in the $\boldsymbol{d}$ variables. The objective function in (4.14) consists of two parts: (a) A deterministic QUBO problem with an objective of maximizing $\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{d}^T\boldsymbol{x}$ for a fixed $\boldsymbol{d}$ and (b) a sum of N univariate convex penalty terms, each of the form $\mathbb{E}_{P_i}[\tilde{c}_i - d_i]^+$. The reformulation in (4.14) is NP-hard to solve since computing the first term in the objective for a fixed vector $\boldsymbol{d}$ is equivalent to solving a QUBO problem. A simple interpretation of this formulation is to find the balance between a deterministic approximation of the random QUBO problem based on the chosen $\boldsymbol{d}$ and a penalty term for choosing the vector $\boldsymbol{d}$ differently from the random vector $\tilde{\boldsymbol{c}}$. This result extends to discrete marginal distributions where in the proof, we need to replace the integrals with summations and use linear programming duality. It is also possible to extend the result of Proposition 4.1 to the case where only the mean and variance of each random variable is known. The result is stated in the next proposition.

Proposition 4.2. *Assume that the mean and variance for each $\tilde{c}_i$ are given , i.e.*

$$\mathbb{P}_i = \{P_i : \mathbb{E}_{P_i}(\tilde{c}_i) = \mu_i, \quad \mathbb{E}_{P_i}(\tilde{c}_i^2) = \mu_i^2 + \sigma_i^2\}, \ i \in [N].$$

Define

$$\beta^* := \sup_{P\in\mathbb{P}(\mathbb{P}_1,\ldots,\mathbb{P}_N)} \mathbb{E}_P\left[\max_{\boldsymbol{x}\in\{0,1\}^N} \left(\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \tilde{\boldsymbol{c}}^T\boldsymbol{x}\right)\right], \tag{4.17}$$

and

$$\beta^{**} = \min_{\boldsymbol{d}\in\Re^N} \left\{ \max_{\boldsymbol{x}\in\{0,1\}^N} (\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{d}^T\boldsymbol{x}) + \sum_{i=1}^{N} \sup_{P_i\in\mathbb{P}_i} \mathbb{E}_{P_i}[\tilde{c}_i - d_i]^+ \right\}. \tag{4.18}$$

Then the optimal objective values of the formulations are equal, $\beta^* = \beta^{**}$.

Proof. First from Proposition 4.1, we know that

$$\begin{aligned}\beta^* &= \sup_{P_i \in \mathbb{P}_i, i \in [N]} \sup_{P \in \mathbb{P}(P_1,\ldots,P_N)} \mathbb{E}_P \left[\max_{\boldsymbol{x} \in \{0,1\}^N} \boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} + \tilde{\boldsymbol{c}}^T \boldsymbol{x} \right] \\ &= \sup_{P_i \in \mathbb{P}_i, i \in [N]} \min_{\boldsymbol{d} \in \Re^N} \left\{ \max_{\boldsymbol{x} \in \{0,1\}^N} (\boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} + \boldsymbol{d}^T \boldsymbol{x}) + \sum_{i=1}^N \mathbb{E}_{P_i}[\tilde{c}_i - d_i]^+ \right\}.\end{aligned}$$

Notice that in the above formula, the objective function is convex with respect to the variable $\boldsymbol{d} \in \Re^N$, and linear with respect to the distribution $P_i \in \mathbb{P}_i, \forall i \in [N]$. Moreover, every probability density function in the distribution set $\mathbb{P}_i$ is bounded in the L_1 space. Hence by Theorem 6 and its corollary in Rockafellar [105], we can exchange the position of sup and min in the above formula. That is

$$\begin{aligned}\beta^* &= \min_{\boldsymbol{d} \in \Re^N} \sup_{P_i \in \mathbb{P}_i, i \in [N]} \left\{ \max_{\boldsymbol{x} \in \{0,1\}^N} (\boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} + \boldsymbol{d}^T \boldsymbol{x}) + \sum_{i=1}^N \mathbb{E}_{P_i}[\tilde{c}_i - d_i]^+ \right\} \\ &= \min_{\boldsymbol{d} \in \Re^N} \left\{ \max_{\boldsymbol{x} \in \{0,1\}^N} (\boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} + \boldsymbol{d}^T \boldsymbol{x}) + \sum_{i=1}^N \sup_{P_i \in \mathbb{P}_i} \mathbb{E}_{P_i}[\tilde{c}_i - d_i]^+ \right\} \\ &= \beta^{**}.\end{aligned}$$

□

4.3 The "Optimal" Preprocessing Vector

Our goal in this section is to find a preprocessing vector $\boldsymbol{u}$ such that the matrix $\boldsymbol{Q} - \mathrm{diag}(\boldsymbol{u})$ is negative semidefinite and it is an "optimal" choice for the extremal probability distribution of the random parameter vector $\tilde{\boldsymbol{c}}$ that attains the upper bound. Note that the "optimal" choice for $\boldsymbol{u}$ has to carefully defined for random QUBO problems since we are solving multiple instances of deterministic QUBO problems drawn from the extremal distribution.

Let $\mathbb{P}_i$ denote the set of possible marginal distributions for the random variable $\tilde{c}_i$. Assume that either the marginal distribution P_i of the random variable $\tilde{c}_i$ is given in which case the set $\mathbb{P}_i = \{P_i\}$ consists of a singleton or the mean and

variance of $\tilde{c}_i$ is given in which case $\mathbb{P}_i = \{P_i : \mathbb{E}_{P_i}(\tilde{c}_i) = \mu_i, \ \ \mathbb{E}_{P_i}(\tilde{c}_i^2) = \mu_i^2 + \sigma_i^2\}$. Perturbing the objective function for the inner deterministic QUBO problem in the reformulations (4.14) or (4.18), we define:

$$\beta^*_{\boldsymbol{u}} \ = \ \min_{\boldsymbol{d}\in\Re^N} \Big\{ \max_{\boldsymbol{x}\in\{0,1\}^N} \big[\boldsymbol{x}^T(\boldsymbol{Q} - \mathrm{diag}(\boldsymbol{u}))\boldsymbol{x} + (\boldsymbol{d}+\boldsymbol{u})^T\boldsymbol{x}\big] + \sum_{i=1}^N \sup_{P_i\in\mathbb{P}_i} \mathbb{E}_{P_i}[\tilde{c}_i - d_i]^+ \Big\}, \tag{4.19}$$

From Propositions 4.1 and 4.2 and the observation that $\boldsymbol{x}^T\mathrm{diag}(\boldsymbol{u})\boldsymbol{x} = \boldsymbol{u}^T\boldsymbol{x}$ for $\boldsymbol{x} \in \{0,1\}^N$, the tight upper bound β^* is exactly equal to $\beta^*_{\boldsymbol{u}}$, namely:

$$\beta^* \ = \ \beta^*_{\boldsymbol{u}} \quad \forall \boldsymbol{u} \in \Re^N, \tag{4.20}$$

Define an upper bound $\bar{\beta}^*_{\boldsymbol{u}}$ on the optimal value β^* by using the continuous relaxation for the binary variables in the deterministic QUBO problem in (4.19):

$$\bar{\beta}^*_{\boldsymbol{u}} = \min_{\boldsymbol{d}\in\Re^N} \Big\{ \max_{\boldsymbol{x}\in[0,1]^N} \big[\boldsymbol{x}^T(\boldsymbol{Q} - \mathrm{diag}(\boldsymbol{u}))\boldsymbol{x} + (\boldsymbol{d}+\boldsymbol{u})^T\boldsymbol{x}\big] + \sum_{i=1}^N \sup_{P_i\in\mathbb{P}_i} \mathbb{E}_{P_i}[\tilde{c}_i - d_i]^+ \Big\}. \tag{4.21}$$

Then, clearly:

$$\beta^* \ \leq \ \bar{\beta}^*_{\boldsymbol{u}} \quad \forall \boldsymbol{u} \in \Re^N. \tag{4.22}$$

For a fixed perturbation vector $\boldsymbol{u}$ such that the matrix $\boldsymbol{Q} - \mathrm{diag}(\boldsymbol{u})$ is negative semidefinite, the objective function in $\bar{\beta}^*_{\boldsymbol{u}}$ is efficiently computable. This brings us to the definition of an "optimal" preprocessing vector for random QUBO problems.

Definition 4.3. *The "optimal" choice of the preprocessing vector for the random QUBO problem is defined as the vector* $\boldsymbol{u}^*_{\mathrm{opt}}$ *such that* $\boldsymbol{Q} - \mathrm{diag}(\boldsymbol{u}^*_{\mathrm{opt}})$ *is negative semidefinite and it minimizes the upper bound* $\bar{\beta}^*_{\boldsymbol{u}}$ *in (4.21) obtained from the continuous relaxation.*

In other words, $\boldsymbol{u}^*_{\mathrm{opt}}$ is chosen to minimize the efficiently computable upper bound on the expectation for the random 0-1 quadratic programming problem obtained from the continuous relaxation. It is the optimal solution to the following minimization problem:

$$\min_{\mathrm{diag}(\boldsymbol{u})-\boldsymbol{Q}\succeq 0} \ \min_{\boldsymbol{d}\in\Re^N} \Big\{ \max_{\boldsymbol{x}\in[0,1]^N} \big[\boldsymbol{x}^T(\boldsymbol{Q} - \mathrm{diag}(\boldsymbol{u}))\boldsymbol{x} + (\boldsymbol{d}+\boldsymbol{u})^T\boldsymbol{x}\big] + \sum_{i=1}^N \sup_{P_i\in\mathbb{P}_i} \mathbb{E}_{P_i}[\tilde{c}_i - d_i]^+ \Big\}. \tag{4.23}$$

Define the smallest upper bound obtained from the continuous relaxation as:

$$\bar{\beta}^* = \bar{\beta}^*_{\boldsymbol{u}^*_{\text{opt}}}.$$

Then $\beta^* \leq \bar{\beta}^* \leq \bar{\beta}^*_{\boldsymbol{u}}$ for any $\boldsymbol{u}$. Changing the order of the minimization in the outer problems in (4.23), we get

$$\bar{\beta}^* = \min_{\boldsymbol{d}\in\Re^N} \Big\{ \min_{\text{diag}(\boldsymbol{u})-\boldsymbol{Q}\succeq 0} \max_{\boldsymbol{x}\in[0,1]^N} \big[\boldsymbol{x}^T(\boldsymbol{Q}-\text{diag}(\boldsymbol{u}))\boldsymbol{x} + (\boldsymbol{d}+\boldsymbol{u})^T\boldsymbol{x}\big] + \sum_{i=1}^{N} \sup_{P_i\in\mathbb{P}_i} \mathbb{E}_{P_i}[\tilde{c}_i - d_i]^+ \Big\}. \tag{4.24}$$

For a fixed vector $\boldsymbol{d}$, the inner subproblem:

$$\min_{\text{diag}(\boldsymbol{u})-\boldsymbol{Q}\succeq 0} \max_{\boldsymbol{x}\in[0,1]^N} [\boldsymbol{x}^T(\boldsymbol{Q}-\text{diag}(\boldsymbol{u}))\boldsymbol{x} + (\boldsymbol{d}+\boldsymbol{u})^T\boldsymbol{x}],$$

is solvable as a SDP using the same approach as for the deterministic QUBO (4.7). This brings us to the main result of the paper.

