Two-Stage Quadratic Integer Programs with Stochastic Right-Hand Sides

Osman Y. Özaltın
Department of Industrial Engineering, University of Pittsburgh, Pittsburgh, PA 15261, USA,
oyo1@pitt.edu

Oleg Prokopyev
Department of Industrial Engineering, University of Pittsburgh, Pittsburgh, PA 15261, USA,
prokopyev@engr.pitt.edu

Andrew J. Schaefer
Department of Industrial Engineering, University of Pittsburgh, Pittsburgh, PA 15261, USA,
schaefer@ie.pitt.edu

Abstract

We consider two-stage quadratic integer programs with stochastic right-hand sides, and present an equivalent reformulation using value functions. We first derive some basic properties of value functions of quadratic integer programs. We then propose a two-phase solution approach. The first phase constructs the value functions of quadratic integer programs in both stages. The second phase solves the reformulation using a global branch-and-bound algorithm. We also consider a level-set approach to reduce the search space in the second phase. We show that our method can solve instances whose extensive forms are hundreds of orders of magnitude larger than the largest quadratic integer programming instances solved in the literature.

1 Introduction

We consider the following class of two-stage quadratic integer programs with stochastic right-hand sides:

$$(P1): \quad \max \quad \frac{1}{2}x^T\Lambda x + c^T x + \mathbb{E}_\omega \mathbf{Q}(x,\omega) \tag{1a}$$

$$\text{subject to} \quad x \in X, \tag{1b}$$

where $X = \{x \in \mathbb{Z}_+^{n_1} \mid Ax \leq b\}$ and,

$$\begin{aligned}
\mathbf{Q}(x,\omega) = \max \quad & \frac{1}{2} y^T \Gamma y + d^T y && \text{(2a)}\\
\text{subject to} \quad & Wy \leq h(\omega) - Tx, && \text{(2b)}\\
& y \in \mathbb{Z}_+^{n_2}. && \text{(2c)}
\end{aligned}$$

The random variable ω from probability space $(\Omega, \mathcal{F}, \mathcal{P})$ describes the realizations of the uncertain parameters, known as *scenarios*. The numbers of constraints and decision variables in stage i are m_i and n_i, for $i = 1, 2$. First-stage objective vector $c \in \mathbb{R}^{n_1}$, right-hand side vector $b \in \mathbb{R}^{m_1}$ and second-stage objective vector $d \in \mathbb{R}^{n_2}$ are known column vectors. First-stage constraint matrix $A \in \mathbb{R}^{m_1 \times n_1}$, technology matrix $T \in \mathbb{R}^{m_2 \times n_1}$, and recourse matrix $W \in \mathbb{R}^{m_2 \times n_2}$ are all deterministic. Furthermore, $\Lambda \in \mathbb{R}^{n_1 \times n_1}$ and $\Gamma \in \mathbb{R}^{n_2 \times n_2}$ are known, and possibly indefinite, symmetric matrices. The stochastic component consists of only $h(\omega) \in \mathbb{R}^{m_2}, \forall \omega \in \Omega$.

The extensive form formulation of $(P1)$ is given by

$$\begin{aligned}
\max \quad & \frac{1}{2} x^T \Lambda x + c^T x + \mathbb{E}_\omega \left[\frac{1}{2} y(\omega)^T \Gamma y(\omega) + d^T y(\omega) \right] && && \text{(3a)}\\
\text{subject to} \quad & x \in X, && && \text{(3b)}\\
& W y(\omega) \leq h(\omega) - Tx && \forall \omega \in \Omega, && \text{(3c)}\\
& y(\omega) \in \mathbb{Z}_+^{n_2} && \forall \omega \in \Omega. && \text{(3d)}
\end{aligned}$$

In this paper we make the following assumptions:

A1 The random variable ω follows a discrete distribution with finite support.

A2 The first-stage feasibility set $X = \{x \in \mathbb{Z}_+^{n_1} \mid Ax \leq b\}$ is nonempty and bounded.

A3 $\mathbf{Q}(x, \xi(\omega))$ is finite for all $x \in X$ and $\omega \in \Omega$.

A4 The first-stage constraint matrix A, technology matrix T, and recourse matrix W are all integral, i.e., $A \in \mathbb{Z}^{m_1 \times n_1}$, $T \in \mathbb{Z}^{m_2 \times n_1}$, $W \in \mathbb{Z}^{m_2 \times n_2}$.

Assumption **A1** is justified by Schultz [63], who showed that the optimal solution to any stochastic program with continuously distributed ω can be approximated within any desired accuracy using a discrete distribution. Assumption **A2** and integrality restrictions in the first stage ensure that X is a finite set. Assumption **A3** ensures that $\mathbf{Q}(x, \xi(\omega))$ is feasible for all $x \in X$ and $\omega \in \Omega$, i.e., relatively complete recourse [76]. Assumption **A4** is not too strong in a sense, as a rational matrix can be converted to an integral one. Much of the stochastic programming studies in the literature make assumptions similar to **A1**-**A3** [10, 20, 43, 65] and **A4** [43]. Without loss of generality, we also assume that $b \in \mathbb{Z}^{m_1}$ and $h(\omega) \in \mathbb{Z}^{m_2}, \forall \omega \in \Omega$, since A, T, and W are all integer matrices. Note that all of the undesirable properties of stochastic integer programs (SIPs), e.g., discontinuity and nonconvexity of $\mathbf{Q}(x, \xi(\omega))$, still exist in $(P1)$.

In this paper, we reformulate $(P1)$ and develop a two-phase solution approach. The key advantage to the reformulation is that it is relatively insensitive to the number of both first- and second-stage variables and scenarios. However, it is very sensitive to the number of feasible right-hand sides in both stages including the number of rows in the constraint matrix as well as the magnitude of feasible right-hand sides.

The remainder of this paper is organized as follows. In Section 2, we review the related literature including quadratic integer programming, stochastic integer programming, and parametric optimization. In Section 3, we first review basic properties of value functions of linear integer programs. We then identify conditions under which some of those results hold for value functions of quadratic integer programs and also derive some new properties. In Section 4, we follow a strategy similar to the one used in Kong et al. [43] and Ahmed et al. [10] for stochastic linear integer programs, and reformulate $(P1)$ using the value functions of quadratic integer programs (QIPs) in both stages. Since ω is discretely distributed, the reformulation results in a family of quadratic integer programs in the first and second stages parameterized over feasible right-hand sides. Section 5 considers the first phase of our solution procedure. In this section, we take advantage of the properties derived in Section 3, and propose four algorithms to construct the value functions of the first- and second-stage quadratic integer programs. One of these algorithms is designed for the diagonal case. The others work efficiently for different problem structures that depend on the number of nonlinear terms in the objective function. In Section 6, we present a global branch-and-bound algorithm and a level-set approach to optimize $(P1)$ over the set of feasible first-stage right-hand side vectors. In Section 7, we discuss the details of implementation and present results of our computational experiments. Methods described in this paper are able to solve two-stage quadratic integer programs with discretely distributed stochastic right-hand sides that have very few rows but many columns in both stages under a huge set of possible scenarios. We make concluding remarks and give future research directions in Section 8.

2 Literature Review

This work draws on and contributes to both quadratic integer programming and stochastic integer programming literatures. We review both areas separately as we are unaware of any study on stochastic quadratic integer programs (SQIPs).

2.1 Quadratic Integer Programming

Quadratic integer programs (QIPs) have been extensively studied; e.g., the quadratic assignment problem [45], the quadratic knapsack problem [26], and discrete version of the bilinear programming problem [4]. Pisinger [61] and Loiola et al. [49] present recent surveys on these problems.

Linearization is widely used for solving 0–1 QIPs [1, 2, 3, 18, 28, 57, 58, 75]. The original problem is transformed into an equivalent linear mixed-integer program via introducing new variables and additional constraints. The linearized problem can be tackled by a standard linear mixed-integer programming solver like CPLEX [36]. The major drawback of these methods is the substantial increase of the problem size and weakness of the LP relaxations [1, 2]. There are various branch-and-bound [5, 7, 12, 16, 25, 35, 50, 51, 59, 62, 73] and cutting plane [6, 13, 32, 33, 46, 47, 60] algorithms proposed for general and 0–1 QIPs. Most of the efficient solution methods rely on some simplifying assumptions, e.g., considering unconstrained 0–1 problems [35, 59], or convex and separable objective functions [16, 62].

The value functions of linear integer programs have been considered in [15, 38, 39, 40, 77]. However, the literature on the value function of QIPs is very sparse. Sensitivity of definite QIPs is studied in [22, 30]. Granot and Skorin-Kapov [30] extends some of the linear integer programming proximity results derived in Schrijver et al. [21] to QIPs. Bank and Hansel [14] investigates the stability of indefinite mixed-integer quadratic programs.

In terms of computational and theoretical results of parameterized QIPs, an early computational study by McBride and Yormark [52] considers a sequence of quadratic 0–1 problems parameterized over the right-hand-side of a single constraint. More recently, Dua et al. [23] develops a global optimization algorithm for the solution of a general class of non-convex mixed integer programs parameterized over the right-hand sides of a set of possibly nonlinear constraints. However, their proposed algorithmic approach utilizes basic convex under- and over-estimators within a generic branch-and-bound algorithm. Furthermore, computational studies are limited to instances with fewer than 10 variables.

Our approach is different from the algorithms proposed for QIPs in the literature since we consider the value function of a general QIP parameterized over a family of right-hand sides. We develop some basic properties of the value function of QIPs in Section 3.2 and contrast and compare them with known results for the value function of linear IPs reviewed in Section 3.1. Then based on the established properties we propose algorithms for finding the value function for particular classes of QIPs (see discussion in Section 5).

2.2 Stochastic Integer Programming

Stochastic integer programs have many applications, including supply chain network design [71], telecommunications [41, 70], server location [56], and dynamic capacity acquisition [8]. However, imposing integrality restrictions on second-stage variables increases the problem complexity significantly, since the expected recourse function becomes nonconvex and discontinuous in general [72]. Algorithms developed for solving general stochastic programs with integer recourse utilize cutting planes and/or branch-and-bound techniques in combination with decomposition methods that exploit block-separability of the underlying problem structure. Here we briefly review some relevant solution approaches for two-stage stochastic programs with integer recourse. We refer the reader to Klein Haneveld and van der Vlerk [31], and Schultz [64] for detailed surveys.

Laporte and Louveaux [44] develops a decomposition algorithm based on branch-and-bound techniques for solving stochastic programs with binary first-stage variables and complete mixed-integer recourse. Norkin et al. [54] proposes a stochastic branch-and-bound algorithm for minimizing the expected recourse function over a finite and discrete first-stage feasible set. Carøe and Tind [20] uses duality theory and generalizes the L-shaped method [74] for solving stochastic programs with integer recourse. No implementation details or computational results are provided in that paper.

Carøe and Schultz [19] employs a scenario decomposition method together with Lagrangian relaxation, and proposes a global branch-and-bound algorithm for solving stochastic programs with integer recourse. Sherali and Fraticelli [69] modifies Benders' decomposition method based on reformulation-linearization techniques to solve stochastic programs that have binary first-stage and mixed 0-1 second stage variables. Sen and Higle [66] introduces disjunctive decomposition method that involves set convexification for solving stochastic programs with mixed-integer recourse. Later, Sen and Sherali [67] proposes alternative decomposition methods in which the second-stage integer subproblems are solved using branch-and-cut methods. Many of disjunctive programming-based algorithms discussed above assume a fixed recourse matrix. Ntaimo [55] introduces a disjunctive decomposition algorithm for two-stage mixed $0-1$ stochastic programs with random recourse matrix. This approach generates cutting planes for one scenario and transforms them into cuts that are valid for other scenario instances. Schultz et al. [65] develops a finite algorithm for stochastic programs with discrete distributions and pure-integer second-stage variables. A Gröbner basis strategy is employed to exploit the common structure of integer scenario subproblems parameterized over their right-hand sides. As an approximation scheme, Kleywegt et al. [42] studies the sample average approximation method [68] for stochastic programs where the set of first-stage decisions is discrete and finite. Ahmed and Shapiro [9] extends this method for two-stage stochastic programs with integer recourse and show that the proposed scheme will produce an optimal solution to the true problem with probability approaching one exponentially fast as the sample size is increased.

Next, we describe the two papers that are most closely related to our work. Ahmed et al. [10] considers two-stage stochastic programs with discrete distributions, mixed-integer first-stage, and pure-integer second-stage problems. A variable transformation is applied to make the discontinuities of the expected recourse function orthogonal to the variable axes. This structure is exploited through a rectangular branching strategy. Then, a bounding strategy is employed to obtain the value function of the second-stage integer program in the absence of discontinuities. Finiteness of the method is established within a bounded search domain.