Proposition 4.4. *The upper bound $\bar{\beta}^*$ on the expected optimal objective value of a QUBO problem obtained from its convex relaxation in (4.24) is equal to the optimal value of the following SDP:*

$$\begin{aligned} \bar{\beta}^* = \min_{\boldsymbol{d},r,\boldsymbol{u}} \quad & r + \sum_{i=1}^{N} \sup_{P_i\in\mathbb{P}_i} \mathbb{E}_{P_i}[\tilde{c}_i - d_i]^+ \\ s.t. \quad & \begin{bmatrix} r & -(\boldsymbol{d}+\boldsymbol{u})^T/2 \\ -(\boldsymbol{d}+\boldsymbol{u})/2 & \text{diag}(\boldsymbol{u}) - \boldsymbol{Q} \end{bmatrix} \succeq 0. \end{aligned} \tag{4.25}$$

Furthermore the optimal decision vector $\boldsymbol{u}$ is $\boldsymbol{u}^*_{\text{opt}}$ which satisfies $\text{diag}(\boldsymbol{u}^*_{\text{opt}})-\boldsymbol{Q} \succeq 0$ *and* $\bar{\beta}^* = \bar{\beta}^*_{\boldsymbol{u}^*_{\text{opt}}}$.

An alternate way to express formulation (4.25) is using the classical semidefinite relaxation of the deterministic QUBO problem as follows:

$$\begin{aligned} \bar{\beta}^* = \min_{\boldsymbol{d}\in\Re^N} \max_{\boldsymbol{x},\boldsymbol{X}} \quad & \sum_{i=1}^{N}\sum_{j=1}^{N} Q_{ij}X_{ij} + \sum_{i=1}^{N} d_i x_i + \sum_{i=1}^{N} \sup_{P_i\in\mathbb{P}_i} \mathbb{E}_{P_i}[\tilde{c}_i - d_i]^+ \\ \text{s.t.} \quad & \begin{bmatrix} 1 & \boldsymbol{x}^T \\ \boldsymbol{x} & \boldsymbol{X} \end{bmatrix} \succeq 0 \\ & X_{ii} = x_i, \quad i \in [N], \end{aligned} \tag{4.26}$$

where $\boldsymbol{u}^*_{\text{opt}}$ is the optimal dual variables to the equality constraints $X_{ii} = x_i, i \in [N]$. The main difference between (4.26) and (4.8) is that the vector $\boldsymbol{d}$ is a decision variable with an additional penalty term that is separable across $i \in [N]$. The convex SDP formulation in (4.25) is a penalized version of the SDP in (4.7) where the penalty term is the sum of N univariate convex functions $\sup_{P_i \in \mathbb{P}_i} \mathbb{E}_{P_i}[\tilde{c}_i - d_i]^+$. Hence (4.25) can be interpreted as a Penalized QCR.

Consider a deterministic vector where $\tilde{\boldsymbol{c}} = \boldsymbol{c}$ with probability 1. We show that in this case, (4.25) reduces to (4.7). For a deterministic instance, formulation (4.25) reduces to

$$\begin{aligned} \bar{\beta}^* = \min_{\boldsymbol{d},r,\boldsymbol{u}} \quad & r + \sum_{i=1}^{N} [c_i - d_i]^+ \\ \text{s.t.} \quad & \begin{bmatrix} r & -(\boldsymbol{d}+\boldsymbol{u})^T/2 \\ -(\boldsymbol{d}+\boldsymbol{u})/2 & \text{diag}(\boldsymbol{u}) - \boldsymbol{Q} \end{bmatrix} \succeq 0. \end{aligned} \tag{4.27}$$

It is straightforward to verify that $\boldsymbol{d} = \boldsymbol{c}$ is optimal for (4.27). Notice that $\bar{\beta}^*$ is the optimal objective value to the problem:

$$\bar{\beta}^* = \min_{\text{diag}(\boldsymbol{u}) - \boldsymbol{Q} \succeq 0} \min_{\boldsymbol{d} \in \Re^N} \Big\{ \max_{\boldsymbol{x} \in [0,1]^N} [\boldsymbol{x}^T(\boldsymbol{Q} - \text{diag}(\boldsymbol{u}))\boldsymbol{x} + (\boldsymbol{d}+\boldsymbol{u})^T\boldsymbol{x}] + \sum_{i=1}^{N} [c_i - d_i]^+ \Big\}. \tag{4.28}$$

Let $\boldsymbol{d}^*$ be the optimal vector in (4.28). If there exists some index i such that $d^*_i > c_i$, by setting $d_i = c_i$ the second term $\sum_{i=1}^{N}[c_i - d_i]^+$ in (4.28) will remain unchanged, while the first term $\max_{\boldsymbol{x} \in [0,1]^N}[\boldsymbol{x}^T(\boldsymbol{Q} - \text{diag}(\boldsymbol{u}))\boldsymbol{x} + (\boldsymbol{d}+\boldsymbol{u})^T\boldsymbol{x}]$ will not increase. Similarly, if there exists some index i such that $d^*_i < c_i$, by setting $d_i = c_i$ the second term $\sum_{i=1}^{N}[c_i - d_i]^+$ in (4.28) will decrease by $c_i - d^*_i$, while the first term $\max_{\boldsymbol{x} \in [0,1]^N}[\boldsymbol{x}^T(\boldsymbol{Q} - \text{diag}(\boldsymbol{u}))\boldsymbol{x} + (\boldsymbol{d}+\boldsymbol{u})^T\boldsymbol{x}]$ will increase by at most $c_i - d^*_i$. Hence $\boldsymbol{d} = \boldsymbol{c}$ is optimal for (4.27). Thus, the SDP reduces to the deterministic formulation (4.7). That means the proposed Penalized QCR method reduces to the QCR method in Billionnet and Elloumi [25].

Proposition 4.5. *Consider a deterministic vector where $\tilde{\boldsymbol{c}} = \boldsymbol{c}$ with probability 1. Then the SDP in (4.25) is equivalent to the SDP in (4.7).*

4.4 Computational Results

In this section, we apply the Penalized QCR method to solve a set of K quadratic unconstrained binary optimization problems where the instances are generated randomly from a probability distribution:

$$\beta(\boldsymbol{c}^{(k)}, \boldsymbol{Q}) = \max_{\boldsymbol{x}\in\{0,1\}^N} \left\{ q\left(\boldsymbol{x}; \boldsymbol{c}^{(k)}\right) := \boldsymbol{x}^T \boldsymbol{Q}\boldsymbol{x} + {\boldsymbol{c}^{(k)}}^T \boldsymbol{x} \right\}, \quad k \in [K]. \tag{4.29}$$

We solve the K instances of problem (4.29) using four different preprocessing approaches:

(a) **Eigenvalue based method:** In this method, we choose a common preprocessing vector by computing the maximum eigenvalue: $\boldsymbol{u}_{\text{eig}} = \lambda_{\max}(\boldsymbol{Q})\boldsymbol{e}$.

(b) **Sample based method:** In this method, we choose an optimal preprocessing vector $\boldsymbol{u}_{\text{opt}}^{(k)}$ for each instance $\boldsymbol{c} = \boldsymbol{c}^{(k)}$ by solving the semidefinite program (4.7). Thus we solve a total of K SDP problems.

(c) **Mean based method:** In this method, we choose a common preprocessing vector $\boldsymbol{u}_{\boldsymbol{\mu}}$ for $\boldsymbol{c} = \boldsymbol{\mu}$ by solving the semidefinite program (4.7). Thus we solve a single SDP.

(d) **Mean and standard deviation based method:** In this method, we choose a common preprocessing vector by solving a single semidefinite program for the Fréchet class of distributions. In our numerical experiments, we assume that only the mean μ_i and the standard deviation σ_i for each random variable is known. In this case, the penalty term $\sup_{P_i\in\mathbb{P}_i} \mathbb{E}[\tilde{c}_i - d_i]^+$ in the SDP (4.25) has a simple closed form expression based on the Cauchy-Schwarz inequality (see Scarf [108]):

$$\sup_{P_i\in\mathbb{P}_i} \mathbb{E}_{P_i}[\tilde{c}_i - d_i]^+ = \frac{1}{2}\left[(\mu_i - d_i) + \sqrt{(\mu_i - d_i)^2 + \sigma_i^2}\right].$$

Thus the preprocessing parameter $\boldsymbol{u}_{\boldsymbol{\mu},\boldsymbol{\sigma}}$ is obtained by solving the SDP:

$$\begin{array}{ll} \min\limits_{\boldsymbol{d},r,\boldsymbol{u}} & r + \dfrac{1}{2}\displaystyle\sum_{i=1}^{N}\left[(\mu_i - d_i) + \sqrt{(\mu_i - d_i)^2 + \sigma_i^2}\right] \\ \text{s.t.} & \begin{bmatrix} r & -(\boldsymbol{d}+\boldsymbol{u})^T/2 \\ -(\boldsymbol{d}+\boldsymbol{u})/2 & \operatorname{diag}(\boldsymbol{u}) - \boldsymbol{Q} \end{bmatrix} \succeq 0. \end{array} \tag{4.30}$$

This is equivalent to the following SDP with one positive semidefinite matrix of size $(N+1)\times(N+1)$ and N second order conic programming (SOCP) constraints:

$$\begin{array}{ll} \min\limits_{\boldsymbol{d},r,\boldsymbol{u},\boldsymbol{t}} & r + \frac{1}{2}\left[\boldsymbol{e}^T(\boldsymbol{\mu}-\boldsymbol{d}) + \boldsymbol{e}^T\boldsymbol{t}\right] \\ \text{s.t.} & \begin{bmatrix} r & -(\boldsymbol{d}+\boldsymbol{u})^T/2 \\ -(\boldsymbol{d}+\boldsymbol{u})/2 & \operatorname{diag}(\boldsymbol{u}) - \boldsymbol{Q} \end{bmatrix} \succeq 0, \\ & \left\| \begin{bmatrix} \mu_i - d_i \\ \sigma_i \end{bmatrix} \right\| \leq t_i,\ i \in [N]. \end{array} \tag{4.31}$$

Define the value $\text{obj}_k(\boldsymbol{c}^{(k)}, \boldsymbol{Q})$ as follows:

$$\text{obj}_k(\boldsymbol{c}^{(k)}, \boldsymbol{Q}) = \begin{cases} \beta(\boldsymbol{c}^{(k)}, \boldsymbol{Q}), \text{if the QUBO problem is solvable within } T \text{ minutes,} \\ \text{Best lower bound found, otherwise.} \end{cases}$$

In our computational experiments, we set $T = 10$ minutes. We define $\text{gap}_k(\boldsymbol{u})$ as the relative difference between the objective function of the convex relaxation for a given preprocessing vector $\boldsymbol{u}$ and the value $\text{obj}_k(\boldsymbol{c}^{(k)}, \boldsymbol{Q})$:

$$\text{gap}_k(\boldsymbol{u}) = \frac{\bar{\beta}(\boldsymbol{u}; \boldsymbol{c}^{(k)}, \boldsymbol{Q}) - \text{obj}_k(\boldsymbol{c}^{(k)}, \boldsymbol{Q})}{\text{obj}_k(\boldsymbol{c}^{(k)}, \boldsymbol{Q})}.$$

Since the running time of the branch-and-bound method to solve the binary quadratic program depends on the strength of its convex relaxation, we say that a vector $\boldsymbol{u}$ is preferable to $\boldsymbol{u}'$ for the kth instance if $\text{gap}_k(\boldsymbol{u}) < \text{gap}_k(\boldsymbol{u}')$. The computational study was performed in Matlab R2012a on an Intel Core 2 Duo CPU (2.8 GHz) laptop with 4 GB of RAM. The SDP problems were solved with CVX ([60, 59]) and SDPT3 ([116, 117]), and the 0-1 quadratic programming problems were solved with CPLEX 12.4 using the Matlab interface.

4.4.1 Randomly Generated Instances

Given the mean $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$, we generate the scenarios $\boldsymbol{c}^{(k)},\ k \in [K]$ from a multivariate normal distribution $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. The parameters are chosen in the following manner:

1. $\boldsymbol{Q}$ is a symmetric random matrix with density $d \in (0, 1]$. The density refers to the probability that an entry of $\boldsymbol{Q}$ is nonzero. Each nonzero entry is the sum of one or more normally distributed random variables.

2. Each component of the mean vector $\boldsymbol{\mu}$ is randomly generated from the standard normal distribution.

3. Each component of the vector of standard deviations $\boldsymbol{\sigma}$ is randomly generated from the uniform distribution $U(0, M)$, where M is a given positive number. The covariance matrix $\boldsymbol{\Sigma}$ is obtained from a randomly generated correlation matrix and the standard deviation vector $\boldsymbol{\sigma}$.

For a given pair of parameters (N, d) we generate the symmetric matrix $\boldsymbol{Q}$ of size $N \times N$ with density d and $K = 100$ instances of $\boldsymbol{c}^{(k)}$ from a normal distribution $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. We compare the quality of the bounds and the CPU times to solve these instances with the four different choices of preprocessing vectors $\boldsymbol{u}$. In our computations, we allow for a maximum CPU time of 10 minutes to solve the binary quadratic program. The numerical results are shown in the Tables 4.1 and 4.2 where we set $M = 1$ and $M = 20$ respectively. In these Tables, we report the following values for the four different choices of preprocessing vectors $\boldsymbol{u} = \boldsymbol{u}_{\text{eig}}$, $\boldsymbol{u} = \boldsymbol{u}^{(k)}_{\text{opt}}$, $\boldsymbol{u} = \boldsymbol{u}_{\boldsymbol{\mu}}$ and $\boldsymbol{u} = \boldsymbol{u}_{\boldsymbol{\mu},\boldsymbol{\sigma}}$:

1. The average gap over the 100 instances given as $\text{gap} = \sum_{k=1}^{100} \text{gap}_k(\boldsymbol{u})/100$.

2. The CPU time taken to compute the preprocessing parameter $\boldsymbol{u}$ denoted by "t_$\boldsymbol{u}$". For the sample based method, "t_$\boldsymbol{u}$" is the total CPU time taken to solve the 100 SDPs. For the mean based and mean and standard deviation

based methods, "t_$\boldsymbol{u}$" is the CPU time taken to solve a single SDP. For the eigenvalue based method, "t_$\boldsymbol{u}$" is the CPU time taken to compute the largest eigenvalue.

3. The total CPU time taken to compute all the QUBO problems solvable by CPLEX within 10 minutes for a given preprocessing parameter $\boldsymbol{u}$. This is denoted by "t_01QP". If we solve every instance within 10 minutes, we report the total CPU time. If there are $m < 100$ instances that are solvable within 10 minutes each, we report the total CPU time to solve these m instances and report the average time for the m solved instances in the parentheses.

4. The number of instances (out of 100) which are solved within 10 minutes is denoted by "solved".

From Table 4.1, we observe that when the standard deviation is small ($\boldsymbol{\sigma} = rand(N, 1)$), the relative gaps for $\boldsymbol{u}_{\boldsymbol{\mu}}$ and $\boldsymbol{u}_{\boldsymbol{\mu},\boldsymbol{\sigma}}$ are much smaller than the relative gap for $\boldsymbol{u}_{\text{eig}}$, and very close to the relative gap for the sample based method. Although the preprocessing parameter $\boldsymbol{u}_{\text{eig}}$ can be computed very efficiently, solving the QUBO problem is much slower than the other three methods. Since the standard deviation is of a similar magnitude as the mean, $\boldsymbol{u}_{\boldsymbol{\mu}}$ and $\boldsymbol{u}_{\boldsymbol{\mu},\boldsymbol{\sigma}}$ have similar relative gaps. These two methods are also much faster than finding the preprocessing step for the sample based method that involves solving 100 SDP instances.

From Table 4.2, we observe that when the standard deviation is larger ($\boldsymbol{\sigma} = 20 * rand(N, 1)$), the relative gap from the sample based method is much smaller than the gaps generated from the other three methods. In these cases, we have $\text{gap}(\boldsymbol{u}_{\text{opt}}^{(k)}) < \text{gap}(\boldsymbol{u}_{\boldsymbol{\mu},\boldsymbol{\sigma}}) < \text{gap}(\boldsymbol{u}_{\boldsymbol{\mu}}) < \text{gap}(\boldsymbol{u}_{\text{eig}})$. The CPU time taken to solve the QUBO problem by choosing $\boldsymbol{u}_{\boldsymbol{\mu},\boldsymbol{\sigma}}$ is smaller than that by choosing $\boldsymbol{u}_{\text{eig}}$ and $\boldsymbol{u}_{\boldsymbol{\mu}}$ in most cases, and it is close to the CPU time taken by usin g $\boldsymbol{u}_{\text{opt}}^{(k)}$. Lastly, the computational time needed for the preprocessing step for the mean and standard deviation based method is much smaller than that for the preprocessing step of the sample based method. As a result, the total CPU time needed to solve all the 100

Table 4.1: Gap and CPU time for different parameters $\boldsymbol{u}$ when $\boldsymbol{\mu} = randn(N,1), \boldsymbol{\sigma} = rand(N,1)$

N	d	$\boldsymbol{u} = \boldsymbol{u}_{\text{eig}}$			$\boldsymbol{u} = \boldsymbol{u}^{(k)}_{\text{opt}}, k \in [K]$				$\boldsymbol{u} = \boldsymbol{u}_{\boldsymbol{\mu}}$				$\boldsymbol{u} = \boldsymbol{u}_{\boldsymbol{\mu},\boldsymbol{\sigma}}$			
		gap	t_$\boldsymbol{u}$	t_01QP	solved	gap	t_$\boldsymbol{u}$	t_01QP	solved	gap	t_$\boldsymbol{u}$	t_01QP	solved	gap	t_$\boldsymbol{u}$	t_01QP	solved
50	0.4	18.5	0.01	101.7	100	5.0	48.76	15.3	100	5.4	0.47	15.0	100	5.6	1.64	12.8	100
50	0.6	13.6	0.02	50.7	100	3.5	45.57	13.2	100	3.9	0.44	13.5	100	4.0	1.66	12.5	100
50	1.0	18.6	0.02	196.9	100	8.4	46.53	32.8	100	8.9	0.46	30.3	100	8.9	1.53	26.3	100
60	0.2	12.0	0.02	161.8	100	2.7	52.46	14.6	100	3.1	0.67	14.3	100	3.2	1.79	12.0	100
60	0.4	22.2	0.02	3599.7	100	8.2	52.91	69.3	100	9.0	0.64	59.4	100	9.0	1.76	45.5	100
70	0.3	19.4	0.02	6355.1	100	4.9	59.50	51.8	100	5.5	0.79	49.9	100	5.5	2.06	35.6	100
70	0.8	19.3	0.02	19955.2	100	8.1	55.91	246.0	100	8.4	0.72	277.8	100	8.5	2.09	223.9	100
80	0.2	23.0	0.02	5504.5(458.7)	12	6.5	65.44	136.8	100	7.5	1.00	140.9	100	7.5	2.54	87.6	100
80	0.7	14.9	0.02	8469.4	100	6.5	67.46	334.0	100	6.8	0.86	327.2	100	6.9	2.82	318.2	100
90	0.3	19.9	0.02	**	0	7.2	90.18	899.0	100	7.6	1.03	923.0	100	7.7	3.55	830.6	100
100	0.1	20.1	0.02	**	0	4.7	103.20	229.3	100	5.7	1.26	238.7	100	5.7	4.09	175.0	100

Table 4.2: Gap and CPU time for different parameters $\boldsymbol{u}$ when $\boldsymbol{\mu} = randn(N,1), \boldsymbol{\sigma} = 20 * rand(N,1)$

N	d	$\boldsymbol{u} = \boldsymbol{u}_{\text{eig}}$			$\boldsymbol{u} = \boldsymbol{u}^{(k)}_{\text{opt}}, k \in [K]$				$\boldsymbol{u} = \boldsymbol{u}_{\boldsymbol{\mu}}$				$\boldsymbol{u} = \boldsymbol{u}_{\boldsymbol{\mu},\boldsymbol{\sigma}}$			
		gap	t_$\boldsymbol{u}$	t_01QP	solved	gap	t_$\boldsymbol{u}$	t_01QP	solved	gap	t_$\boldsymbol{u}$	t_01QP	solved	gap	t_$\boldsymbol{u}$	t_01QP	solved
50	0.4	10.0	0.01	18.9	100	2.6	46.19	9.2	100	8.0	0.46	9.5	100	5.5	1.54	9.3	100
50	0.6	9.4	0.02	27.9	100	2.5	45.24	11.9	100	8.4	0.44	13.2	100	5.7	1.48	11.9	100
50	1.0	11.6	0.01	70.0	100	4.7	45.88	20.6	100	10.6	0.45	21.7	100	8.2	1.52	19.2	100
60	0.2	5.6	0.02	6.3	100	0.8	54.04	6.2	100	5.1	0.59	6.2	100	2.7	1.78	6.1	100
60	0.4	8.5	0.02	39.3	100	2.4	53.68	11.9	100	6.9	0.61	13.4	100	5.6	1.79	12.8	100
70	0.3	8.5	0.02	100.4	100	2.0	61.98	17.1	100	7.3	1.08	23.5	100	4.8	2.15	20.7	100
70	0.8	12.2	0.02	2097.2(21.2)	99	4.0	60.15	80.5	100	9.3	0.89	115.7	100	7.1	2.07	105.7	100
80	0.2	6.9	0.02	203.5	100	1.4	77.65	12.6	100	5.4	0.90	19.4	100	3.9	3.01	17.4	100
80	0.7	9.3	0.02	2373.9	100	3.6	68.24	126.8	100	8.7	0.66	255.6	100	6.7	2.41	151.2	100
90	0.3	8.6	0.02	4638.1	100	2.6	85.46	88.9	100	7.0	0.83	215.3	100	5.7	2.76	170.9	100
100	0.1	6.6	0.02	39.7	100	1.2	99.15	9.9	100	5.3	0.94	12.9	100	3.6	3.11	12.3	100

QUBO problems to optimality using the preprocessing vector $\boldsymbol{u}_{\boldsymbol{\mu},\boldsymbol{\sigma}}$ is substantially smaller than that needed by the sample based preprocessing method.