Kong et al. [43] considers a class of stochastic programs with stochastic right-hand sides, pure-integer first- and second-stage problems that have linear objective functions in both stages, i.e., $\Lambda = \Gamma = \mathbf{0}$ in $(P1)$. Similar to Ahmed et al. [10], their approach is based on an equivalent variable transformation that uses the value functions of both stages. Superadditive duality properties are exploited to characterize the value functions efficiently. In contrast to [10], they find the value functions of both stages in advance, which are then utilized within a global branch-and-bound algorithm, or an implicit exhaustive search procedure. The extensive forms of the stochastic linear integer programming instances that are solved in Kong et al. [43] are the largest ones reported in the literature. Our solution approach is related to that of Kong et al. [43]. However, we consider the more general problem of stochastic quadratic integer programs.

3 Basic Properties of the Value Functions

We first review basic properties of value functions of linear integer programs (IPs). We then identify conditions under which some of those properties hold for value functions of QIPs.

3.1 Properties of Linear IP Value Functions

Given $G \in \mathbb{Z}^{m\times n}$, and $\gamma \in \mathbb{Z}^n$, consider the following family of parameterized linear IPs:

$$(\text{PIP}): \quad \zeta(\beta) = \max\{\gamma^T x \mid x \in S(\beta)\}, \quad S(\beta) = \{x \in \mathbb{Z}^n_+ \mid Gx \leq \beta\} \quad \text{for } \beta \in \mathbb{R}^m.$$

The function, $\zeta(\cdot) : \mathbb{Z}^m_+ \mapsto \mathbb{Z}$, is called the *value function* of (PIP). We define $\widehat{opt}(\beta) = \text{argmax}\{\gamma^T x \mid Gx \leq \beta, x \in \mathbb{Z}^n_+\}$, that is, the set of optimal solutions to (PIP) for a given right-hand side β. Let $S_{LP}(\beta) = \{x \in \mathbb{R}^n_+ \mid Gx \leq \beta\}$ and define $\zeta_{LP}(\beta) = \max\{\gamma^T x \mid x \in S_{LP}(\beta)\}$, the linear relaxation of $\zeta(\beta)$.

Next we state some basic properties of $\zeta(\cdot)$. The proofs of these propositions can be found in [53].

Proposition 3.1 *$\zeta(\mathbf{0}) \in \{0, \infty\}$. If $\zeta(\mathbf{0}) = \infty$, then $\zeta(\beta) = \pm\infty$ for all $\beta \in \mathbb{R}^m$. If $\zeta(\mathbf{0}) = 0$, then $\zeta(\beta) < \infty$ for all $\beta \in \mathbb{R}^m$.*

Problems with $\zeta(\beta) = \pm\infty$ are basically feasibility problems. Hence, often it is assumed that $\zeta(\mathbf{0}) = 0$ and thus $\zeta(\beta) < \infty$ for all $\beta \in \mathbb{R}^m$.

Proposition 3.2 *Let g_j and γ_j be the j^{th} column of the constraint matrix G and the j^{th} coefficient of the objective function $\gamma^T x$ of (PIP), respectively. Then, $\zeta(g_j) \geq \gamma_j$ for $j = 1, \ldots, n$.*

Proposition 3.3 *The value function of (PIP) is nondecreasing over $\mathbb{Z}^m$.*

Proposition 3.4 *The value function of (PIP) is superadditive over $D = \{\beta \in \mathbb{Z}^m \mid S(\beta) \neq \emptyset\}$. That is, for all $\beta_1, \beta_2 \in D$, if $\beta_1 + \beta_2 \in D$, then*

$$\zeta(\beta_1) + \zeta(\beta_2) \leq \zeta(\beta_1 + \beta_2).$$

Gomory [29] showed strong duality for the group problem and pioneered superadditive duality. An explicit statement of superadditive duality first appeared in Johnson [37] in the context of cyclic group problem. The superadditivity of the value functions of pure IPs was extensively studied in Blair and Jeroslow [15], and Wolsey [77]. The treatment for other IPs was given in Johnson [39, 40]. For a summary of results see Johnson [38] and Nemhauser [53].

Proposition 3.5 *If $\hat{x} \in \widehat{opt}(\beta)$, then*

$$\zeta(Gx) = \gamma^T x \quad \text{and} \quad \zeta(Gx) + \zeta(\beta - Gx) = \zeta(Gx) + \zeta(G(\hat{x} - x)) = \zeta(\beta),$$

for all $x \in \mathbb{Z}_+^n$ such that $x \leq \hat{x}$.

Proposition 3.5 is referred to as *Integer Complementary Slackness (ICS)*. One major consequence of this result is that it allows us to obtain the value function for a subset of right-hand sides without actually solving the respective linear IPs.

Corollary 3.1 *If $\zeta(g_j) > \gamma_j$, then for all $\beta \in \mathbb{Z}^m$ and all $\hat{x} \in \widehat{opt}(\beta)$, $\hat{x}_j = 0$.*

3.2 Properties of Quadratic IP Value Functions

Given a symmetric matrix $Q \in \mathbb{Z}^{n \times n}$, column vectors $c \in \mathbb{Z}^n, \beta \in \mathbb{Z}^m$, and constraint matrix $G \in \mathbb{Z}^{m \times n}$, we consider the following family of parametric QIPs:

$$(PQIP): \quad z(\beta) = \max\left\{\frac{1}{2}x^T Q x + c^T x \mid x \in S(\beta)\right\}. \tag{4}$$

The function, $z(\cdot) : \mathbb{Z}_+^m \mapsto \mathbb{Z}$, is called the value function of $(PQIP)$. We define $opt(\beta) = \text{argmax}\left\{\frac{1}{2}x^T Q x + c^T x \mid x \in S(\beta)\right\}$; that is, the set of optimal solutions to $(PQIP)$ given a right-hand side β. We also define

$z_{QP}(\beta) = \max\left\{\frac{1}{2}x^TQx + c^Tx \mid x \in S_{LP}(\beta)\right\}$, the continuous relaxation of $z(\beta)$. In addition, let q_i be i^{th} column and q_{ij} be $(i,j)^{th}$ element of matrix Q, respectively. Next two results are similar to properties of $\zeta(\cdot)$ given in Propositions 3.2 and 3.3.

Proposition 3.6 $z(g_j) \geq c_j + \frac{1}{2}q_{jj}$.

Proposition 3.7 *$z(\beta)$ is nondecreasing over $\mathbb{Z}^m$.*

Unfortunately superadditivity does not hold for $z(\cdot)$ in general. Next we identify some sufficient conditions under which superadditivity holds.

Proposition 3.8 *If Q is nonnegative, then $z(\beta)$ is superadditive over $D = \{\beta \in \mathbb{Z}^m \mid S(\beta) \neq \emptyset\}$. That is, for all $\beta_1, \beta_2 \in D$, if $\beta_1 + \beta_2 \in D$, then*

$$z(\beta_1) + z(\beta_2) \leq z(\beta_1 + \beta_2).$$

Otherwise, if Q contains a negative element, then there exists a matrix G such that $z(\beta)$ is not superadditive.

Proof: Let $x_1 \in opt(\beta_1)$ and $x_2 \in opt(\beta_2)$, then $x_1 + x_2 \in S(\beta_1 + \beta_2)$, and

$$\begin{aligned} z(\beta_1+\beta_2) &\geq c^T(x_1+x_2) + \frac{1}{2}(x_1+x_2)^TQ(x_1+x_2) \\ &= c^Tx_1 + c^Tx_2 + \frac{1}{2}x_1^TQx_1 + \frac{1}{2}x_2^TQx_2 + x_1^TQx_2 \qquad (5) \\ &= z(\beta_1) + z(\beta_2) + x_1^TQx_2, \end{aligned}$$

which implies that $z(\beta_1+\beta_2) \geq z(\beta_1) + z(\beta_2)$ since $q_{ij} \geq 0 \ \forall i, j$.

Next suppose that $\exists\, i, j$ such that $q_{ij} < 0$. If $i = j$, i.e. $q_{ii} < 0$, consider the following feasible region:

$$S(\beta) = \left\{ x \in \mathbb{Z}^n_+ \;\middle|\; \sum_{k:\ k \neq i} x_k \leq \beta_1,\ x_i \leq \beta_2,\ -x_i \leq \beta_3 \right\}. \qquad (6)$$

Let $\beta = (0, 1, -1)^T$. Then, for any given c, we obtain $z(\beta) = c_i + \frac{1}{2}q_{ii}$ and

$$z(2\beta) = 2c_i + 2q_{ii} = z(\beta) + z(\beta) + q_{ii} < z(\beta) + z(\beta).$$

Otherwise, if $i \neq j$ and $q_{ij} < 0$. Consider the following feasible region:

$$S(\beta) = \left\{ x \in \mathbb{Z}_+^n \;\middle|\; \sum_{k:\ k \neq i, k \neq j} x_k \leq \beta_1,\ x_i \leq \beta_2,\ -x_i \leq \beta_3,\ x_j \leq \beta_4,\ -x_j \leq \beta_5 \right\}. \tag{7}$$

Let $\beta = (0, 1, -1, 0, 0)^T$ and $\beta' = (0, 0, 0, 1, -1)^T$. Then, for any given c, we obtain $z(\beta) = c_i + \frac{1}{2}q_{ii}$, $z(\beta') = c_j + \frac{1}{2}q_{jj}$ and

$$z(\beta + \beta') = c_i + c_j + \frac{1}{2}q_{ii} + \frac{1}{2}q_{jj} + q_{ij} = z(\beta) + z(\beta') + q_{ij} < z(\beta) + z(\beta').$$

Proposition 3.9 *For any right-hand side $\bar{\beta}$, $S(\bar{\beta}) \neq \emptyset$, let $u \in \mathbb{R}_+^n$ be such that $x \leq u$ $\forall x \in S_{LP}(\bar{\beta})$. Define*

$$m_i = \min_{x \in S_{LP}(\bar{\beta})} q_i^T x = \sum_{j=1}^n \min\{0, q_{ij}u_j\}, \qquad i = 1, \ldots, n;$$

$$M_i = \max_{x \in S_{LP}(\bar{\beta})} q_i^T x = \sum_{j=1}^n \max\{0, q_{ij}u_j\}, \qquad i = 1, \ldots, n.$$

If

$$\min_{x_1, x_2 \in S_{LP}(\bar{\beta})} \left[\sum_{i=1}^n \max\left\{ m_i x_{1i}, M_i x_{1i} + u_i q_i^T x_2 - u_i M_i \right\} \right] \geq 0,$$

then $z(\cdot)$ is superadditive over $\mathcal{D} = \left\{ \beta \in \mathbb{Z}^m \mid \beta \leq \bar{\beta} \text{ and } S(\beta) \neq \emptyset \right\}$.

Proof: Define

$$\Upsilon_i = \left\{ (x_i, y_i) \in \mathbb{R}^2 \mid 0 \leq x_i \leq u_i, m_i \leq y_i \leq M_i \right\}$$

and let $\Upsilon = \Upsilon_1 \times \ldots \times \Upsilon_n$. Al-Khayyal and Falk [11] showed that

$$\sum_{i=1}^n \max\{m_i x_i, M_i x_i + u_i y_i - u_i M_i\}, \tag{8}$$

is a convex envelope of $x^T y$ over $\Upsilon \supseteq S_{LP}(\bar{\beta})$. Substituting $x = x_1$ and $y = Qx_2$ into (8) gives that

$$\sum_{i=1}^n \max\left\{ m_i x_{1i}, M_i x_{1i} + u_i q_i^T x_2 - u_i M_i \right\},$$

is a convex envelope of $x_1^T Q x_2$ over Υ. Hence, if

$$\min_{x_1, x_2 \in S_{LP}(\bar{\beta})} \left[\sum_{i=1}^{n} \max \left\{ m_i x_{1i}, M_i x_{1i} + u_i q_i^T x_2 - u_i M_i \right\} \right] \geq 0,$$

then $x_1^T Q x_2 \geq 0$ over $S(\beta)$, $\forall \beta \leq \bar{\beta}$. As a result, from inequality (5):

$$z(\beta_1 + \beta_2) \geq z(\beta_1) + z(\beta_2) + x_1^T Q x_2 \geq z(\beta_1) + z(\beta_2),$$

for any $\beta_1, \beta_2 \leq \bar{\beta}$ and $x_1 \in \mathit{opt}(\beta_1)$, $x_2 \in \mathit{opt}(\beta_2)$.

Next we investigate whether an analogue of Proposition 3.1 holds for QIPs. Note that Assumptions **A2** and **A3** ensure that $z(\cdot)$ is finite for all β. We relax these two assumptions for the results directly related to the finiteness of $z(\cdot)$.