4.4.2 Instances from Billionnet and Elloumi [25] and Pardalos and Rodgers [95]

We use the set of randomly generated instances as in Billionnet and Elloumi [25] and Pardalos and Rodgers [95]. We choose the parameters as follows:

1. The linear coefficients c_i are chosen uniformly and independently in the range $[-100, 100]$.
2. The diagonal entries of $\boldsymbol{Q} \in \Re^{N\times N}$ are all 0, and the off-diagonal coefficients of the symmetric matrix $\boldsymbol{Q}$ are in the range $[-50, 50]$.
3. The matrix $\boldsymbol{Q}$ has density d. The density refers to the probability that a nonzero will occur in any off-diagonal entry.

In this example, the data $\boldsymbol{c}^{(k)}$ for $K = 100$ samples are given. We use the sample mean and the sample standard deviation to compute the preprocessing parameters $\boldsymbol{u}_{\boldsymbol{\mu}}$ and $\boldsymbol{u}_{\boldsymbol{\mu},\boldsymbol{\sigma}}$. Again, we use the four different preprocessing methods to solve the QUBO problems, and the maximum CPU time taken to solve the QUBO problem is set to be 10 minutes. The results are listed in Table 4.3. In addition to the average gap, we plot the distributions of the relative gaps for the 100 scenarios using the boxplot in Figure 4.1. From the results, we observe that the average relative gap of using $\boldsymbol{u}_{\boldsymbol{\mu},\boldsymbol{\sigma}}$ is always smaller than using $\boldsymbol{u}_{\text{eig}}$ and $\boldsymbol{u}_{\boldsymbol{\mu}}$. In addition to the average value, from Figure 4.1 we observe that the relative gap of using $\boldsymbol{u}_{\boldsymbol{\mu},\boldsymbol{\sigma}}$ has a smaller sample minimum, lower quartile (25th percentile), median, upper quartile (75th percentile), and sample maximum than using $\boldsymbol{u}_{\text{eig}}$ and $\boldsymbol{u}_{\boldsymbol{\mu}}$. The relative gap using the sample based method is the smallest as should be expected. Hence, in terms of the relative gap between the optimal value of the QUBO problem and its convex

Table 4.3: Gap and CPU time for different parameters $\boldsymbol{u}$

N	d	$\boldsymbol{u} = \boldsymbol{u}_{\text{eig}}$			$\boldsymbol{u} = \boldsymbol{u}_{\text{opt}}^{(k)}, k \in [K]$			$\boldsymbol{u} = \boldsymbol{u}_{\boldsymbol{\mu}}$			$\boldsymbol{u} = \boldsymbol{u}_{\boldsymbol{\mu},\boldsymbol{\sigma}}$		
		gap	t_$\boldsymbol{u}$	t_01QP	gap	t_$\boldsymbol{u}$	t_01QP	solved	gap	t_$\boldsymbol{u}$	t_01QP	solved	gap	t_$\boldsymbol{u}$	t_01QP	solved
50	0.4	13.8	0.02	106.0	4.9	45.27	23.0	100	8.9	0.48	20.5	100	7.3	1.52	18.0	100
50	0.6	15.1	0.01	109.0	6.7	45.39	28.2	100	10.0	0.46	24.3	100	9.0	1.55	19.4	100
50	1.0	12.5	0.02	85.9	6.6	45.84	30.9	100	8.8	0.45	26.0	100	8.5	1.55	24.1	100
60	0.2	17.2	0.01	1022.4	5.8	53.90	43.5	100	10.7	0.63	53.6	100	8.7	1.82	35.8	100
60	0.4	14.0	0.01	1136.3	4.4	51.96	50.8	100	7.1	0.50	42.9	100	6.1	1.78	37.1	100
70	0.3	18.5	0.02	1249.4	8.2	58.50	280.0	100	12.2	0.69	292.5	100	10.9	2.20	261.5	100
80	0.2	18.4	0.03	16569.2(218.0)	7.8	70.68	450.9	100	12.4	0.67	545.1	100	10.8	2.40	435.3	100
90	0.6	18.3	0.03	13764.3(327.7)	9.1	83.60	7040.1	100	11.4	0.83	7095.1	100	11.0	3.26	6783.9	100
100	0.1	16.5	0.02	4761.2(297.6)	3.5	100.83	178.2	100	7.9	1.21	317.1	100	5.9	4.05	210.7	100
120	0.2	21.1	0.05	**	10.4	129.52	5770.7(360.7)	16	13.6	1.35	3610.4(361.0)	10	13.0	4.03	5500.0(343.8)	16

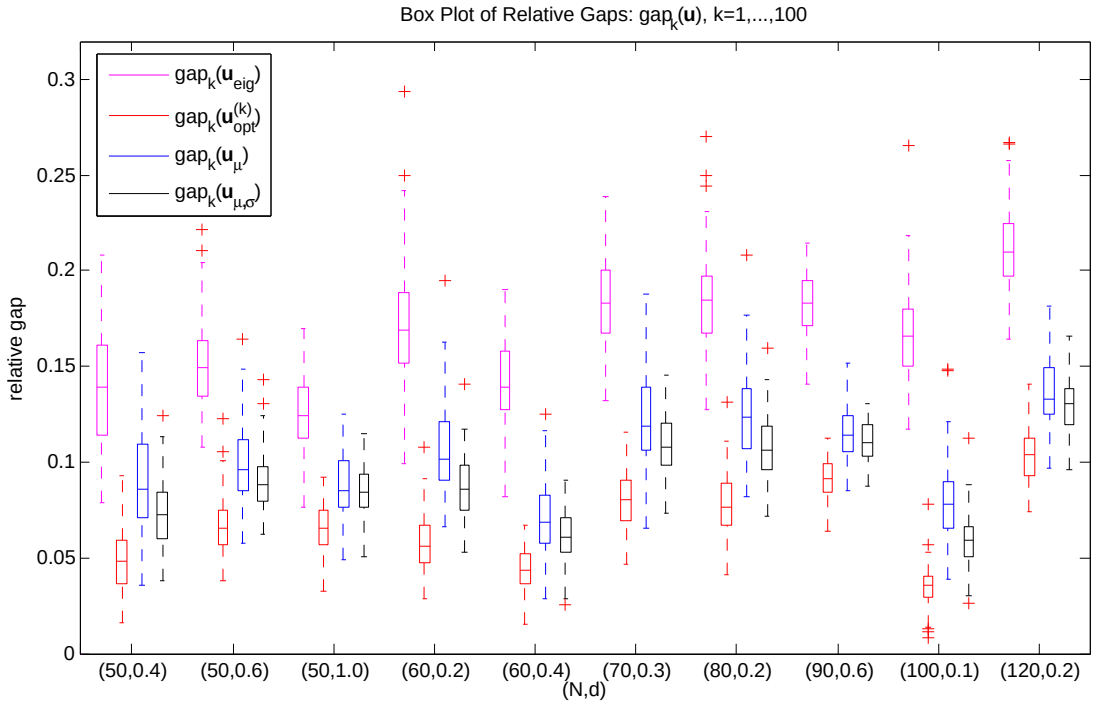
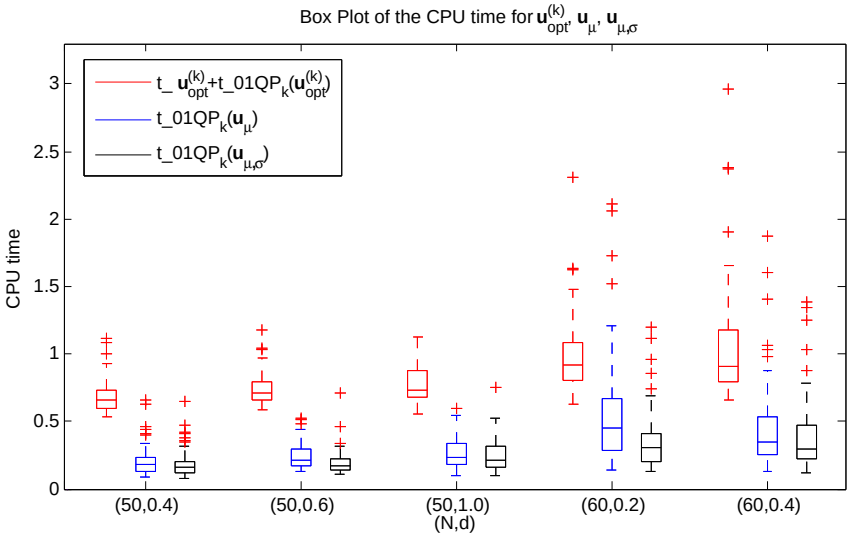


Figure 4.1: Boxplot of the Relative Gaps for all the 100 scenarios

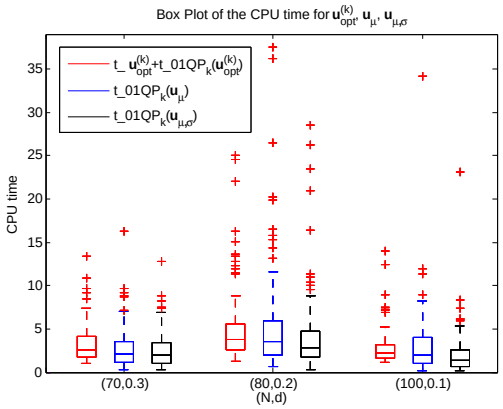
relaxation, parameter $\boldsymbol{u}_{\boldsymbol{\mu},\boldsymbol{\sigma}}$ is better than $\boldsymbol{u}_{\text{eig}}$ and $\boldsymbol{u}_{\boldsymbol{\mu}}$ and closest to the sample based method.

We also plot the CPU time taken to solve the QUBO problem for every scenario $\boldsymbol{c}^{(k)}$, $k \in [K]$. In Figure 4.2, "t_01QP$_k(\boldsymbol{u})$", denotes the CPU time taken to solve the convex QUBO problem with the preprocessing parameter $\boldsymbol{u}$ for scenario $\boldsymbol{c}^{(k)}$. Since the CPU time taken to solve the QUBO problem by using $\boldsymbol{u}_{\text{eig}}$ is much larger than the other three methods, we only focus on three methods excluding the eigenvalue based method. Since $\boldsymbol{u}_{\boldsymbol{\mu}}$ and $\boldsymbol{u}_{\boldsymbol{\mu},\boldsymbol{\sigma}}$ are common preprocessing parameters for all the 100 instances, and we can compute them quickly by solving a single SDP problem, the CPU time of getting $\boldsymbol{u}_{\boldsymbol{\mu}}$ and $\boldsymbol{u}_{\boldsymbol{\mu},\boldsymbol{\sigma}}$ is negligible in Figure 4.2. However, to use $\{\boldsymbol{u}_{\text{opt}}^{(k)},\ k \in [K]\}$ we must solve an SDP problem for every instance. Hence in the plot of the CPU time to solve the QUBO problem, the time (t_$\boldsymbol{u}_{\text{opt}}^{(k)}$) taken to compute $\boldsymbol{u}_{\text{opt}}^{(k)}$ is added.