Remark 3.1 *There are instances of $(PQIP)$ such that $z(\mathbf{0}) \notin \{0, \infty\}$. Consider the following parametric QIP instance:*

$$z(\beta) = \max \left\{ 3x_1 - \frac{3}{2} x_1^2 + x_2 \;\middle|\; x_2 \leq \beta \text{ and } x \in \mathbb{Z}_+^2 \right\}.$$

Obviously, $z(\mathbf{0}) = \frac{3}{2} \notin \{0, \infty\}$ with $\hat{x} = (1, 0)^T$.

However, if $z(\cdot)$ is superadditive, then first two properties of Proposition 3.1 extend to QIPs.

Proposition 3.10 *If $z(\cdot)$ is superadditive, then $z(\mathbf{0}) \in \{0, \infty\}$. Moreover, if $z(\mathbf{0}) = \infty$, then $z(\beta) = \pm\infty$ for all $\beta \in \mathbb{R}^m$.*

Proof: Suppose $z(\mathbf{0}) < \infty$. If $z(\cdot)$ is superadditive, then $z(\mathbf{0}) \leq 0$ [53]. Note that $z(\mathbf{0}) \geq 0$ since $\mathbf{0} \in S(\mathbf{0})$, which implies that $z(\mathbf{0}) = 0 \in \{0, \infty\}$.

Now suppose $z(\mathbf{0}) = \infty$. If $S(\beta) = \emptyset$, then $z(\beta) = -\infty$. If $S(\beta) \neq \emptyset$, then from superadditivity $z(\mathbf{0}) + z(\beta) \leq z(\beta) \Rightarrow \infty \leq z(\beta)$, and the result follows.

Remark 3.2 *There are instances of $(PQIP)$ such that $z(\beta) = +\infty$ for some $\beta \in \mathbb{R}^m$ while $z(\mathbf{0}) = 0$, which implies that the last statement of Proposition 3.1 does not hold for $(PQIP)$. Consider the following parametric QIP instance:*

$$z(\beta) = \max \left\{ x_2 + x_1 x_2 \;\middle|\; x_2 - x_1 \leq \beta_1, x_2 \leq \beta_2, x \in \mathbb{Z}_+^2 \right\}.$$

Note that $z(\mathbf{0}) = 0$, and $\hat{v} = (1,0)^T \in opt(\mathbf{0})$. If $\beta = (0,1)^T$, then $z(\beta) = +\infty$, since $\hat{x} = (1,1)^T + t(1,0)^T \in S(\beta)$ for any $t \geq 0$. Note that $z(\cdot)$ is also superadditive since Q is nonnegative.

Next Proposition 3.11 provides a sufficient condition for the finiteness of the value function of QIPs. As compared to linear IPs (see Proposition 3.1), we have one more condition in addition to $z(\mathbf{0}) = 0$.

Lemma 3.1 *If $z(\mathbf{0}) = 0$, then $\forall t \in \mathbb{Z}_+$ and $\forall v \in S(\mathbf{0})$ we have:*

1. *$tc^T v + \frac{1}{2}t^2 v^T Q v \leq 0$;*
2. *$v^T Q v \leq 0$;*
3. *if $v^T Q v = 0$, then $c^T v \leq 0$.*

Proof: If $v \in S(\mathbf{0})$, then $tv \in S(\mathbf{0})$, $\forall t \in \mathbb{Z}_+$ and

$$c^T(tv) + \frac{1}{2}(tv)^T Q(tv) \leq z(\mathbf{0}) = 0, \tag{9}$$

which proves the first claim. Note that if $v^T Q v > 0$, then $c^T(tv) + \frac{1}{2}(tv)^T Q(tv) > 0$ as $t \to +\infty$ which contradicts $z(\mathbf{0}) = 0$. Hence $v^T Q v \leq 0$ for all $v \in S(\mathbf{0})$ and the second claim follows. Finally, if $v^T Q v = 0$, then inequality (9) reduces to $c^T(tv) \leq z(\mathbf{0}) = 0$ for $t \in \mathbb{Z}_+$ which proves the third claim.

Proposition 3.11 *If $z(\mathbf{0}) = 0$ and $x^T Q v \leq 0$, $\forall x \in S(\beta)$ and $\forall v \in S(\mathbf{0})$, then $z(\beta) < \infty$, $\forall \beta \in \mathbb{R}^m$.*

Proof: Since $z(\mathbf{0}) = 0$ the results of Lemma 3.1 are valid. If $x \in S(\beta)$, then $(x + tv) \in S(\beta)$, $\forall t \in \mathbb{Z}_+$ and $\forall v \in S(\mathbf{0})$. It follows that

$$\begin{aligned} c^T(x+tv) + \frac{1}{2}(x+tv)^T Q(x+tv) &= c^T x + c^T tv + \frac{1}{2}x^T Q x + \frac{1}{2}t^2 v^T Q v + t x^T Q v \\ &\leq c^T x + \frac{1}{2}x^T Q x + t x^T Q v \\ &\leq c^T x + \frac{1}{2}x^T Q x < \infty \text{ as } t \to +\infty, \end{aligned}$$

The first inequality follows from Lemma 3.1 and the second inequality holds since $x^T Q v \leq 0$, $\forall x \in S(\beta)$ and $\forall v \in S(\mathbf{0})$.

Note that Proposition 3.11 is somewhat analogous to Eaves' existence theorem [24] for continuous quadratic programs.

Remark 3.3 *The superadditivity of $z(\cdot)$ does not imply that $x^TQv \leq 0$. This can be verified by the instance considered in Remark 3.2, where $x^TQv = 1$ for $x = (1,1)^T$ and $v = (1,0)^T$.*

Remark 3.4 *If $x^TQv \leq 0$, $\forall x \in S(\beta)$ and $\forall v \in S(\mathbf{0})$, then $z(\cdot)$ is not necessarily superadditive. Consider the following instance of $(PQIP)$:*

$$z(\beta) = \max\left\{3x - x^2 \;\middle|\; x \leq \beta, x \in \mathbb{Z}^1_+\right\}.$$

$z(\cdot)$ is not superadditive since $z(1) = 2$, $z(2) = 2$, and $z(3) = 2$. Moreover, $x^TQv = 0$ for all x since $v = 0$ when $\beta = 0$.

As a result there is not a direct relation between the superadditivity of the value function $z(\cdot)$ and Proposition 3.11. Next we consider the extensions of Proposition 3.5 (Integer Complementary Slackness) for QIPs.

Proposition 3.12 *Let $\hat{x} \in opt(\beta)$. Then $\forall x \leq \hat{x}$,*

$$z(G\hat{x}) - x^TQ(\hat{x} - x) \leq z(Gx) + z(G(\hat{x} - x)) \leq z(Gx) + z(\beta - Gx). \quad (10)$$

Proof: The right inequality follows since $z(\cdot)$ is nondecreasing and $\hat{x} \in S(\beta)$. To show the left inequality, consider:

$$\begin{aligned} z(Gx) + z(G(\hat{x} - x)) &\geq c^Tx + \frac{1}{2}x^TQx + c^T(\hat{x} - x) + \frac{1}{2}(\hat{x} - x)^TQ(\hat{x} - x) \\ &= c^T\hat{x} + \frac{1}{2}x^TQx + \frac{1}{2}\hat{x}^TQ\hat{x} + \frac{1}{2}x^TQx - x^TQ\hat{x} \\ &= c^T\hat{x} + \frac{1}{2}\hat{x}^TQ\hat{x} - x^TQ(\hat{x} - x) = z(G\hat{x}) - x^TQ(\hat{x} - x). \end{aligned}$$

Remark 3.5 *Either bound in (10) can be tight with nonzero Q. Consider the following instance of $(PQIP)$:*

$$z(\beta) = \max\left\{-x^2 + 6x \;\middle|\; x \leq \beta, x \in \mathbb{Z}^1_+\right\}. \quad (11)$$

For $\beta = 3$, we have $\hat{x} = 3$ and $z(G\hat{x}) = z(3) = 9$. Let $x = 2 \leq \hat{x} = 3$. Then, $z(G(\hat{x} - x)) = z(1) = 5$, $z(Gx) = z(2) = 8$, and $z(G\hat{x}) - x^TQ(\hat{x} - x) = z(Gx) + z(G(\hat{x} - x)) = 13$, which implies that the left bound in (10) can be tight with nonzero Q. Note that $z(\beta - Gx) = z(G(\hat{x} - x)) = z(1)$ and the right bound can be tight as well.

Proposition 3.13 *Suppose $z(\cdot)$ is superadditive and let $\hat{x} \in opt(\beta)$. Then $\forall x \leq \hat{x}$*

$$z(Gx) \leq c^T x + \frac{1}{2} x^T Q x + x^T Q(\hat{x} - x). \tag{12}$$

Proof: Suppose (12) is not true for some $x \leq \hat{x}$. Then,

$$z(Gx) > c^T x + \frac{1}{2} x^T Q x + x^T Q(\hat{x} - x)$$

so that

$$\begin{aligned} z(Gx) + z(G(\hat{x} - x)) &> c^T x + \frac{1}{2} x^T Q x + x^T Q(\hat{x} - x) + z(G(\hat{x} - x)) \\ &\geq c^T x + \frac{1}{2} x^T Q x + x^T Q(\hat{x} - x) + c^T(\hat{x} - x) + \frac{1}{2}(\hat{x} - x)^T Q(\hat{x} - x) \\ &= c^T \hat{x} + \frac{1}{2} x^T Q x + x^T Q(\hat{x} - x) + \frac{1}{2}\hat{x}^T Q \hat{x} + \frac{1}{2} x^T Q x - x^T Q \hat{x} \\ &= c^T \hat{x} + \frac{1}{2}\hat{x}^T Q \hat{x} + x^T Q(\hat{x} - x) + x^T Q(x - \hat{x}) = z(G\hat{x}) \end{aligned}$$

which contradicts the superadditivity of $z(\cdot)$.

Remark 3.6 *The bound in Proposition 3.13 can be tight even when $x^T Q(\hat{x} - x) \neq 0$. Consider the following instance of $(PQIP)$:*

$$\max \left\{ x_1^2 + x_1 x_2 - x_1 + x_2 \;\middle|\; 2x_1 + x_2 \leq 4, x \in \mathbb{Z}_2^+ \right\}.$$

Clearly, $\hat{x} = (1, 2)^T \in opt(4)$ with an objective function value of $z(4) = 4$. Consider $x = (1, 1)^T \leq \hat{x} = (1, 2)^T$. Then, $Gx = 3$ and $(0, 3)^T \in opt(3)$ with an objective function value of $z(Gx) = z(3) = 3$. Moreover, $x^T Q(\hat{x} - x) = 1$, $c^T x + \frac{1}{2} x^T Q x = 2$, and $z(\cdot)$ is superadditive since Q is nonnegative. As a result,

$$c^T x + \frac{1}{2} x^T Q x + x^T Q(\hat{x} - x) = 3 = z(Gx).$$

Remark 3.7 *For linear IPs, $\zeta(\beta - Gx) = \zeta(G(\hat{x} - x))$ for all $x \leq \hat{x} \in \widehat{opt}(\beta)$. However, this property does not necessarily hold for QIPs even if the value function $z(\cdot)$ is superadditive. Consider the following QIP instance:*

$$\max \left\{ x_1 + x_2^2 + 2x_2 + 2x_1 x_2 \;\middle|\; x_1 + 3x_2 \leq 9, x_1 + x_2 \leq 4, x \in \mathbb{Z}_2^+ \right\}.$$

Clearly, $\hat{x} = (2,2)^T \in opt((9,4)^T)$ with an objective function value of $z((9,4)^T) = 18$ and $z(\cdot)$ is superadditive since Q is nonnegative. Let $x = (0,2)^T \leq \hat{x} = (2,2)^T$. Then, $Gx = (6,2)^T$ and $\beta - Gx = (3,2)^T$. Moreover, $G\hat{x} = (8,4)^T$ and $G(\hat{x}-x) = (2,2)^T$. We have $z(\beta - Gx) = z((3,2)^T) = 3$ (with a solution of $(0,1)^T$) and $z(G(\hat{x}-x)) = z((2,2)^T) = 2$ (with a solution of $(2,0)^T$). As a result, $z(\beta - Gx) > z(G(\hat{x}-x))$.

However, if $z(\cdot)$ is superadditive, we can find an upper bound on the difference between $z(\beta - Gx) - z(G(\hat{x}-x))$, which also provides us with some necessary optimality conditions as a corollary.

Proposition 3.14 *Suppose $z(\cdot)$ is superadditive and let $\hat{x} \in opt(\beta)$. Then $\forall x \leq \hat{x}$ we have*

$$0 \leq z(\beta - Gx) - z(G(\hat{x}-x)) \leq x^T Q(\hat{x}-x). \tag{13}$$

Proof: The left inequality follows from Proposition 3.7. To show the right inequality, since $z(\cdot)$ is superadditive we have:

$$\begin{aligned} z(\beta - Gx) - z(G(\hat{x}-x)) &\leq z(\beta) - z(Gx) - z(G(\hat{x}-x)) \\ &\leq c^T\hat{x} + \frac{1}{2}\hat{x}^T Q\hat{x} - c^T x - \frac{1}{2}x^T Qx \qquad (14) \\ &- c^T(\hat{x}-x) - \frac{1}{2}(\hat{x}-x)^T Q(\hat{x}-x) \\ &= \frac{1}{2}\hat{x}^T Q\hat{x} - \frac{1}{2}x^T Qx - \frac{1}{2}\hat{x}^T Q\hat{x} - \frac{1}{2}x^T Qx + x^T Q\hat{x} \qquad (15) \\ &= x^T Q(\hat{x}-x). \end{aligned}$$

Corollary 3.2 *Suppose $z(\cdot)$ is superadditive and let $\hat{x} \in opt(\beta)$. Then $\forall x \leq \hat{x}$,*

$$x^T Q(\hat{x}-x) \geq 0. \tag{16}$$

Our next result is an extension of Proposition 3.5 for QIPs.

Corollary 3.3 *Suppose $z(\cdot)$ is superadditive and let $\hat{x} \in opt(\beta)$. Then $\forall x \leq \hat{x}$*

$$c^T x + \frac{1}{2}x^T Qx \leq z(Gx) \leq c^T x + \frac{1}{2}x^T Qx + x^T Q(\hat{x}-x) \tag{17}$$

and

$$z(G\hat{x}) - x^T Q(\hat{x}-x) \leq z(Gx) + z(G(\hat{x}-x)) \leq z(Gx) + z(\beta - Gx) \leq z(G\hat{x}). \tag{18}$$

Observe that if $Q = \mathbf{0}$, then $z(\cdot)$ is superadditive and Corollary 3.3 simplifies to classical Integer Complementary Slackness for linear IPs.

The following corollaries allow us to get either exact values or perform some simple column/value elimination when matrix Q has a certain structure.

Corollary 3.4 *Suppose $z(\cdot)$ is superadditive and let $\hat{x} \in opt(\beta)$. If there exists j such that $\forall i \neq j$ $q_{ij} = q_{ji} = 0$, then*

$$z(\hat{x}_j g_j) = c_j \hat{x}_j + \frac{1}{2} q_{jj} \hat{x}_j^2.$$

Proof: Consider vector $x = (0, \ldots, 0, \hat{x}_j, 0, \ldots, 0)^T$, with $x_j = \hat{x}_j$ and $x_i = 0$ for all $i \neq j$. Then $x \leq \hat{x}$, $Gx = \hat{x}_j g_j$, and $x^T Q(\hat{x} - x) = 0$, and the result follows from Corollary 3.3.

Corollary 3.5 *Let $Q = diag(q_{11}, \ldots, q_{nn}) \succeq 0$ and $\hat{x} \in opt(\beta)$. Then for all x such that $x_i = 0$ or $x_i = \hat{x}_i$ $\forall i$ we have*

$$z(Gx) = c^T x + \frac{1}{2} x^T Q x. \tag{19}$$

Proof: If x is such that $x_i = 0$ or $x_i = \hat{x}_i$ $\forall i$, then $x \leq \hat{x}$ and $x^T Q(\hat{x} - x) = 0$, since Q is diagonal. This implies the necessary result from Corollary 3.3.

Corollary 3.6 *Suppose is $z(\cdot)$ superadditive. If there exists j such that $\forall i$ $q_{ij} = q_{ji} = 0$ and $z(g_j) > c_j$, then $\hat{x}_j = 0$, $\forall \beta \in \mathbb{Z}^m$ and $\forall \hat{x} \in opt(\beta)$.*

Proof: Consider $\beta \in \mathbb{Z}^m$ and $\hat{x} \in opt(\beta)$. If $\hat{x}_j \geq 1$, then from Proposition 3.13 we get

$$z(g_j) \leq c^T e_j + \frac{1}{2} e_j^T Q e_j + e_j^T Q(\hat{x} - e_j) = c_j,$$

which contradicts $z(g_j) > c_j$.

Corollary 3.6 is analogous to Corollary 3.1. It holds when $z(\cdot)$ is superadditive and the respective variable does not appear in the nonlinear part of the objective function. Our next result is a slight generalization of Corollary 3.6 for diagonal $Q \succeq 0$.

Corollary 3.7 *Let $Q = diag(q_{11}, \ldots, q_{nn}) \succeq 0$. If $\exists j, k, h \in \mathbb{Z}_+^1$ such that $k \geq h \geq 1$ and $z(hg_j) > hc_j + h(k - \frac{1}{2}h)q_{jj}$, then $\hat{x}_j \notin [h, k]$, $\forall \beta \in \mathbb{Z}^m$ and $\forall \hat{x} \in opt(\beta)$.*

Proof: Note that $z(\cdot)$ is superadditive from Proposition 3.8. Consider $\beta \in \mathbb{Z}^m$ and $\hat{x} \in opt(\beta)$. Let $k \geq 1$ and $h \geq k$. If $\hat{x}_j \in [h, k]$, then from Proposition 3.13 we have

$$z(hg_j) \leq hc^T e_j + \frac{1}{2}h^2 e_j^T Q e_j + he_j^T Q(\hat{x} - he_j) = hc_j + h(\hat{x}_j - \frac{1}{2}h)q_{jj}.$$

4 Value Function Reformulation

In this section, we demonstrate that the reformulation strategy proposed for two-stage linear IPs with stochastic right-hand sides in Kong et al. [43] and Ahmed et al. [10] extends to two-stage QIPs with stochastic right-hand sides. We reformulate $(P1)$ using the value functions of QIPs in both stages. Let $\mathbf{B}^1$ denote the set of vectors $\beta_1 \in \mathbb{R}^{m_2}$ such that there exists $x \in X$ satisfying $\beta_1 = Tx$, i.e., $\mathbf{B}^1 = \{\beta_1 \in \mathbb{R}^{m_2} \mid \exists x \in X, \beta_1 = Tx\}$, where $X \subseteq \mathbb{Z}_+^{n_1}$ is the first-stage feasibility set. Furthermore, let $\mathbf{B}^2$ denote the set of vectors $\beta_2 \in \mathbb{R}^{m_2}$ such that there exists $\beta_1 \in \mathbf{B}^1$ and $\omega \in \Omega$ satisfying $\beta_2 = h(\omega) - \beta_1$, i.e., $\mathbf{B}^2 = \cup_{\beta_1 \in \mathbf{B}^1} \cup_{\omega \in \Omega} \{h(\omega) - \beta_1\}$. Note that all vectors in $\mathbf{B}^1$ are integral since $T \in \mathbb{Z}^{m_2 \times n_1}$. Together with the condition $h(\omega) \in \mathbb{Z}^{m_2}$ $\forall \omega \in \Omega$, all vectors in $\mathbf{B}^2$ are also integral.

For any $\beta_1 \in \mathbb{Z}^{m_2}$, we give the first-stage value function as:

$$\psi(\beta_1) = \max\left\{\frac{1}{2}x^T \Lambda x + c^T x \mid x \in S_1(\beta_1)\right\}, \quad S_1(\beta_1) = \{x \in X \mid Tx \leq \beta_1\}. \tag{20}$$

Note that the condition $Tx = \beta_1$ in the definition of $\mathbf{B}^1$ is replaced by $Tx \leq \beta_1$ in (20). This is justified by nondecreasing property of the value function in Proposition 3.7. Using the inequality instead of the equality allows us to apply properties considered in Section 3.2 for algorithmic developments.

Next, for any $\beta_2 \in \mathbb{Z}^{m_2}$, we write the second-stage value function as:

$$\phi(\beta_2) = \max\left\{\frac{1}{2}y^T \Gamma y + d^T y \mid y \in S_2(\beta_2)\right\}, \ S_2(\beta_2) = \left\{y \in \mathbb{Z}_+^{n_2} \mid Wy \leq \beta_2\right\}. \tag{21}$$

Then we reformulate $(P1)$ as:

$$(P2): \quad \max\left\{\psi(\beta) + \mathbb{E}_\omega \phi(h(\omega) - \beta) \mid \beta \in \mathbf{B}^1\right\}. \tag{22}$$

The variables β in $(P2)$ are known as the *tender variables* [10, 43]. Instead of searching X, we search the space of tender variables to obtain a global optimum.

The following result establishes the correspondence between the optimal solutions of (P1) and (P2).

Theorem 4.1 *Let β^* be an optimal solution to (P2). Then, $\hat{x} \in opt(\beta^*)$ is an optimal solution to (P1). Furthermore, the optimal objective values of the two problems are equal.*

The proof of Theorem 4.1 is similar to the one for Theorem 3.2 of Ahmed et al. [10] and is omitted. Finding the value functions in both stages is critical for solving (P2). In the next section, we present algorithms for finding the value functions $\psi(\cdot)$ and $\phi(\cdot)$. For the ease of exposition we continue using $z(\cdot)$ as a generic value function.

5 Constructing the Value Function of a Parameterized Quadratic IP

As we discussed in Section 2.1 very little is known about how to compute the value functions of QIPs efficiently. On the other hand, there are algorithms proposed in the literature for calculating the value functions of linear IPs, see, e.g., algorithms in [43]. Motivated by some of the ideas used in the linear case, we develop four algorithms to construct the value functions of QIPs.

Assumptions **A1** and **A2** ensure the finiteness of $\mathbf{B}^1$ and $\mathbf{B}^2$. Therefore, we consider $(PQIP)$ parameterized over a finite set of right-hand sides $\beta \in \mathbf{B}$. Under assumptions **A2** and **A3**, the value functions in both stages $\psi(\beta_1)$ and $\phi(\beta_2)$, are finite for all $\beta_1 \in \mathbf{B}^1$ and $\beta_2 \in \mathbf{B}^2$, respectively. Therefore, we assume that $z(\beta)$ is finite for all $\beta \in \mathbf{B}$.

For each of the proposed algorithms, we assume that either the objective function, or the constraint matrix has a particular structure. Our first algorithm is based on the bounds derived in Section 3 for superadditive QIP value functions. The next three algorithms are designed for problems with nonnegative constraint matrix G. The second algorithm applies to problems with diagonal $Q \succeq 0$, where as the remaining two assumes that the objective function can be decomposed into sum of a small number of products of linear functions.

5.1 An Exact Algorithm Based on Superadditivity

In this section we assume that $z(\cdot)$ is superadditive. Let $l(\cdot)$ and $u(\cdot)$ be lower and upper bounds of $z(\cdot)$, respectively. We maintain $l(\beta) \leq z(\beta) \leq u(\beta)$ for all $\beta \in \mathbf{B}$ throughout the algorithm. Once $l(\beta) = u(\beta)$, $z(\beta)$ is known and the algorithm terminates when $z(\beta)$ is determined for all $\beta \in \mathbf{B}$. In addition to the bounds derived in Section 3, we utilize the following properties of the value functions.

Lemma 5.1 *Suppose $z(\cdot)$ is superadditive and let $\hat{x} \in opt(\beta)$. Then, $z(\bar{\beta}) = 0$ for all $\bar{\beta}$ such that $0 \leq \bar{\beta} \leq \beta - G\hat{x}$.*

Lemma 5.2 *Suppose $z(\cdot)$ is nondecreasing and let $\hat{x} \in opt(\beta)$. Then, $z(\bar{\beta}) = z(\beta)$ for all $\bar{\beta}$ such that $G\hat{x} \leq \bar{\beta} \leq \beta$.*

At each iteration, we update $l(\beta)$ and $u(\beta)$ for some $\beta \in \mathbf{B}$ by performing the following two basic operations:

1. Solve a quadratic integer program exactly for a given right-hand side $\beta \in \mathbf{B}$ (e.g., using a simple dynamic programming based algorithm, or any QIP solver) and obtain primal optimal solution $\hat{x}$.
2. Given $\hat{x}$, update the lower and upper bounds for a subset right-hand sides in $\mathbf{B}$ utilizing:
    i. properties of value functions given by Lemmas 5.1 and 5.2;
    ii. nondecreasing and superadditivity properties of $z(\cdot)$ (Propositions 3.7 and 3.8), bounds from Proposition 3.13, and some basic feasibility arguments (see details below).

Algorithm 1. *The Exact-Superadditive Algorithm.*

Step 0: Initialize the lower bound $l^0(\beta) = -\infty$ for all $\beta \in \mathbf{B}$. For $j = 1, \ldots, n$, if $g_j \in \mathbf{B}$, set $l^0(g_j) = \frac{1}{2}q_{jj} + c_j$. Without loss of generality, we assume that there are no duplicate columns. Initialize the upper bound $u^0(\beta) = +\infty$ for all $\beta \in \mathbf{B}$. Initialize $\mathcal{L}^k = \emptyset$ and set $k \leftarrow 1$.