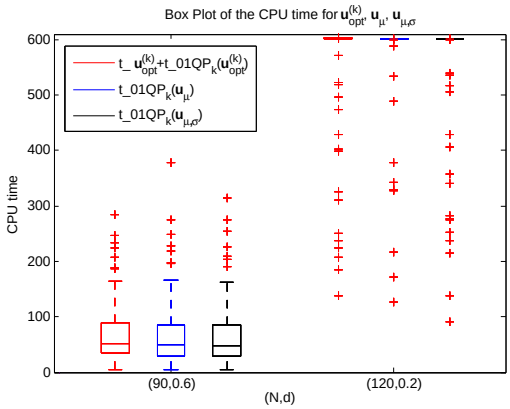
From Figure 4.2, we see that for the small size instances (subfigure (a)) and the medium size instances (subfigure (b)), the mean and standard deviation method



(a) Small Size Instances



(b) Medium Size Instances



(c) Hard to Solve Instances

Figure 4.2: Boxplot of the CPU Time: (for the instances which can not be solved in 10 minutes, we just plot its CPU time as 600 seconds in the figure)

is better than the sample based and mean based method. The CPU time of using $\boldsymbol{u}_{\boldsymbol{\mu},\boldsymbol{\sigma}}$ to solve the QUBO problem have the smallest sample minimum, lower quartile (25th percentile), median, upper quartile (75th percentile), and sample maximum. For the hard to solve instances (subfigure (c)), the three methods look more similar in Figure 4.2. From Table 4.3, we can see that using $\boldsymbol{u}_{\boldsymbol{\mu},\boldsymbol{\sigma}}$, we need the smallest CPU time to solve all the 100 instances when $N = 90, d = 0.6$. For the largest and most difficult set of instances with $N = 120, d = 0.2$, very few instances can be solved to optimality in 10 minutes. By using $\boldsymbol{u}_{\boldsymbol{\mu},\boldsymbol{\sigma}}$, we solve 16 instances to optimality which is the same as using the sample based method.

Robustness Tests using Permutations

Next, we test the robustness of the mean and standard deviation based method using permutation experiments. The Penalized QCR method in this paper is developed for the Fréchet class of distributions with fixed marginal distributions. However no assumption is made on the dependency structure between random variables. To test the robustness of the solutions, we generate other feasible distributions in this set by permuting the individual components of the randomly generated samples in the following manner. Given the sample data $\boldsymbol{c}^{(1)}, \ldots, \boldsymbol{c}^{(100)}$, we compute the sample mean $\boldsymbol{\mu}$ and the sample standard deviation $\boldsymbol{\sigma}$. For $i \in [N]$, we randomly permute the ith component sequence of the vectors $c_i^{(1)}, c_i^{(2)}, \ldots, c_i^{(100)}$. By performing this permutation independently for each $i \in [N]$, we generate a new set of samples $\{\bar{\boldsymbol{c}}^{(k)},\ k \in [K]\}$. See [91] for a similar set of experiments in the context of stochastic knapsack problems. Note that the sample mean $\boldsymbol{\mu}$ and the standard deviation $\boldsymbol{\sigma}$ will not change after these permutations. Hence the preprocessing parameter $\boldsymbol{u}_{\boldsymbol{\mu}}$ and $\boldsymbol{u}_{\boldsymbol{\mu},\boldsymbol{\sigma}}$ will not change. However clearly, $\boldsymbol{u}_{\text{opt}}^{(k)}$ might change since the samples have changed. As a control, we also use the average sample based preprocessing vector defined as $\boldsymbol{u}_{\text{ave}} := \sum_{k=1}^{100} \boldsymbol{u}_{\text{opt}}^{(k)}/100$. Since $\boldsymbol{Q} - \text{diag}(\boldsymbol{u}_{\text{opt}}^{(k)}) \succeq 0, \forall k \in [K]$, we have $\boldsymbol{Q} - \text{diag}(\boldsymbol{u}_{\text{ave}}) \succeq 0$.

For the tests, we use two sets of parameters ($(N, d) = (50, 0.6)$ and $(N, d) = (70, 0.3)$) from Table 4.3 and perform numerical tests for the samples after the

random permutations. For each set of data, we test the results across 15 permutations. The preprocessing vectors $\boldsymbol{u}_{\text{ave}}$, $\boldsymbol{u}_{\boldsymbol{\mu}}$ and $\boldsymbol{u}_{\boldsymbol{\mu},\boldsymbol{\sigma}}$ are obtained one time only and hence the CPU time of computing these preprocessing parameters is ignored. However, $u_{\text{opt}}^{(k)}$, $k \in [K]$ must be recalculated for every permutation. The numerical results are shown in Table 4.4 and 4.5.

Table 4.4: Gap and CPU time with 15 permutations: $N = 50$, $d = 0.6$

No.	$\boldsymbol{u} = \boldsymbol{u}_{\text{opt}}^{(k)}, k \in [K]$			$\boldsymbol{u} = \boldsymbol{u}_{\text{ave}}$			$\boldsymbol{u} = \boldsymbol{u}_{\boldsymbol{\mu}}$			$\boldsymbol{u} = \boldsymbol{u}_{\boldsymbol{\mu},\boldsymbol{\sigma}}$		
	gap	t_$\boldsymbol{u}$	t_01QP	solved	gap	t_01QP	solved	gap	t_01QP	solved	gap	t_01QP	solved
1	6.6	47.78	22.6	100	10.5	22.1	100	10.0	19.7	100	8.9	15.2	100
2	6.5	55.79	27.9	100	10.4	25.3	100	9.9	23.9	100	8.8	18.7	100
3	6.7	56.25	29.6	100	10.6	27.7	100	10.1	25.8	100	9.0	19.8	100
4	6.7	58.73	27.7	100	10.7	26.5	100	10.1	24.5	100	9.0	19.1	100
5	6.5	53.95	27.8	100	10.4	26.3	100	9.9	24.2	100	8.8	19.0	100
6	6.5	54.24	27.3	100	10.4	25.6	100	9.9	23.8	100	8.8	18.8	100
7	6.7	56.78	28.4	100	10.9	27.8	100	10.4	25.7	100	9.1	19.5	100
8	6.6	54.65	27.2	100	10.5	26.0	100	9.9	23.8	100	8.9	18.7	100
9	6.5	55.60	26.7	100	10.4	24.7	100	9.9	23.4	100	8.8	18.3	100
10	6.5	45.54	26.7	100	10.4	25.0	100	9.9	23.1	100	8.8	18.0	100
11	6.6	52.42	27.4	100	10.5	26.6	100	10.0	24.7	100	8.9	18.9	100
12	6.5	52.29	27.4	100	10.4	26.2	100	9.8	23.5	100	8.8	18.8	100
13	6.7	54.48	28.3	100	10.6	26.8	100	10.1	24.9	100	9.0	19.4	100
14	6.6	53.38	26.9	100	10.5	25.0	100	10.0	22.8	100	8.9	18.3	100
15	6.4	56.66	24.5	100	10.3	22.8	100	9.8	21.5	100	8.7	17.0	100

Table 4.5: Gap and CPU time with 15 permutations: $N = 70$, $d = 0.3$

No.	$\boldsymbol{u} = \boldsymbol{u}_{\text{opt}}^{(k)}, k \in [K]$			$\boldsymbol{u} = \boldsymbol{u}_{\text{ave}}$			$\boldsymbol{u} = \boldsymbol{u}_{\boldsymbol{\mu}}$			$\boldsymbol{u} = \boldsymbol{u}_{\boldsymbol{\mu},\boldsymbol{\sigma}}$		
	gap	t_$\boldsymbol{u}$	t_01QP	solved	gap	t_01QP	solved	gap	t_01QP	solved	gap	t_01QP	solved
1	8.3	62.42	303.7	100	13.1	500.2	100	12.3	322.9	100	11.0	278.3	100
2	8.3	63.63	275.1	100	13.1	498.1	100	12.2	300.0	100	11.0	267.7	100
3	8.5	62.99	340.7	100	13.4	580.9	100	12.6	368.2	100	11.3	328.8	100
4	8.3	62.75	275.3	100	13.1	481.2	100	12.2	293.1	100	11.0	267.7	100
5	8.2	62.87	291.5	100	13.0	489.0	100	12.2	301.8	100	10.9	267.4	100
6	8.4	63.07	339.4	100	13.1	600.4	100	12.2	350.8	100	11.0	321.6	100
7	8.2	62.71	278.8	100	13.1	453.9	100	12.3	288.4	100	11.0	265.8	100
8	8.2	62.96	271.5	100	13.0	471.8	100	12.2	286.9	100	10.9	258.2	100
9	8.2	62.95	274.4	100	13.1	493.4	100	12.2	307.3	100	10.9	264.0	100
10	8.1	62.76	246.7	100	12.9	423.5	100	12.0	264.0	100	10.8	237.4	100
11	8.3	63.12	276.3	100	13.2	471.6	100	12.4	297.0	100	11.0	261.6	100
12	8.2	63.09	279.6	100	13.1	501.2	100	12.2	302.1	100	10.9	266.1	100
13	8.2	63.15	282.8	100	13.0	485.3	100	12.2	302.2	100	10.9	266.5	100
14	8.5	63.37	330.5	100	13.4	579.2	100	12.6	350.0	100	11.2	311.7	100
15	8.2	66.04	293.7	100	13.0	497.6	100	12.2	311.1	100	10.9	274.0	100

From Tables 4.4 and 4.5, we see that by using $\boldsymbol{u}_{\boldsymbol{\mu},\boldsymbol{\sigma}}$ the average gap is smaller than using $\boldsymbol{u}_{\boldsymbol{\mu}}$ and $\boldsymbol{u}_{\text{ave}}$. Moreover the total CPU time taken to solve the QUBO problem is always the smallest for all the permutations by using $\boldsymbol{u}_{\boldsymbol{\mu},\boldsymbol{\sigma}}$. This shows that the mean and standard deviation based penalized QCR method is robust for the small and medium size instances.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

For the linear combinatorial optimization problem $\max_{\boldsymbol{x}\in\mathcal{X}\subseteq\{0,1\}^N} \tilde{\boldsymbol{c}}^T\boldsymbol{x}$ with uncertainty in the random vector $\tilde{\boldsymbol{c}}$, two probabilistic models were considered in this thesis. The simpler model is to minimize the WCVaR of cost:

$$\min_{\boldsymbol{x}\in\mathcal{X}} \mathrm{WCVaR}_\alpha(-\tilde{\boldsymbol{c}}^T\boldsymbol{x}). \tag{5.1}$$

With a fixed $\boldsymbol{x}\in\mathcal{X}$, we developed a tractable convex optimization reformulation for the subproblem $\mathrm{WCVaR}_\alpha(-\tilde{\boldsymbol{c}}^T\boldsymbol{x})$ under the marginal distribution and marginal moment models. Furthermore, we showed that problem (5.1) can be solved to optimality as a deterministic linear combinatorial optimization problem.