Step 1: Set $l^k(\beta) \leftarrow l^{k-1}(\beta)$ and $u^k(\beta) \leftarrow u^{k-1}(\beta)$ for all $\beta \in \mathbf{B}$. Select $\beta^k \in \mathbf{B} \setminus \mathcal{L}^k$. Solve the quadratic integer program with right-hand side β^k to obtain an optimal solution $\hat{x}^k$.

(1a) For all $\beta \in \mathbf{B} \setminus \mathcal{L}^k$ such that $G\hat{x}^k \leq \beta \leq \beta^k$, set $l^k(\beta) = u^k(\beta) = c^T\hat{x}^k + \frac{1}{2}\hat{x}^{kT}Q\hat{x}^k$, and $\mathcal{L}^k \leftarrow \mathcal{L}^k \cup \{\beta\}$,

(1b) For all $\beta \in \mathbf{B} \setminus \mathcal{L}^k$ such that $0 \leq \beta \leq \beta^k - G\hat{x}^k$, set $l^k(\beta) = u^k(\beta) = 0$, and $\mathcal{L}^k \leftarrow \mathcal{L}^k \cup \{\beta\}$.

(1c) For all $\beta \in \mathbf{B}\setminus\mathcal{L}^k$ such that $\beta \geq \beta^k$, set $l^k(\beta) \leftarrow \max\left\{l^k(\beta), l^k(\beta^k)\right\}$. If $\beta - \beta^k \in \mathbf{B} \setminus \mathcal{L}^k, l^k(\beta) \leftarrow \max\left\{l^k(\beta), l^k(\beta^k) + l^k(\beta - \beta^k)\right\}$.

(1d) For all $\beta \in \mathbf{B}\setminus\mathcal{L}^k$ such that $\beta \leq \beta^k$, set $u^k(\beta) \leftarrow \min\left\{u^k(\beta), u^k(\beta^k)\right\}$. If $\beta^k - \beta \in \mathbf{B} \setminus \mathcal{L}^k, u^k(\beta) \leftarrow \min\left\{u^k(\beta), u^k(\beta^k) - l^k(\beta^k - \beta)\right\}$.

(1e) For all $\beta \in \mathbf{B}\setminus\mathcal{L}^k$, if $\beta+\beta^k \in \mathbf{B}\setminus\mathcal{L}^k$, $u^k(\beta) \leftarrow \min\left\{u^k(\beta), u^k(\beta + \beta^k) - l^k(\beta^k)\right\}$.

Step 2: Select all $x \in \mathbb{Z}^n_+$ such that $x \leq \hat{x}^k$. If $Gx \in \mathbf{B}$, set

(2a) $l^k(Gx) \leftarrow \max\left\{l^k(Gx), c^Tx + \frac{1}{2}x^TQx\right\}$,

(2b) $u^k(Gx) \leftarrow \min\left\{u^k(Gx), c^Tx + \frac{1}{2}x^TQx + x^TQ(\hat{x} - x)\right\}$.

If $\beta^k - Gx \in \mathbf{B}$, set

(2c) $l^k(\beta^k - Gx) \leftarrow \max\left\{l^k(\beta^k - Gx), l^k(G\hat{x}^k) - x^TQ(\hat{x}^k - x) - c^Tx - \frac{1}{2}x^TQx\right\}$,

(2d) $u^k(\beta^k - Gx) \leftarrow \min\left\{u^k(\beta^k - Gx), u^k(G\hat{x}^k) - c^Tx - \frac{1}{2}x^TQx\right\}$.

Step 3: If $l^k(\beta) = u^k(\beta)$ for all $\beta \in \mathbf{B}$, terminate with solution $z(\cdot) = l^k(\cdot) = u^k(\cdot)$; otherwise, set $k \leftarrow k + 1$ and go to Step 1.

Lemma 5.3 *At any iteration k in the Exact-Superadditive Algorithm, $l^k(\beta) \leq z(\beta) \leq u^k(\beta)$ for all $\beta \in \mathbf{B}$.*

Proof: The lower bounds in Step 0 follows from Proposition 3.6. Suppose that at iteration $k - 1 \geq 0$, Lemma 5.3 holds. Consider iteration k.

- Steps (1a) and (1b) are correct due to Lemmas 5.1 and 5.2, respectively.
- Steps (1c)-(1e) are due to the nondecreasing and superadditivity properties of $z(\cdot)$, respectively.
- Step (2a) holds since $x \in S(Gx)$.
- Step (2b) follows from Proposition 3.13.

- Step (2c) holds since $(\hat{x}^k - x) \in S(\beta^k - Gx)$ and $z(\cdot)$ is nondecreasing. Specifically,

$$z(\beta - Gx) \geq z(G(\hat{x} - x)) \geq c^T(\hat{x} - x) + \frac{1}{2}(\hat{x} - x)^T Q(\hat{x} - x), \quad (23)$$

  which after simple manipulations results in the necessary inequality.

- Step (2d) holds since $x \in S(Gx)$ and $z(\cdot)$ is superadditive. Specifically,

$$z(\beta - Gx) \leq z(G\hat{x}) - z(Gx) \leq z(G\hat{x}) - c^T x - \frac{1}{2}x^T Qx,$$

which completes the proof.

Proposition 5.1 *The Exact-Superadditive Algorithm terminates finitely with optimal solutions to $z(\cdot)$ for all $\beta \in \mathbf{B}$.*

Proof: Consider any iteration $k \geq 1$. After Step 1 of the iteration, there exists at least one $\beta \in \mathbf{B}$ such that $l^k(\beta) = u^k(\beta) = z(\beta)$ while $l^{k-1}(\beta) \neq z(\beta)$ or $u^{k-1}(\beta) \neq z(\beta)$. By Lemma 5.3, $l^k(\beta) \leq z(\beta) \leq u^k(\beta)$ for all $\beta \in \mathbf{B}$ at any iteration k, and since $\mathbf{B}$ is finite, the result follows.

5.2 A DP-based Algorithm for Diagonal $Q \succeq 0$

In this section we assume that $Q = diag(q_{11}, \ldots, q_{nn}) \succeq 0$, i.e., Q is a diagonal matrix and all q_{ii}'s are nonnegative. Therefore, by Proposition 3.8, $z(\cdot)$ is superadditive. We also assume that G is a nonnegative matrix. Since $\mathbf{B}$ is finite, there exists a nonnegative hyper-rectangle $\mathcal{B}$ rooted at the origin that contains $\mathbf{B}$. Let $b = (b_1, \ldots, b_m)$ denote the largest vector in $\mathcal{B}$ componentwise. For simplicity of exposition, we let $\mathbf{B} = \mathcal{B} \cap \mathbb{Z}^m_+$, where $\mathcal{B} = \{[0, b_1] \times [0, b_2] \times \ldots \times [0, b_m]\}$. We use the same notation in the rest of the algorithms presented in Section 5.

Next we present some results that are used later in this paper. Let B_j denote the set containing all $\beta \in \mathbf{B}$ such that $\beta \geq g_j$.

Proposition 5.2 *For all $\beta \in \mathbf{B} \setminus \cup_{j=1}^n B_j$, $\zeta(\beta) = 0$ and $z(\beta) = 0$.*

The following result is due to Gilmore and Gomory [27] for linear IPs:

Theorem 5.1 *[27]. For any $\beta \in \cup_{j=1}^{n} B_j$,*

$$\zeta(\beta) = \max\{0, \gamma_j + \zeta(\beta - g_j) \mid g_j \in \mathbf{B}, j = 1, \ldots, n\}. \tag{24}$$

This theorem can be easily extended to $(PQIP)$ with diagonal $Q \succeq 0$ as follows:

Lemma 5.4 *For any $\beta \in \cup_{j=1}^{n} B_j$, let*

$$\begin{aligned} z(\beta) = \max_{\mu,j}\{0, c_j\mu + \frac{1}{2}q_{jj}\mu^2 + z(\beta - \mu g_j) \mid \mu \in \mathbb{Z}_{++}, \\ \beta - \mu g_j \geq \mathbf{0},\ g_j \in \mathbf{B}, j = 1, \ldots, n\}. \end{aligned} \tag{25}$$

In contrast to the exact superadditive algorithm, the following method only defines $l(\cdot)$ and does not need to solve any quadratic integer program. The key idea is to update $l(\cdot)$ utilizing the superadditive property of $z(\cdot)$. This approach is motivated by the DP-based algorithm for finding the value function of linear IPs proposed in [43]. The major difference of our algorithm is the initialization of lower bounds in Step 0, which follows from Lemma 5.4.

Algorithm 2. *The Diagonal-Q Algorithm.*

Step 0: Initialize the lower bound $l^0(\beta) = 0$ for all $\beta \in \mathbf{B}$. For $j = 1, \ldots, n$, if $g_j \in \mathbf{B}$, $l^0(\mu g_j) \leftarrow \max\{l^0(\mu g_j), c_j\mu + \frac{1}{2}q_{jj}\mu^2\}$, $\forall \mu$ such that $b - \mu g_j \geq 0, \mu \in \mathbb{Z}_+$. Insert μg_j into a vector list $\mathcal{L}$. Set $l^1(\beta) = l^0(\beta)$ for all $\beta \in \mathbf{B}$. Set $k \leftarrow 1$.

Step 1: Denote the k^{th} vector in $\mathcal{L}$ by β^k and the i^{th} element of a vector β by β_i. Let $\beta = \beta^k$. Update all vectors β' such that $\beta' \in \mathbf{B}$ and $\beta' \geq \beta^k$ with the following lexicographic order:

(1a) Set $\beta_1 \leftarrow \beta_1 + 1$ and $l^k(\beta) \leftarrow \max\{l^k(\beta), l^k(\beta^k) + l^k(\beta - \beta^k)\}$.

(1b) If $\beta_1 \geq b_1$, go to Step (1c); otherwise, go to Step (1a).

(1c) If for all $i = 1, \ldots, m$, $\beta_i \geq b_i$, go to Step 2. Otherwise, let $s = \min\{i : \beta_i < b_i\}$. Set $\beta_i \leftarrow \beta_i^k$ for $i = 1, \ldots, s-1$. Set $\beta_s \leftarrow \beta_s + 1$ and go to Step (1a).

Step 2: If $k = |\mathcal{L}|$, terminate with solution $z(\cdot) = l^k(\cdot)$. Otherwise, put $l^{k+1}(\beta) \leftarrow l^k(\beta)$ for all $\beta \in \mathbf{B}$, set $k \leftarrow k + 1$ and go to Step 1.

Next we provide the proof of correctness and finite termination of the outlined procedure, which basically follows the proof of the respective result in [43] for the linear case subject to changes in Step 0 and the results of Lemma 5.4. Let $\mu_{\max}$ denote the maximum feasible scalar value that any variable can take in a solution. Note that $\mu_{\max}$ is finite since G is nonnegative and $\mathbf{B}$ is finite.

Proposition 5.3 *The Diagonal-Q Algorithm terminates with optimal solutions to $z(\cdot)$ in at most $n\mu_{\max}$ iterations.*

Proof: For any $\beta \in \mathbf{B} \setminus \cup_{j=1}^{n} B_j$, we initialize $l^0(\beta) = 0$ in Step 0 and do not update them afterward. By Proposition 5.2, $z(\beta) = 0$ for $\beta \in \mathbf{B} \setminus \cup_{j=1}^{n} B_j$. Assume that the algorithm terminates at iteration $k^* = |\mathcal{L}|$. Then $l^{k^*}(\beta) = z(\beta)$ for all $\beta \in \mathbf{B} \setminus \cup_{j=1}^{n} B_j$. Suppose there exists $\beta \in \cup_{j=1}^{n} B_j$ such that $l^{k^*}(\beta) \neq z(\beta)$, and for all $\beta' \leq \beta$, $\beta' \in \cup_{j=1}^{n} B_j$, we have $l^{k^*}(\beta') = z(\beta')$. Then clearly $l^{k^*}(\beta) < z(\beta)$ by Lemma 5.3. It follows that there exists a $j^* \in \{1, \ldots, n\}$ and $\mu_* \geq 1$ such that $l^{k^*}(\beta) < c_{j^*}\mu_* + \frac{1}{2} q_{j^*j^*}\mu_*^2 + z(\beta - g_{j^*}\mu_*)$ by Lemma 5.4. Since $l^{k^*}(g_{j^*}\mu_*) \geq c_{j^*}\mu_* + \frac{1}{2} q_{j^*j^*}\mu_*^2$ and $l^{k^*}(\beta - g_{j^*}\mu_*) = z(\beta - g_{j^*}\mu_*)$, it follows that $l^{k^*}(g_{j^*}\mu_*) + l^{k^*}(\beta - g_{j^*}\mu_*) \geq c_{j^*}\mu_* + \frac{1}{2} q_{j^*j^*}\mu_*^2 + z(\beta - g_{j^*}\mu_*) > l^{k^*}(\beta)$, which contradicts the superaddivity of $l^{k^*}(\cdot)$. Hence, $l^{k^*}(\beta) = z(\beta)$ for all $\beta \in \cup_{j=1}^{n} B_j$. The result follows since $k^* = |\mathcal{L}| \leq n\mu_{\max}$.