The other model we proposed was to minimize the WCVaR of regret for the random linear combinatorial optimization problem:

$$\min_{\boldsymbol{x}\in\mathcal{X}} \mathrm{WCVaR}_\alpha(\max_{\boldsymbol{y}\in\mathcal{X}} \tilde{\boldsymbol{c}}^T\boldsymbol{y} - \tilde{\boldsymbol{c}}^T\boldsymbol{x}). \tag{5.2}$$

This generalized the interval uncertainty minmax regret model by incorporating additional marginal distribution information on the data. By generalizing the earlier bounds of Meilijson and Nadas [88] to the regret framework, we proved a convex optimization formulation for $\mathrm{WCVaR}_\alpha(\max_{\boldsymbol{y}\in\mathcal{X}} \tilde{\boldsymbol{c}}^T\boldsymbol{y} - \tilde{\boldsymbol{c}}^T\boldsymbol{x})$ when $\boldsymbol{x}$ is fixed, and

showed the WCVaR of regret is computable in polynomial time if the deterministic combinatorial optimization problem is solvable in polynomial time. For the class of combinatorial optimization problems with a compact convex hull representation, a polynomial sized mixed integer linear program (MILP) is formulated under the discrete marginal distribution model. We also developed MILP formulations for the marginal moment model when (a) the range and mean are given and (b) the range, mean and mean absolute deviation are given. In the case (c) the range, mean and standard deviation are given, a mixed integer second order cone program was formulated.

For the subset selection problem of choosing a subset of K items from N randomly weighted items, we designed a polynomial time algorithm to solve the problem of minimizing the WCVaR of regret with complexity $O(N^2 J_{max}^2)$ under the discrete marginal distribution model, where J_{max} is the maximum among the numbers of the supporting points for the N random weights. This complexity is reduced to $O(N^2)$ under the marginal moment model when (a) the range and mean are given, or (b) the range, mean and mean absolute deviation are given. This polynomial time algorithm can be regarded as a generalization of the polynomial time algorithms designed by Conde [37] and Averbakh [9] for the minmax regret subset selection problem with range information only. The numerical results showed the proposed polynomial time algorithm is fast and stable in comparison with the general purpose mixed integer linear programming solver in CPLEX. This approach was also used to develop the polynomial algorithm for the distributionally robust k-sum optimization problem. It can also be used to show that when the linear sum combinatorial optimization problem is polynomially solvable, the k-sum optimization problem is also polynomially solvable.

Finally, we designed a Penalized QCR method to find the "optimal" preprocessing parameter for the quadratic unconstrained binary optimization problem with random linear coefficients. The SDP formulation for the random version of the problem can be viewed as a penalized version of the SDP used for deterministic

QUBO problems. Using this SDP formulation, we found a common preprocessing parameter for a set of instances which differs only in the linear term of the objective. Computationally, we showed that by using limited probabilistic information such as the mean and variance and solving a single SDP across random instances of the problem, we can obtain significant computational advantages over alternative preprocessing methods.

5.2 Future Work

5.2.1 Linear Programming Reformulation and Polynomial Time Algorithm

In this thesis, we have designed a polynomial time algorithm for the problem of minimizing the WCVaR of regret for the subset selection problem under the discrete marginal distribution model. Furthermore this polynomial time algorithm also solves the problem to optimality in the marginal moment model when (a) the range and mean are given, or (b) the range, mean and mean absolute deviation are given. Since most of the polynomially solvable integer linear programming problems have equivalent linear programming (LP) reformulations, a natural open question is:

Question 5.1. *Can we find an equivalent LP reformulation for the problem of minimizing the* WCVaR *of regret under the discrete marginal moment model for the subset selection problem?*

To consider Question 5.1, we can first try to to find an equivalent LP reformulation for the minmax regret subset selection problem in the interval uncertainty case.

In the marginal moment model, when (c) the range, mean and standard deviation are given, we do not know the computational complexity of the probabilistic regret problem for the subset selection problem. An open question is:

Question 5.2. *What is the complexity of minimizing the* WCVaR *of regret for the subset selection problem in the marginal moment model when the range, mean and standard deviation are given?*

Recall the probabilistic regret model for the subset selection problem

$$\min_{\boldsymbol{x}\in\mathcal{X}} \mathrm{WCVaR}_\alpha(Z(\tilde{\boldsymbol{c}}) - \tilde{\boldsymbol{c}}^T\boldsymbol{x}), \tag{5.3}$$

where $\mathcal{X} = \left\{x \in \{0,1\}^N : \sum_{i=1}^N x_i = K\right\}$, and $Z(\tilde{\boldsymbol{c}}) = \max_{\boldsymbol{x}\in\mathcal{X}} \tilde{\boldsymbol{c}}^T\boldsymbol{x}$. In Chapter 3, we showed that under the marginal moment model, problem (5.3) is formulated as

$$\min_{\lambda_0,\boldsymbol{x}\in\mathcal{X},\boldsymbol{d}\in\Omega} \sum_{i=1}^N F_i(d_i, x_i, \lambda_0) + K\lambda_0, \tag{5.4}$$

where

$$F_i(d_i, x_i, \lambda_0) = [d_i - \lambda_0]^+ + \frac{\alpha}{1-\alpha} d_i x_i + \frac{1}{1-\alpha} \sup_{P_i\in\mathbb{P}_i} \mathbb{E}_{P_i}[\tilde{c}_i - d_i]^+ - \frac{1}{1-\alpha}\mu_i x_i.$$

In the cases (a) and (b), the worst-case distribution for the item $\sup_{P_i\in\mathbb{P}_i} \mathbb{E}_{P_i}[\tilde{c}_i - d_i]^+$ are two-point and three-point discrete distributions which are independent of the variable d_i, see Ben-Tal and Hochman [16]. Hence the same analysis for the discrete marginal distribution model can be used to the marginal moment model in the cases (a) and (b). However, in the case (c) the range, mean and standard deviation are given, the worst-case distribution for the item $\sup_{P_i\in\mathbb{P}_i} \mathbb{E}_{P_i}[\tilde{c}_i - d_i]^+$ is a two-point discrete distribution, but it depends on the variables d_i, (see Birdge and Maddox [30]). Hence the same analysis for the discrete marginal distribution model cannot be applied to the case (c). In the case (c), finding a polynomial algorithm of the probabilistic regret model for the subset selection problem or showing it is NP-hard remains an open question.

5.2.2 WCVaR of Cost and Regret in Cross Moment Model

The distributional model we considered in the thesis is the Fréchet class of distributions, i.e. the marginal distribution model and marginal moment model. However, no correlation information has been involved in these models. Consider the

combinatorial optimization problem $Z(\tilde{\boldsymbol{c}}) := \max_{\boldsymbol{x}\in\mathcal{X}} \tilde{\boldsymbol{c}}^T\boldsymbol{x}$ with uncertainty in the random vector $\tilde{\boldsymbol{c}}$. To incorporate the correlation information of $\tilde{\boldsymbol{c}}$, the simplest model to be considered is the cross moment model, that is we assume the mean and the covariance matrix of the random vector $\tilde{\boldsymbol{c}}$ is given. Let $\tilde{\boldsymbol{c}} \sim (\boldsymbol{\mu}, \boldsymbol{\Sigma})$, i.e. the distribution of $\tilde{\boldsymbol{c}}$ lies in the distributional set with mean equals to $\boldsymbol{\mu}$ and covariance matrix equals to $\boldsymbol{\Sigma}$. First we consider the problem of minimizing the WCVaR of cost:

$$\min_{\boldsymbol{x}\in\mathcal{X}} \mathrm{WCVaR}_\alpha(-\tilde{\boldsymbol{c}}^T\boldsymbol{x}) = \min_{\boldsymbol{x}\in\mathcal{X}, v\in\Re} \left\{ v + \frac{1}{1-\alpha} \sup_{\tilde{\boldsymbol{c}}\sim(\boldsymbol{\mu},\boldsymbol{\Sigma})} \mathbb{E}[-\tilde{\boldsymbol{c}}^T\boldsymbol{x} - v]^+ \right\}. \tag{5.5}$$

Using the projection property in Popescu [99], the sup problem in (5.5) can be formulated as a univariate mean-variance distributionally robust optimization problem, hence we obtain

$$\begin{aligned}
\sup_{\tilde{\boldsymbol{c}}\sim(\boldsymbol{\mu},\boldsymbol{\Sigma})} \mathbb{E}_P[-\tilde{\boldsymbol{c}}^T\boldsymbol{x} - v]^+ =& \sup_{\tilde{r}\sim(\boldsymbol{\mu}^T\boldsymbol{x}, \boldsymbol{x}^T\boldsymbol{\Sigma}\boldsymbol{x})} \mathbb{E}[-\tilde{r} - v]^+ \\
=& \frac{1}{2}\left[(\boldsymbol{\mu}^T\boldsymbol{x} - v) + \sqrt{(\boldsymbol{\mu}^T\boldsymbol{x} + v)^2 + \boldsymbol{x}^T\boldsymbol{\Sigma}\boldsymbol{x}}\right] \\
=& \min_{t\in\Re} \frac{1}{2}\left[-\boldsymbol{\mu}^T\boldsymbol{x} - v + t\right] \\
& \text{s.t. } t \geq \left\| \begin{pmatrix} \boldsymbol{\mu}^T\boldsymbol{x} + v \\ \boldsymbol{\Sigma}^{1/2}\boldsymbol{x} \end{pmatrix} \right\|_2 .
\end{aligned}$$

Therefore (5.5) can be formulated as the following mixed integer second order cone program (SOCP)

$$\begin{aligned}
\min_{\boldsymbol{x}\in\mathcal{X}} \mathrm{WCVaR}_\alpha(-\tilde{\boldsymbol{c}}^T\boldsymbol{x}) = \quad & \min_{\boldsymbol{x},v,t} \quad \left\{ v + \frac{1}{2(1-\alpha)}(-\boldsymbol{u}^T\boldsymbol{x} - v + t) \right\} \\
& \text{s.t.} \quad t \geq \left\| \begin{pmatrix} \boldsymbol{\mu}^T\boldsymbol{x} + v \\ \boldsymbol{\Sigma}^{1/2}\boldsymbol{x} \end{pmatrix} \right\|_2 \\
& \quad\; \boldsymbol{x} \in \mathcal{X}.
\end{aligned}$$

As in the SOCP formulation of minimizing the WCVaR of cost, it is important to analyze the WCVaR of regret under correlation information. The question is:

Question 5.3. *To find computationally implementable formulations for the following problem*

$$\min_{\boldsymbol{x}\in\mathcal{X}} \mathrm{WCVaR}_{\alpha}(Z(\tilde{\boldsymbol{c}}) - \tilde{\boldsymbol{c}}^T\boldsymbol{x}) = \min_{\boldsymbol{x}\in\mathcal{X}, v\in\Re} \left\{ v + \frac{1}{1-\alpha} \sup_{\tilde{\boldsymbol{c}}\sim(\boldsymbol{\mu},\boldsymbol{\Sigma})} \mathbb{E}[Z(\tilde{\boldsymbol{c}}) - \tilde{\boldsymbol{c}}^T\boldsymbol{x} - v]^+ \right\}. \tag{5.6}$$

A problem related to (5.6) is

$$\sup_{\tilde{\boldsymbol{c}}\sim(\boldsymbol{\mu},\boldsymbol{\Sigma})} \mathbb{E}[Z(\tilde{\boldsymbol{c}})]. \tag{5.7}$$

(5.7) is NP-hard, and a copositive programming reformulation was proposed in Natarajan et al. [93]. There is potential for extending the techniques therein to solve these classes of problems.