Proposition 5.4 *The overall running time of the Diagonal-Q Algorithm is $O(n\mu_{\max}|\mathbf{B}|)$.*

Proof: Step 0 requires $O(n\mu_{\max})$ calculations. Step 1 of the algorithm requires at most $O(|\mathbf{B}|)$ calculations. Since Step 1 is executed at most $n\mu_{\max}$ times, the overall running time of the algorithm is $O(n\mu_{\max}|\mathbf{B}|)$.

We note that the worst case estimation of the overall running time of the most efficient algorithm for the linear case in Kong et al. [43] is $O(n|\mathbf{B}|)$. Since $\mu_{\max} << |B|$, the worst case running time estimation of the *Diagonal-Q Algorithm* is almost as good as that of the most efficient algorithm of Kong et al. [43].

5.3 An Iterative Fixing Algorithm for Low-Rank Q

For some $\ell \leq n$, the quadratic objective function of $(PQIP)$ can be represented as:

$$\frac{1}{2} x^T Q x + c^T x = (\chi_1^T x)(\sigma_1^T x) + \ldots + (\chi_\ell^T x)(\sigma_\ell^T x) + c^T x, \qquad (26)$$

where χ_i and σ_i are vectors in $\mathbb{R}^n$ for $i = 1, \ldots, \ell$. In this section, we are interested in classes of (26) with small $\ell << n$ and either all χ's or all σ's are nonnegative integer vectors.

To motivate the representation given in (26), note that any quadratic function can be written as follows:

$$\frac{1}{2}x^TQx + c^Tx = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} q_{ij}x_ix_j + c^Tx = \frac{1}{2}\sum_{i=1}^{n}\left(x_i \cdot \sum_{j=1}^{n} q_{ij}x_j\right) + c^Tx. \tag{27}$$

Suppose that for every nonlinear term x_ix_j in (27) either $i = 1, \ldots, \ell$, or $j = 1, \ldots, \ell$. Then due to symmetry of Q, quadratic function in (27) simplifies to

$$\frac{1}{2}x^TQx + c^Tx = \sum_{i=1}^{\ell}\left(x_i \cdot \sum_{j=1}^{n} q_{ij}x_j\right) + c^Tx, \tag{28}$$

where $\chi_1 = (1, 0, \ldots, 0)^T, \ldots, \chi_\ell = (0, \ldots, 0, 1, 0, \ldots, 0)^T$, and $\sigma_1 = (q_{11}, \ldots, q_{1n})^T$, $\ldots$, $\sigma_\ell = (q_{\ell 1}, \ldots, q_{\ell n})^T$. That is, columns and rows of Q can be rearranged in a such way that only first ℓ rows and ℓ columns may contain nonzero elements for some small ℓ, while the rest of Q is a zero submatrix.

Another motivation is that any symmetric matrix Q can be decomposed into

$$Q = U\ diag(\lambda_1, \ldots, \lambda_n)\ U^T, \tag{29}$$

where U is the matrix of eigenvectors u_1, $\ldots$, u_n (stored as columns) and $\lambda_1, \ldots, \lambda_n$ are eigenvalues of Q, respectively. If only ℓ eigenvalues of Q are nonzero, which for small ℓ corresponds to a low-rank matrix Q, then we have

$$\frac{1}{2}x^TQx + c^Tx = \lambda_1(u_1^Tx)^2 + \ldots + \lambda_\ell(u_\ell^Tx)^2 + c^Tx, \tag{30}$$

and we can set $\chi_1 = \lambda_1 u_1$, $\ldots$, $\chi_\ell = \lambda_\ell u_\ell$, and $\sigma_1 = u_1$, $\ldots$, $\sigma_\ell = u_\ell$.

Let $\mathcal{H}$ and Σ be $\ell \times n$ matrices with rows composed by $(\chi_i)^T$'s and $(\sigma_i)^T$'s, $i = 1, \ldots, \ell$, respectively. In this section, we assume that $\mathcal{H} \in \mathbb{Z}_+^{\ell \times n}$, and $\ell << n$. We present an algorithm for finding the value function of $(PQIP)$. Note that we do not require $z(\cdot)$ to be superadditive.

For $y \in \mathbb{Z}_+^\ell$, we define problem $\mathbf{P}^y(\beta)$ by:

$$z^y(\beta) = \max_{x \in \mathbb{Z}_+^n}\left\{ y^T\Sigma x + c^Tx \;\middle|\; Gx \le \beta,\ \mathcal{H}x = y \right\}. \tag{31}$$

Moreover, we formulate an auxiliary problem where the equality constraint of $\mathbf{P}^y(\beta)$ is replaced with an inequality. For $\delta, y \in \mathbb{Z}_+^\ell$, we define problem $\mathbf{P}^{\delta,y}(\beta)$ by:

$$z^{\delta,y}(\beta) = \max_{x \in \mathbb{Z}_+^n} \left\{ \delta^T \mathcal{H}x + y^T \Sigma x + c^T x \;\middle|\; Gx \leq \beta,\ \mathcal{H}x \leq y \right\}. \tag{32}$$

Let $\delta \in \mathbb{Z}_+^\ell$ be such that $\frac{1}{2}\delta^T e_i > \max_x \left\{ |y^T \Sigma x + c^T x| \;\middle|\; Gx \leq b,\ \mathcal{H}x \leq y \right\}$ for all $i = 1 \ldots \ell$. Then the following results hold.

Lemma 5.5 *Let $\hat{x}$ be an optimal solution to $\mathbf{P}^{\delta,y}(\beta)$. Then $z^{\delta,y}(\beta) \geq \delta^T y - \frac{1}{2}\min_i \delta^T e_i$ if and only if $y = \mathcal{H}\hat{x}$.*

Proof:
"$\Leftarrow$" If $y = \mathcal{H}\hat{x}$, then $z^{\delta,y}(\beta) = \delta^T y + y^T \Sigma \hat{x} + c^T \hat{x}$. This value is at least as large as $\delta^T y - \frac{1}{2}\min_i \delta^T e_i$ because $\frac{1}{2}\delta^T e_i + y^T \Sigma x + c^T x > 0$ for all i and for all feasible x by definition of δ.
"$\Rightarrow$" If $z^{\delta,y}(\beta) \geq \delta^T y - \frac{1}{2}\min_i \delta^T e_i$, then

$$\begin{aligned} &\delta^T \mathcal{H}\hat{x} + y^T \Sigma \hat{x} + c^T \hat{x} \geq \delta^T y - \frac{1}{2}\min_i \delta^T e_i \Rightarrow \\ &\qquad y^T \Sigma \hat{x} + c^T \hat{x} \geq \delta^T (y - \mathcal{H}\hat{x}) - \frac{1}{2}\min_i \delta^T e_i \Rightarrow y = \mathcal{H}\hat{x}. \end{aligned}$$

The last equality holds since $\mathcal{H}\hat{x} \leq y$, and $\frac{1}{2}\delta^T e_i > \max \left\{ |y^T \Sigma x + c^T x| \middle| Gx \leq b, \mathcal{H}x \leq y \right\}$ for all i and for all feasible x by definition of δ. Note that if $\mathcal{H}\hat{x} < y$, then $\delta^T (y - \mathcal{H}\hat{x}) \geq \min_i \delta^T e_i$ since both y and $\mathcal{H}\hat{x}$ are integer vectors.

Corollary 5.1 *If $z^{\delta,y}(\beta) \geq \delta^T y - \frac{1}{2}\min_i \delta^T e_i$, then $z^y(\beta) = z^{\delta,y}(\beta) - \delta^T y$.*

Denote R be the set of vectors $y \in \mathbb{Z}_+^\ell$ that satisfy $y = \mathcal{H}x$ for $x \in \left\{ x \in \mathbb{Z}_+^n \middle| Gx \leq b \right\}$. Formally,

$$R = \{ y \in \mathbb{Z}_+^\ell \mid Gx \leq b,\ \mathcal{H}x = y,\ x \in \mathbb{Z}_+^n \}. \tag{33}$$

Lemma 5.6 $z(\beta) = \max_{y \in R} z^y(\beta)$, *for $\beta \in \mathbf{B}$.*

Lemma 5.6 directly follows since set R contains all possible values that $\mathcal{H}x$ can take for $\beta \in \mathbf{B}$. If ℓ is small and matrix $\mathcal{H}$ is sparse, then all possible y's in set R might be enumerated. Otherwise, let $\bar{y}_i = \max_x\{(\mathcal{H}x)^T e_i \,\Big|\, Gx \leq b\},\ i = 1 \ldots \ell$ and define $R' \supseteq R$ by

$$R' = \{y \in \mathbb{Z}_+^\ell | \mathbf{0} \leq y \leq \bar{y}\}.$$

We first give an iterative algorithm that searches over R'. Then, in Section 5.4 we present a modification of this algorithm which enumerates all vectors in set R. Let $\mathfrak{G} = [G \quad \mathcal{H}]$ and for $y \in R'$ define

$$\mathbf{\Pi}_y = \left\{\pi \in \mathbb{Z}_+^{m+\ell} | \pi_i \leq b_i, \forall i \leq m \text{ and } \pi_i \leq y_{i-m}, \forall i > m\right\}.$$

Let $\mathbf{\Pi}_{yj}$ denote the set consisting of all $\pi \in \mathbf{\Pi}_y$ such that $\pi \geq \mathfrak{g}_j \in \mathfrak{G}$.

Algorithm 3. *Sparse-Fixing Algorithm.*

Step 0: Set $\tau \leftarrow 1$. Denote the τ^{th} vector in R' by r^τ. Initialize the global bound $v^0(\beta) = 0$ for all $\beta \in \mathbf{B}$.

Step 1: Set $y \leftarrow r^\tau$. Let $\gamma^T = \delta^T \mathcal{H} + y^T Q + c^T$ and $\hbar = [b \quad y]$. Initialize the lower bound $l^0(\pi) = 0$ for all $\pi \in \mathbf{\Pi}_y$. For $j = 1, \ldots, n$, if $\mathfrak{g}_j \in \mathbf{\Pi}_y, l^0(\mathfrak{g}_j) = \gamma_j$ and insert $\mathfrak{g}_j$ into a vector list $\mathcal{L}$. Set $l^1(\pi) = l^0(\pi)$ for all $\pi \in \mathbf{\Pi}_y$. Set $k \leftarrow 1$.

Step 2: Denote the k^{th} vector in $\mathcal{L}$ by π^k and the i^{th} element of a vector π by π_i. Let $\pi = \pi^k$. Update all vectors π' such that $\pi' \in \mathbf{\Pi}_y$ and $\pi' \geq \pi^k$ with the following lexicographic order:

(2a) Set $\pi_1 \leftarrow \pi_1 + 1$ and $l^k(\pi) \leftarrow \max\{l^k(\pi), l^k(\pi^k) + l^k(\pi - \pi^k)\}$.

(2b) If $\pi_1 \geq \hbar_1$, go to Step (2c); otherwise, go to Step (2a).

(2c) If $\pi_i \geq \hbar_i$ for all $i = 1, \ldots, m + \ell$, go to Step 3. Otherwise, let $s = \min\{i : \pi_i < \hbar_i\}$. Set $\pi_i \leftarrow \pi_i^k$ for $i = 1, \ldots, s-1$. Set $\pi_s \leftarrow \pi_s + 1$ and go to Step (2a).

Step 3: If $k = |\mathcal{L}|$, go to Step 4. Otherwise, $l^{k+1}(\pi) \leftarrow l^k(\pi)$ for all $\pi \in \mathbf{\Pi}_y$, set $k \leftarrow k + 1$ and go to Step 2.

Step 4: For all $\beta \in \cup_{j=1}^n B_j$, let $\beta_y = [\beta \quad y]$. If $l^k(\beta_y) \geq \delta^T y - \frac{1}{2}\min_i \delta^T e_i$, then update $v^\tau(\beta) \leftarrow \max\{l^k(\beta_y) - \delta^T y, v^{\tau-1}(\beta)\}$. If $\tau = |R'|$, stop. Otherwise, empty out vector list $\mathcal{L}$, set $\tau \leftarrow \tau + 1$ and go to Step 1.

Lemma 5.7 *If $\beta \in \mathbf{B} \setminus \cup_{j=1}^{n} B_j$, then $\beta_y = [\beta \quad y] \in \mathbf{\Pi}_y \setminus \cup_{j=1}^{n} \mathbf{\Pi}_{yj}$ for all $y \in R'$.*

Proposition 5.5 *$z^{\delta,y}(\beta) = 0$, $\forall \beta \in \mathbf{B} \setminus \cup_{j=1}^{n} B_j$.*

For $\pi \in \mathbf{\Pi}_y$, we define π^+ to be the subvector composed by the first m elements of π and π^- to be the subvector composed by the last ℓ elements of π.

Theorem 5.2 *The Sparse-Fixing Algorithm terminates with optimal solutions to $\mathbf{P}(b)$ in at most $|R'|$ iterations of τ.*

Proof: First, consider a particular $y \in R'$. Define $\beta_y = [\beta \quad y]$. For any $\pi \in \mathbf{\Pi}_y \setminus \cup_{j=1}^{n} \mathbf{\Pi}_{yj}$, we initialize $l^0(\pi) = 0$ in Step 1 and do not update them afterward. By Proposition 5.5, $z^{\delta,y}(\beta) = 0$ for $\beta \in \mathbf{B} \setminus \cup_{j=1}^{n} B_j$. Assume the algorithm enters Step 4, when $k^* = |\mathcal{L}|$. Then for all $\beta \in \mathbf{B} \setminus \cup_{j=1}^{n} B_j$, $l^{k^*}(\beta_y) = z^{\delta,y}(\beta)$ since $\beta_y \in \mathbf{\Pi}_y \setminus \cup_{j=1}^{n} \mathbf{\Pi}_{yj}$ by Lemma 5.7. Suppose there exists $\beta \in \cup_{j=1}^{n} B_j$ such that $l^{k^*}(\beta_y) \neq z^{\delta,y}(\beta)$ and for all $\pi \leq \beta_y$ and $\pi \in \mathbf{\Pi}_y$, $l^{k^*}(\pi) = z^{\delta,\pi^-}(\pi^+)$. Then $l^{k^*}(\beta_y) < z^{\delta,y}(\beta)$ by Lemma 5.3. It follows there exists a $j^* \in \{1, \ldots, n\}$, such that $l^{k^*}(\beta_y) < \gamma_{j^*} + z^{\delta,(y-\mathfrak{g}_{j^*}^-)}(\beta - \mathfrak{g}_{j^*}^+)$ by Theorem 5.1. Since $l^{k^*}(\mathfrak{g}_{j^*}) \geq \gamma_{j^*}$ and $l^{k^*}(\beta_y - \mathfrak{g}_{j^*}) = z^{\delta,(y-\mathfrak{g}_{j^*}^-)}(\beta - \mathfrak{g}_{j^*}^+)$, it follows that $l^{k^*}(\mathfrak{g}_{j^*}) + l^{k^*}(\beta_y - \mathfrak{g}_{j^*}) \geq \gamma_{j^*} + z^{\delta,(y-\mathfrak{g}_{j^*}^-)}(\beta - \mathfrak{g}_{j^*}^+) > l^{k^*}(\beta_y)$, which contradicts the superadditivity of $l^{k^*}(\cdot)$. As a result, $l^{k^*}(\beta_y) = z^{\delta,y}(\beta), \forall \beta \in \cup_{j=1}^{n} B_j$ when the algorithm enters Step 4. From Lemma 5.5, $l^{k^*}(\beta_y) = z^{\delta,y}(\beta) \geq \delta^T y - \frac{1}{2} \min_i \delta^T e_i, \forall \beta \in \cup_{j=1}^{n} B_j$ if and only if $y \in R$. In Step 4, we subtract $\delta^T y$ from $z^{\delta,y}(\beta)$ obtaining $z^y(\beta), \forall \beta \in \cup_{j=1}^{n} B_j$ by Corollary 5.1.

The algorithm searches over every $y \in R'$ by updating the global lower bound $v^\tau(\beta)$ if $y \in R$. As a result, $v^{|R'|}(\beta) = \max_{y \in R} z^y(\beta) = z(\beta), \forall \beta \in \cup_{j=1}^{n} B_j$ by Lemma 5.6.

Proposition 5.6 *The Sparse-Fixing Algorithm runs in $O(\sum_{y \in R'} n|\mathbf{\Pi}_y|)$ time.*

Proof: For each $y \in R'$, the algorithm makes $O(n|\mathbf{\Pi}_y|)$ calculations since $|\mathcal{L}|$ is at most n and Step 2 requires $O(|\mathbf{\Pi}_y|)$ calculations. The algorithm iterates over all $y \in R'$, so the overall running time is $O(\sum_{y \in R'} n|\mathbf{\Pi}_y|)$.

5.4 An Iterative Fixing Algorithm for Low-Rank Q with Enumeration of the Set R

In this section, we present a modified version of the sparse fixing algorithm which enumerates all y's in set R. Our computational experiments show that, this version runs much faster than the previous one if ℓ is small and matrix $\mathcal{H}$ is sparse.

Let $\hat{\mathcal{H}}$ be the set of nonzero columns in $\mathcal{H}$. We define $\hat{G} \subseteq G$ and $\hat{\Sigma} \subseteq \Sigma$ to be the submatrices corresponding to the columns in $\hat{\mathcal{H}}$. Likewise, let $\hat{c}$ be the linear part of the objective function corresponding to the columns in $\hat{\mathcal{H}}$. We give the set of feasible solutions for the columns in $\hat{\mathcal{H}}$ by:

$$\hat{X} = \left\{ x \in \mathbb{Z}_+^{|\hat{\mathcal{H}}|} : \hat{G}x \leq b \right\}.$$

Algorithm 4. *Sparse-Enumeration Algorithm.*

Step 0: Set $t \leftarrow 1$. Denote the t^{th} vector in $\hat{X}$ by x^t. Initialize the global bound $v^0(\beta) = 0$ for all $\beta \in \mathbf{B}$.

Step 1: Set $y \leftarrow \hat{\mathcal{H}}x^t$. Let $\gamma^T = y^T\Sigma + c^T$ and $\hbar = b - \hat{G}x^t$. Define the set $\mathbf{B}^t = \{\beta \in \mathbb{Z}^m | \beta_i \leq \hbar_i, \forall i \leq m\}$ and for $j \notin \hat{\mathcal{H}}$ let B_j^t to be the set of all $\beta \in \mathbf{B}^t$ such that $\beta \geq g_j$. Initialize the lower bound $l^0(\beta) = 0$ for all $\beta \in \mathbf{B}^t$. For $j \notin \hat{\mathcal{H}}$, if $g_j \in \mathbf{B}^t, l^0(g_j) = \gamma_j$ and insert g_j into a vector list $\mathcal{L}$. Set $l^1(\beta) = l^0(\beta)$ for all $\beta \in \mathbf{B}^t$. Set $k \leftarrow 1$.

Step 2: Denote the k^{th} vector in $\mathcal{L}$ by β^k and the i^{th} element of a vector β by β_i. Let $\beta = \beta^k$. Update all vectors β' such that $\beta' \in \mathbf{B}^t$ and $\beta' \geq \beta^k$ with the following lexicographic order:

(2a) Set $\beta_1 \leftarrow \beta_1 + 1$ and $l^k(\beta) \leftarrow \max\{l^k(\beta), l^k(\beta^k) + l^k(\beta - \beta^k)\}$.

(2b) If $\beta_1 \geq \hbar_1$, go to Step (2c); otherwise, go to Step (2a).

(2c) If $\beta_i \geq \hbar_i$ for all $i = 1, \ldots, m$, go to Step 3. Otherwise, let $s = \min\{i : \beta_i < \hbar_i\}$. Set $\beta_i \leftarrow \beta_i^k$ for $i = 1, \ldots, s-1$. Set $\beta_s \leftarrow \beta_s + 1$ and go to Step (2a).

Step 3: If $k = |\mathcal{L}|$, go to Step 4. Otherwise, $l^{k+1}(\beta) \leftarrow l^k(\beta)$ for all $\beta \in \mathbf{B}^t$, set $k \leftarrow k+1$ and go to Step 2.

Step 4: For all $\beta \in \cup_{j \notin \hat{\mathcal{H}}} B_j^t$, update $v^t(\beta + \hat{G}x^t) \leftarrow \max\{l^k(\beta) + y^T\hat{\Sigma}x^t + \hat{c}^T x^t, v^{t-1}(\beta + \hat{G}x^t)\}$. If $t = |\hat{X}|$, stop. Otherwise, empty out vector list $\mathcal{L}$, set $t \leftarrow t+1$ and go to Step 1.

Theorem 5.3 *The Sparse-Enumeration Algorithm terminates with optimal solutions to $z(\cdot)$ in at most $|\hat{X}|$ iterations of t.*

Proof: First, consider a particular $x^t \in \hat{X}$ and let $y = \hat{\mathcal{H}}x^t$. For any $\beta \in \mathbf{B}^t \setminus \cup_{j \notin \hat{H}} B_j^t$, we initialize $l^0(\beta) = 0$ in Step 1 and do not update them afterward. Let $z^t(\cdot)$ be the optimum objective function value of the problem obtained after setting the columns in $\hat{\mathcal{H}}$ to $x_t \in \mathbb{Z}_+^{|\hat{\mathcal{H}}|}$. By Proposition 5.5, $z^t(\beta) = 0$ for $\beta \in \mathbf{B}^t \setminus \cup_{j \notin \hat{\mathcal{H}}} B_j^t$. Assume the algorithm enters Step 4, when $k^* = |\mathcal{L}|$. Then $l^{k^*}(\beta) = z^t(\beta)$ for all $\beta \in \mathbf{B}^t \setminus \cup_{j \notin \hat{\mathcal{H}}} B_j^t$. Suppose there exists $\beta \in \cup_{j \notin \hat{\mathcal{H}}} B_j^t$ such that $l^{k^*}(\beta) \neq z^t(\beta)$ and for all $\beta' \leq \beta$ and $\beta' \in \cup_{j \notin \hat{\mathcal{H}}} B_j^t$, $l^{k^*}(\beta') = z^t(\beta')$. Then clearly $l^{k^*}(\beta) < z^t(\beta)$ by Lemma 5.3. It follows that there exists a $j^* \notin \hat{\mathcal{H}}$, such that $l^{k^*}(\beta) < \gamma_{j^*} + z^t(\beta - g_{j^*})$ by Theorem 5.1. Since $l^{k^*}(g_{j^*}) \geq \gamma_{j^*}$ and $l^{k^*}(\beta - g_{j^*}) = z^t(\beta - g_{j^*})$, it follows that $l^{k^*}(g_{j^*}) + l^{k^*}(\beta - g_{j^*}) \geq \gamma_{j^*} + z^t(\beta - g_{j^*}) > l^{k^*}(\beta)$, which contradicts the superadditivity of $l^{k^*}(\cdot)$. Hence, $l^{k^*}(\beta) = z^t(\beta)$ for all $\beta \in \cup_{j=1}^n B_j^t$. In Step 4, the algorithm updates the global bounds for all $\beta \in \cup_{j=1}^n B_j$ for which setting the variables corresponding to columns in $\hat{\mathcal{H}}$ to x^t is feasible. Note that we also add the constant value gained from fixed variables to $l^k(\beta)$.

If $x^t \in \hat{X}$, then $\hat{\mathcal{H}}x = y \in R$. We enumerate each $y \in R$ by considering every $x^t \in \hat{X}$. The algorithm searches over all $x^t \in \hat{X}$ by updating the global lower bound $v^t(\beta)$. As a result, $v^{|\hat{X}|}(\beta) = \max_{y \in R} z^y(\beta) = z(\beta), \forall \beta \in \cup_{j=1}^n B_j$ by Lemma 5.6.

Proposition 5.7 *The Sparse-Enumeration Algorithm runs in $O(\sum_{x^t \in \hat{X}} n|\mathbf{B}^t|)$ time.*

Proof: The proof is similar to that of Proposition 5.6.

6 Finding the Optimal Tender

In this section we propose two methods to determine an optimal tender β^* for (P2) given the value functions in both stages. These methods are more efficient than exhaustive search which evaluates the objective function in (P2) for all $\beta \in \mathbf{B}^1$. Our first method is a global branch-and-bound algorithm in which bounds are designed for hyper-rectangular partitions of $\mathbf{B}^1$. The second method is a level-set approach that evaluates the objective function in (P2) only for a subset of $\mathbf{B}^1$.

6.1 A Global Branch-and-Bound Algorithm

Our global branch-and-bound algorithm is based on a generic framework described in Horst and Tuy [34]. In the following description of the algorithm, $\mathcal{M}$ is a list of unfathomed hyper-rectangles, each of which is associated with a subproblem of the form $f^k = \max\left\{\psi(\beta) + \mathbb{E}_\xi \phi(h(\omega) - \beta) \mid \beta \in \mathcal{P}^k \cap \mathbb{Z}^{m_2}\right\}$, and a lower bound $\mu^k \leq f^k$ and an upper bound $v^k \geq f^k$.

Algorithm 5. A global branch-and-bound algorithm to determine an optimal tender β^* for (P2).