5.2.3 Random Quadratic Optimization with Constraints

In the thesis, a Penalized QCR method is designed to solve the random quadratic unconstrained binary optimization problems. Further research can be done for more general quadratic programs with constraints. For the deterministic linearly-constrained binary quadratic program, Billionnet et. al [29] developed the QCR method by a tight convex reformulation. Similar to the QCR method for unconstrained binary quadratic program, the SDP formulation is used to find the "optimal" preprocessing parameter. They also extended the QCR method to general mixed integer programs [28]. It is natural to extend the Penalized QCR method to more general binary quadratic optimization with constraints. A future research question is:

Question 5.4. *To develop a Penalized QCR method for the random binary quadratic optimization with linear constraints:*

$$\max\left\{q(\boldsymbol{x};\boldsymbol{c},\boldsymbol{Q}) := \boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c}^T\boldsymbol{x} \mid \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x}\in\{0,1\}^N\right\},\ \forall \boldsymbol{c}\in\mathcal{C}. \tag{5.8}$$

Furthermore, it would be interesting to consider the problems with quadratic constraints and generalize the method to problems with random $\boldsymbol{Q}$ matrices.

Bibliography

[1] C. Acerbi and D. Tasche. On the coherence of expected shortfall. *Journal of Banking and Finance*, 26(7):1487–1503, 2002.

[2] W. P. Adams and H. D. Sherali. Mixed-integer bilinear programming problems. *Mathematical Programming*, 59(1-3):279–305, 1993.

[3] S. Ahmed, U. Çakmak, and A. Shapiro. Coherent risk measures in inventory problems. *European Journal of Operational Research*, 182(1):226–238, 2007.

[4] H. Aissi, C. Bazgan, and D. Vanderpooten. Min-max and min-max regret versions of combinatorial optimization problems: A survey. *European Journal of Operational Research*, 197(2):427–438, 2009.

[5] B. Alidaee, G. A. Kochenberger, and A. Ahmadian. 0-1 quadratic programming approach for optimum solutions of two scheduling problems. *International Journal of Systems Science*, 25(2):401–408, 1994.

[6] K. Allemand, K. Fukuda, T. M. Liebling, and E. Steiner. A polynomial case of unconstrained zero-one quadratic optimization. *Mathematical Programming*, 91(1):49–52, 2001.

[7] J. Ang, F. Meng, and J. Sun. Two-stage stochastic linear programs with incomplete information on uncertainty. *European Journal of Operational Research*, 233(1):16–22, 2014.

[8] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath. Coherent measures of risk. *Mathematical Finance*, 9(3):203–228, 1999.

[9] I. Averbakh. On the complexity of a class of combinatorial optimization problems with uncertainty. *Mathematical Programming*, 90(2):263–272, 2001.

[10] I. Averbakh and V. Lebedev. Interval data minmax regret network optimization problems. *Discrete Applied Mathematics*, 138(3):289–301, 2004.

[11] F. Barahona. A solvable case of quadratic 0–1 programming. *Discrete Applied Mathematics*, 13(1):23–26, 1986.

[12] F. Barahona, M. Grötschel, M. Jünger, and G. Reinelt. An application of combinatorial optimization to statistical physics and circuit layout design. *Operations Research*, 36(3):493–513, 1988.

[13] F. Barahona, M. Jünger, and G. Reinelt. Experiments in quadratic 0–1 programming. *Mathematical Programming*, 44(1):127–137, 1989.

[14] J. E. Beasley. Heuristic algorithms for the unconstrained binary quadratic programming problem. *London, UK: Management School, Imperial College*, 4, 1998.

[15] A. Ben-Tal, L. El Ghaoui, and A.S. Nemirovskiĭ. *Robust Optimization*. Princeton University Press, 2009.

[16] A. Ben-Tal and E. Hochman. More bounds on the expectation of a convex function of a random variable. *Journal of Applied Probability*, 9(4):803–812, 1972.

[17] A. Ben-Tal and Nemirovskiĭ. *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*.

[18] A. Ben-Tal and M. Teboulle. An old-new concept of convex risk measures: The optimized certainty equivalent. *Mathematical Finance*, 17(3):449–476, 2007.

[19] Fernando P. Bernardo and Pedro M. Saraiva. Robust optimization framework for process parameter and tolerance design. *AIChE journal*, 44(9):2007–2017, 1998.

[20] D. Bertsimas, K. Natarajan, and C. P. Teo. Persistence in discrete optimization under data uncertainty. *Mathematical Programming*, 108(2-3):251–274, 2006.

[21] D. Bertsimas, K. Natarajan, and C.P. Teo. Probabilistic combinatorial optimization: Moments, semidefinite programming, and asymptotic bounds. *SIAM Journal on Optimization*, 15(1):185–209, 2004.

[22] D. Bertsimas and I. Popescu. Optimal inequalities in probability theory: A convex optimization approach. *SIAM Journal on Optimization*, 15(3):780–804, 2005.

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

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

[25] A. Billionnet and S. Elloumi. Using a mixed integer quadratic programming solver for the unconstrained quadratic 0-1 problem. *Mathematical Programming*, 109(1):55–68, 2007.

[26] A. Billionnet, S. Elloumi, and A. Lambert. A branch and bound algorithm for general mixed-integer quadratic programs based on quadratic convex relaxation. *To appear in Journal of Combinatorial Optimization*, 2012.

[27] A. Billionnet, S. Elloumi, and A. Lambert. An efficient compact quadratic convex reformulation for general integer quadratic programs. *To appear in Computational Optimization and Applications*, 2012.

[28] A. Billionnet, S. Elloumi, and A. Lambert. Extending the QCR method to general mixed-integer programs. *Mathematical Programming*, 131(1-2):381–401, 2012.

[29] A. Billionnet, S. Elloumi, and M-C. Plateau. Improving the performance of standard solvers for quadratic 0-1 programs by a tight convex reformulation: The qcr method. *Discrete Applied Mathematics*, 157(6):1185–1197, 2009.

[30] J. R. Birge and M. J. Maddox. Bounds on expected project tardiness. *Operations Research*, 43(5):838–850, 1995.

[31] E. Boros and P. L. Hammer. The max-cut problem and quadratic 0-1 optimization; polyhedral aspects, relaxations and bounds. *Annals of Operations Research*, 33(3):151–180, 1991.

[32] E. Boros, P. L. Hammer, and G. Tavares. Preprocessing of unconstrained quadratic binary optimization. *Technical Report RRR 10-2006, RUTCOR*, 2006.

[33] D.B. Brown. Large deviations bounds for estimating conditional value-at-risk. *Operations Research Letters*, 35(6):722–730, 2007.

[34] M. W. Carter. The indefinite zero-one quadratic problem. *Discrete Applied Mathematics*, 7(1):23–44, 1984.

[35] W. Chen, M. Sim, J. Sun, and C.P. Teo. From cvar to uncertainty set: Implications in joint chance-constrained optimization. *Operations Research*, 58:470–485, 2010.

[36] Millie Chu, Yuriy Zinchenko, Shane G Henderson, and Michael B. Sharpe. Robust optimization for intensity modulated radiation therapy treatment

planning under uncertainty. *Physics in Medicine and Biology*, 50(23):5463, 2005.

[37] E. Conde. An improved algorithm for selecting p items with uncertain returns according to the minmax-regret criterion. *Mathematical Programming*, 100(2):345–353, 2004.

[38] E. Conde. A minmax regret approach to the critical path method with task interval times. *European Journal of Operational Research*, 197(1):235–242, 2009.

[39] G. Dall'aglio. Fréchet classes: the beginnings. *Advances in Probability Distributions with Given Marginals, G. Dallaglio, S. Kotz, S., and G. Salinetti, editors, Kluwer Academic Publishers, Dordrecht*, pages 1–12.

[40] G. Dall'aglio. Fréchet classes and compatibility of distribution functions. *Symposia Math.*, 9:131–150, 1972.

[41] George B. Dantzig. Linear programming under uncertainty. *Management Science*, 1(3-4):197–206, 1955.

[42] Erick Delage and Yinyu Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations Research*, 58(3):595–612, 2010.

[43] M. Demir, B. Tansel, and G. Scheuenstuhl. Tree network 1-median location with interval data: a parameter space-based approach. *IIE Transactions*, 37(5):429–439, 2005.

[44] X. V. Doan and K. Natarajan. On the complexity of non-overlapping multivariate marginal bounds for probabilistic combinatorial optimization. *Operations Research*, 60(1):138–149, 2012.

[45] L. Duan, S. Xiaoling, Shenshen G., Jianjun G., and Chunli L. Polynomially solvable cases of binary quadratic programs. *Optimization and Optimal Control, Springer Optimization and Its Applications*, pages 199–225, 2010.

[46] Frank J. Fabozzi, Petter N. Kolm, Dessislava Pachamanova, Sergio M. Focardi, et al. *Robust portfolio optimization and management*. Wiley, 2007.

[47] H. Föllmer and A. Schied. Convex measures of risk and trading constraints. *Finance and Stochastics*, 6(4):429–447, 2002.

[48] M. Frittelli and E. R. Gianin. Putting order in risk measures. *Journal of Banking and Finance*, 26(7):1473–1486, 2002.

[49] Harold N. Gabow and Robert E. Tarjan. Algorithms for two bottleneck optimization problems. *Journal of Algorithms*, 9(3):411–417, 1988.

[50] Letchford A. N. Galli, A. A compact variant of the qcr method for 0-1 quadratically constrained quadratic programs. *Working Paper*, 2013.

[51] M. R. Garey and D. S. Johnson. *Computers and intractability*, volume 174. 1979.

[52] S. Geetha and KPK. Nair. A stochastic bottleneck transportation problem. *Journal of the Operational Research Society*, pages 583–588, 1994.

[53] B. Ghaddar, J. C. Vera, and M. F. Anjos. Second-order cone relaxations for binary quadratic polynomial programs. *SIAM Journal of Optimization*, 21(1):391–414, 2011.

[54] F. Glover. Improved linear integer programming formulations of nonlinear integer problems. *Management Science*, 22(4):455–460, 1975.

[55] F. Glover, G. A. Kochenberger, and B. Alidaee. Adaptive memory search for binary quadratic programs. *Management Science*, 44(3):336–345, 1998.

[56] F. Glover and E. Woolsey. Further reduction of zero-one polynomial programming problems to zero-one linear programming problems. *Operations Research*, 21(1):156–161, 1973.

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

[58] J. Goh and M. Sim. Distributionally robust optimization and its tractable approximations. *Operations Research*, 58(4-Part-1):902–917, 2010.

[59] M. Grant and S. Boyd. Graph implementations for nonsmooth convex programs. *Recent Advances in Learning and Control (a tribute to M. Vidyasagar), V. Blondel, S. Boyd, and H. Kimura, editors, Lecture Notes in Control and Information Sciences*, pages 95–110, 2008.

[60] M. Grant and S. Boyd. Cvx research, inc. cvx: Matlab software for disciplined convex programming, version 2.0. beta. 2012.

[61] O. Gross. The bottleneck assignment problem. Technical report, DTIC Document, 1959.

[62] Martin Grötschel, László Lovász, and Alexander Schrijver. Geometric algorithms and combinatorial optimization. *Algorithms and Combinatorics*, 2.

[63] SK. Gupta and AP. Punnen. k-sum optimization problems. *Operations Research Letters*, 9(2):121–126, 1990.

[64] P. L. Hammer, P. Hansen, and B. Simeone. Roof duality, complementation and persistency in quadratic 0-1optimization. *Mathematical Programming*, 28:121–155, 1984.

[65] P.L. Hammer and A.A Rubin. Some remarks on quadratic programming with 0-1 variables. *Revue Francaise Informatique et de Recherche Operationnelle*, 4(3):67–79, 1970.

[66] W. K. Klein Haneveld. Robustness against dependence in PERT: An application of duality and distributions with known marginals. *Mathematical Programming Studies*, 27:153–182, 1986.

[67] Iiro Harjunkoski, Tapio Westerlund, and Ray Pörn. Numerical and environmental considerations on a complex industrial mixed integer nonlinear programming (minlp) problem. *Computers & Chemical Engineering*, 23(10):1545–1561, 1999.

[68] T. Hayashi. Regret aversion and opportunity dependence. *Journal of Economic Theory*, 139(1):242–268, 2008.

[69] C. Helmberg and F. Rendl. Solving quadratic $(0, 1)$-problems by semidefinite programs and cutting planes. *Mathematical Programming*, 82(3):291–315, 1998.

[70] W.L. Hsu and G.L. Nemhauser. Easy and hard bottleneck location problems. *Discrete Applied Mathematics*, 1(3):209–215, 1979.

[71] H. Ishii and T. Nishida. Stochastic bottleneck spanning tree problem. *Networks*, 13(3):443–449, 1983.

[72] K. Isii. On the sharpness of tchebycheff-type inequalities. *Annals of the Institute of Statistical Mathematics*, 14:185–197, 1963.

[73] S. Ji, Xiaojin Z., and Xiaoling S. An improved convex 0-1 quadratic program reformulation for quadratic knapsack problems. *Pacific Journal of Optimization*, 8(1):75–87, 2012.

[74] M. Jünger, A. Martin, G. Reinelt, and R. Weismantel. Quadratic 0/1 optimization and a decomposition approach for the placement of electronic circuits. *Mathematical Programming*, 63(1):257–279, 1994.

[75] J. Kamburowski. A note on the stochastic shortest route problem. *Operations Research*, 33(3):696–698, 1985.

[76] A. Kasperski. *Discrete Optimization with Interval Data: Minmax Regret and Fuzzy Approach*. Springer Verlag, 2008.

[77] A. Kasperski and P. Zieliński. The robust shortest path problem in series-parallel multidigraphs with interval data. *Operations Research Letters*, 34(1):69–76, 2006.

[78] A. Kasperski and P. Zieliński. On the existence of an FPTAS for minmax regret combinatorial optimization problems with interval data. *Operations Research Letters*, 35(4):525–532, 2007.

[79] J.E. Kelley. Critical-path planning and scheduling: Mathematical basis. *Operations Research*, pages 296–320, 1961.

[80] Pramod P. Khargonekar, Ian R. Petersen, and Kemin Zhou. Robust stabilization of uncertain linear systems: quadratic stabilizability and H^∞ control theory. *Automatic Control, IEEE Transactions on*, 35(3):356–361, 1990.

[81] S. Kim and M. Kojima. Second order cone programming relaxation of non-convex quadratic optimization problems. *Optimization Methods and Software*, 15(3-4):201–224, 2001.

[82] F. Körner. A tight bound for the boolean quadratic optimization problem and its use in a branch and bound algorithm. *Optimization*, 19(5):711–721, 1988.

[83] P. Kouvelis and G. Yu. *Robust discrete optimization and its applications*, volume 14. Kluwer Academic Publishers, 1997.

[84] J. B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001.

[85] J. B. Lasserre. A "joint+marginal" approach to parametric polynomial optimization. *SIAM Journal of Optimization*, 20:1995–2022, 2010.

[86] Shanthikumar J. G. Lim, A.E.B. and Vahn G. Y. Robust portfolio choice with learning in the framework of regret: Single period case. *To appear in Management Science*, 2012.

[87] A. Madansky. Bounds on the expectation of a convex function of a multivariate random variable. *The Annals of Mathematical Statistics*, 30(3):743–746, 1959.

[88] I. Meilijson and A. Nadas. Convex majorization with an application to the length of critical paths. *Journal of Applied Probability*, 16(3):671–677, 1979.

[89] M. Muramatsu and T. Suzuki. A new second-order cone programming relaxation for max-cut problems. *Journal of Operations Research of Japan*, 43:164–177, 2003.

[90] K. Natarajan, M. Sim, and J. Uichanco. Tractable robust expected utility and risk models for portfolio optimization. *Mathematical Finance*, 20(4):695–731, 2010.

[91] K. Natarajan, M. Song, and C. P. Teo. Persistency model and its applications in choice modeling. *Management Science*, 55(3):453–469, 2009.

[92] K. Natarajan, C.P. Teo, and Z. Zheng. Distribution of the optimal value of a stochastic mixed zero-one linear optimization problem under objective uncertainty. *Working paper*, 2011.

[93] K. Natarajan, C.P. Teo, and Z. Zheng. Mixed 0-1 linear programs under objective uncertainty: A completely positive representation. *Operations research*, 59(3):713–728, 2011.

[94] Y. Nesterov. *Structure of non-negative polynomials and optimization problems*, volume 200. Kluwer Academic Publishers, 1997.

[95] P. M. Pardalos and G. P. Rodgers. Computational aspects of a branch and bound algorithm for quadratic zero-one programming. *Computing*, 45(2):131–144, 1990.

[96] G. Perakis and G. Roels. Regret in the newsvendor model with partial information. *Operations Research*, 56(1):188–203, 2008.

[97] J.C. Picard and H.D. Ratliff. Minimum cuts and related problems. *Networks*, 5(4):357–370, 1975.

[98] S. Poljak, F. Rendl, and H. Wolkowicz. A recipe for semidefinite relaxation for (0, 1)-quadratic programming. *Journal of Global Optimization*, 7(1):51–73, 1995.

[99] I. Popescu. Robust mean-covariance solutions for stochastic optimization. *Operations Research*, 55(1):98–112, 2007.

[100] Abraham. P. Punnen and YP. Aneja. On k-sum optimization. *Operations Research Letters*, 18(5):233–236, 1996.

[101] D. Reich and L. Lopes. The most likely path on series-parallel networks. *Networks*, 2010.

[102] R. T. Rockafellar. *Convex analysis*, volume 28. Princeton Univ Pr, 1997.

[103] R. T. Rockafellar and S. Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2:21–42, 2000.

[104] R. T. Rockafellar and S. Uryasev. Conditional value-at-risk for general loss distributions. *Journal of Banking and Finance*, 26:1443–1471, 2002.

[105] R.T. Rockafellar. Saddle-points and convex analysis. *Differential games and related topics,H.W. Kuhn and G.P. Szego, eds.*, pages 109–128, 1971.

[106] Maria Analia Rodriguez and Aldo Vecchiettia. Disjunctive linearization technique for solving problems with bilinear terms. *20th European Symposium of Computer Aided Process Engineering–ESCAPE20*, 2010.

[107] L. J. Savage. The theory of statistical decision. *Journal of the American Statistical Association*, 46(253):55–67, 1951.

[108] H. Scarf. A min-max solution of an inventory problem. *Studies in The Mathematical Theory of Inventory and Production*, pages 201–209, 1958.

[109] A. Schrijver. *Theory of linear and integer programming*. Wiley, 1998.

[110] Hanif D. Sherali and Amine Alameddine. A new reformulation-linearization technique for bilinear programming problems. *Journal of Global Optimization*, 2(4):379–410, 1992.

[111] N. Z Shor. Class of global minimum bounds of polynomial functions. *Cybernetics and Systems Analysis*, 23(6):731–734, 1987.

[112] M. Sim. *Robust optimization*. PhD thesis, Massachusetts Institute of Technology, 2004.

[113] C. Simone, M. Diehl, M. Jünger, P. Mutzel, G. Reinelt, and G. Rinaldi. Exact ground states of ising spin glasses: new experimental results with a branch-and-cut algorithm. *Journal of Statistical Physics*, 80(1):487–496, 1995.

[114] A. M-C So, J. Zhang, and Y. Ye. Stochastic combinatorial optimization with controllable risk aversion. *Mathematics of Operations Research*, 34(3):522–537, 2009.

[115] HM Soroush. The most critical path in a pert network. *Journal of the Operational Research Society*, pages 287–300, 1994.

[116] K. C. Toh, M. J. Todd, and R. H. Tutuncu. SDPT3 — a matlab software package for semidefinite programming. *Optimization Methods and Software*, 11:545–581, 1999.

[117] K. C. Toh, M. J. Todd, and R. H. Tutuncu. Solving semidefinite-quadratic-linear programs using SDPT3. *Mathematical Programming*, 95:189–217, 2003.

[118] A. Wald. Contributions to the theory of statistical estimation and testing hypotheses. *The Annals of Mathematical Statistics*, 10(4):299–326, 1939.

[119] A. Wald. Statistical decision functions which minimize the maximum risk. *The Annals of Mathematics*, 46(2):265–280, 1945.

[120] G. Weiss. Stochastic bounds on distributions of optimal value functions with applications to pert, network flows and reliability. *Operations Research*, 34(4):595–605, 1986.

[121] H. Yaman, O. E. Karaşan, and M. C. Pinar. The robust spanning tree problem with interval data. *Operations Research Letters*, 29(1):31–40, 2001.

[122] U. Yechiali. A stochastic bottleneck assignment problem. *Management Science*, 14(11):732–733, 1968.

[123] Chian-Son Yu and Han-Lin Li. A robust optimization model for stochastic logistic problems. *International Journal of Production Economics*, 64(1):385–397, 2000.

[124] J. Yue, B. Chen, and M.C. Wang. Expected value of distribution information for the newsvendor problem. *Operations Research*, 54(6):1128–1136, 2006.

[125] S. Zhu and M. Fukushima. Worst-case conditional value-at-risk with application to robust portfolio management. *Operations Research*, 57(5):1155–1168, 2009.

[126] P. Zieliński. The computational complexity of the relative robust shortest path problem with interval data. *European Journal of Operational Research*, 158(3):570–576, 2004.

[127] S. Zymler, D. Kuhn, and B. Rustem. Distributionally robust joint chance constraints with second-order moment information. *Mathematical Programming*, 137(1-2):167–198, 2013.

REGRET MODELS AND PREPROCESSING TECHNIQUES FOR COMBINATORIAL OPTIMIZATION UNDER UNCERTAINTY

SHI DONGJIAN

NATIONAL UNIVERSITY OF SINGAPORE

2013