Step 0: (Initialization) Construct a hyper-rectangle $\mathcal{P}^0 := [\lambda^0, \eta^0] = \Pi_{i=1}^{m_2} [\lambda_i^0, \eta_i^0]$ such that $\mathbf{B}^1 \subseteq \mathcal{P}^0 \cap \mathbb{Z}^{m_2}$. Initialize list $\mathcal{M} \leftarrow \{\mathcal{P}^0\}$ and a global lower bound $L = \psi(\beta^0) + \mathbb{E}_\xi \phi(h(\omega) - \beta^0)$ with an arbitrarily selected $\beta^0 \in \mathbf{B}^1$. Set $\mu^0 = \psi(\lambda^0) + \mathbb{E}_\xi \phi(h(\omega) - \eta^0)$ and $v^0 = \psi(\eta^0) + \mathbb{E}_\xi \phi(h(\omega) - \lambda^0)$. Set $k \leftarrow 1$.

Step 1: (Subproblem selection) If $\mathcal{M} = \emptyset$, terminate with optimal solution β^*; otherwise, select and delete from $\mathcal{M}$ a hyper-rectangle $\mathcal{P}^k := \left[\lambda^k, \eta^k\right] = \Pi_{i=1}^{m_2} \left[\lambda_i^k, \eta_i^k\right]$.

Step 2: (Subproblem pruning)

(2a) If $v^k \leq L$ or $\mathcal{P}^k \cap \mathbf{B}^1 = \emptyset$, go to Step 1.

(2b) If $\mu^k < v^k$, i.e., $\mathcal{P}^k$ is an unfathomed hyper-rectangle, go to Step 3.

(2c) If $\mu^k = v^k$ and $L < \mu^k$, update $L = \mu^k = v^k = f^k$, and arbitrarily select $\beta \in \mathcal{P}^k \cap \mathbf{B}^1$ and set $\beta^* = \beta$.

(2d) Delete from $\mathcal{M}$ all hyper-rectangles $\mathcal{P}^{k'}$ with $v^{k'} \leq L$ and go to Step 1.

Step 3: (Subproblem partitioning) Choose a dimension $i', 1 \leq i' \leq m_2$, such that $\lambda_{i'}^k < \eta_{i'}^k$. Divide $\mathcal{P}^k$ into two hyper-rectangles $\mathcal{P}^{k_1}$ and

$\mathcal{P}^{k_2}$ along dimension i' as: $\mathcal{P}^{k_1} := \left[\lambda^{k_1}, \eta^{k_1}\right] = \left[\lambda^k_{i'}, \left\lfloor(\eta^k_{i'} + \lambda^k_{i'})/2\right\rfloor\right] \times \Pi_{i \neq i'} \left[\lambda^k_i, \eta^k_i\right]$ and $\mathcal{P}^{k_2} := \left[\lambda^{k_2}, \eta^{k_2}\right] = \left[\left\lfloor(\eta^k_{i'} + \lambda^k_{i'})/2\right\rfloor + 1, \eta^k_{i'}\right] \times \Pi_{i \neq i'} \left[\lambda^k_i, \eta^k_i\right]$. Add the two hyper-rectangles $\mathcal{P}^{k_i}, i = 1, 2$, to $\mathcal{M}$, i.e., $\mathcal{M} \leftarrow \mathcal{M} \cup \left\{\mathcal{P}^{k_1}, \mathcal{P}^{k_2}\right\}$. Set $\mu^{k_i} = \psi(\lambda^{k_i}) + \mathbb{E}_\xi \phi(h(\omega) - \eta^{k_i})$ and $v^{k_i} = \psi(\eta^{k_i}) + \mathbb{E}_\xi \phi(h(\omega) - \lambda^{k_i}), i = 1, 2$. Set $k \leftarrow k + 1$ and go to Step 1.

Theorem 6.1 *[43] Algorithm 5 terminates with an optimal solution to (P2) after a finite number of steps.*

6.2 The Minimal Tender Approach

In this section we describe a level-set approach to reduce the search space $\mathbf{B}^1$ based on the observation that there must exist an optimal right-hand side vector β^* to $(P2)$ satisfying that each smaller right-hand side vector has also a strictly smaller objective value in the first stage. We call such right-hand side vectors *minimal tenders*, and form a set that only contains all minimal tenders and thus has fewer candidate solutions. However, this approach only applies when the technology matrix T in *(P1)* is nonnegative. In those cases, $\mathbf{B}^1 \subset \mathbb{Z}^{m_2}_+$. The minimal tender approach is first introduced in Kong et al. [43] for linear IPs. We generalize their results to QIPs.

Definition 6.1 *[43] A vector $\beta \in \mathbf{B}^1$ is a minimal tender if for all i, $i = 1, \ldots, m_2$, either $\beta_i = 0$ or $\psi(\beta - e_i) < \psi(\beta)$, where e_i is the i^{th} unit vector. We let Θ be the set containing all minimal tenders in $\mathbf{B}^1$.*

Theorem 6.2 *[43] There exists an optimal solution to $(P2)$ that is a minimal tender. That is,*

$$\max_{\beta \in \Theta} \left\{\psi(\beta) + \mathbb{E}_\xi \phi(h(\omega) - \beta)\right\} = \max_{\beta \in \mathbf{B}^1} \left\{\psi(\beta) + \mathbb{E}_\xi \phi(h(\omega) - \beta)\right\}.$$

Define $\rho = |\Theta|/|\mathbf{B}^1|$. Intuitively, as $\rho \to 1$, the computational benefit of searching Θ could be surpassed by the computational burden of determining Θ. The value of ρ is typically not known until $\psi(\cdot)$ is completely determined. In some special cases, Θ and ρ can be identified analytically.

Proposition 6.1 *Suppose $\forall i$ $c_i + \frac{1}{2}\Lambda_{ii} > 0$ and $\forall j \neq i$ $\Lambda_{ij} \geq 0$. Then if T contains I_{m_2}, the m_2-dimensional identity matrix as a submatrix, then $\Theta = \mathbf{B}^1$ and so that $\rho = 1$.*

Let $\overline{opt}(\beta) \subseteq \mathbb{Z}_+^{n_1}$ denote the set of optimal solutions to $\psi(\cdot)$ for $\beta \in \mathbb{Z}_+^{m_2}$.

Lemma 6.1 *[43] For any $\beta \in \Theta$ and for any $\hat{x} \in \overline{opt}(\beta)$, $T\hat{x} = \beta$.*

Proposition 6.2 *Let $\Lambda = diag(\Lambda_{11}, \ldots, \Lambda_{nn}) \succeq 0$ and $\hat{x} \in \overline{opt}(\beta)$. If $\beta \in \Theta \setminus \{\boldsymbol{0}\}$, then for all x such that $\forall i$ $x_i = 0$ or $x_i = \hat{x}_i$, we have $Tx \in \Theta$.*

Proof:

Since $\forall i, j$ $\Lambda_{ij} \geq 0$, then by Proposition 3.8, $\psi(\beta)$ is superadditive. Suppose for some $\beta \in \Theta \setminus \{\mathbf{0}\}$ and $\hat{x} \in \overline{opt}(\beta)$ we have $T\hat{x} \in \Theta$. Let $x \leq \hat{x}$ such that $\forall i$ $x_i = \hat{x}_i$ or $x_i = 0$. Then by Corollary 3.5 $\psi(Tx) = \frac{1}{2}x^T\Lambda x + c^T x$. Assume that $Tx \notin \Theta$. Then there exists an $i \in \{1, \ldots, m_2\}$, such that $Tx - e_i \geq 0$ and $\psi(Tx - e_i) = \psi(Tx)$. Let $y \in \overline{opt}(Tx - e_i)$. Thus $y \neq x$, $y \in S_1(Tx - e_i)$, and $\frac{1}{2}x^T\Lambda x + c^T x = \frac{1}{2}y^T\Lambda y + c^T y$. Consider $\tilde{x} = \hat{x} - x + y$. Then, $T\tilde{x} = T(\hat{x} - x + y) \leq T\hat{x} - Tx + Tx - e_i = T\hat{x} - e_i$ and $\tilde{x} \in S_1(T\hat{x} - e_i)$.

Since $\forall i$ $x_i = 0$ or $x_i = \hat{x}_i$ we have that

$$\sum_i \Lambda_{ii}\hat{x}_i x_i = \sum_i \Lambda_{ii} x_i^2.$$

Taking into account the above equality, consider the objective function value for $\tilde{x}$:

$$\begin{aligned}
&\frac{1}{2}(\hat{x} - x + y)^T\Lambda(\hat{x} - x + y) + c^T(\hat{x} - x + y) \\
=&\frac{1}{2}\sum_i \Lambda_{ii}(\hat{x}_i - x_i)^2 + \frac{1}{2}\sum_i \Lambda_{ii} y^2 + \sum_i \Lambda_{ii}(\hat{x}_i - x_i)y + c^T(\hat{x} - x + y) \\
\geq&\frac{1}{2}\sum_i \Lambda_{ii}\hat{x}_i^2 - \sum_i \Lambda_{ii}\hat{x}_i x_i + \frac{1}{2}\sum_i \Lambda_{ii} x_i^2 + \frac{1}{2}\sum_i \Lambda_{ii} y^2 + c^T(\hat{x} - x + y) \\
=&\frac{1}{2}\sum_i \Lambda_{ii}\hat{x}_i^2 + c^T\hat{x} - (\frac{1}{2}\sum_i \Lambda_{ii} x_i^2 + c^T x) + \frac{1}{2}\sum_i \Lambda_{ii} y^2 + c^T y \\
=&\frac{1}{2}\hat{x}^T\Lambda\hat{x} + c^T\hat{x},
\end{aligned}$$

which implies that $\psi(T\hat{x} - e_i) = \psi(T\hat{x})$. Therefore, $T\hat{x} \notin \Theta$, which contradicts the fact that $T\hat{x} = \beta \in \Theta$.

Corollary 6.1 *Let $\Lambda = diag(\Lambda_{11}, \ldots, \Lambda_{nn}) \succeq 0$. For any $\beta \in \Theta \setminus \{\mathbf{0}\}$ and $\hat{x} \in \overline{opt}(\beta)$, if $\hat{x}_\ell \geq 1$, then $\hat{x}_\ell t_\ell \in \Theta$.*

Proof: Let $x \in \mathbb{Z}^n_+$ such that $x_\ell = \hat{x}_\ell$ and $x_i = 0$ for all $i \neq \ell$. Then the result follows directly from Proposition 6.2.

Remark 6.1 *Corollary 6.1 is a generalization of the result from Kong et al. [43] for linear integer programs, which states that for any $\beta \in \Theta \setminus \{\mathbf{0}\}$ and $\hat{x} \in \overline{opt}(\beta)$, if $\hat{x}_\ell \geq 1$, then $t_\ell \in \Theta$. Proposition 6.2 and Corollary 6.1 hold for linear IPs, however the result of Kong et al. [43] does not hold for diagonal QIPs. We illustrate this with the following example:*

$$\max \left\{ 3x_1^2 + x_2^2 + 2x_1 + 4x_2 \mid 2x_1 + x_2 \leq 4,\ 2x_1 + 2x_2 \leq 4,\ x \in \mathbb{Z}^2_+ \right\}.$$

Clearly, $\hat{x} = (2,0)^T \in \overline{opt}((4,4)^T)$ with an objective function value of $z((4,4)^T) = 16$. $(4,4)^T$ is a minimal tender since $z((3,4)^T) = 12 < 16$ and $z((4,3)^T) = 5 < 16$. Note that, $t_1 = (2,2)^T$ and $\hat{x}_1 t_1 = (4,4)^T$. Finally, observe that $t_1 = (2,2)^T$ is not a minimal tender since $z((2,2)^T) = z((1,2)^T) = 5$.

We show that if a particular variable x_ℓ has only linear part in the objective function (i.e., $\Lambda_{\ell\ell} = 0$), then the result of Kong et al. [43] holds and $t_\ell \in \Theta$.

Proposition 6.3 *Let $\Lambda = diag(\Lambda_{11}, \ldots, \Lambda_{nn}) \succeq 0$. For any $\beta \in \Theta \setminus \{\mathbf{0}\}$ and $\hat{x} \in \overline{opt}(\beta)$, if $\hat{x}_\ell \geq 1$ and $\Lambda_{\ell\ell} = 0$, then $t_\ell \in \Theta$.*

Proof: The proof is similar to that of Proposition 6.2.

6.3 Reduction of the Primal Formulation Using Minimal Tenders

In this section we assume that Λ is a diagonal matrix with non-negative diagonal elements, i.e., $\forall i\ \Lambda_{ii} \geq 0$ and $\forall j \neq i\ \Lambda_{ij} = 0$. Define $\mathcal{T}$ to be the index set of columns t_j in T such that

- if $\Lambda_{jj} > 0$ then there exists integer k such that $kt_j \in \Theta$,
- if $\Lambda_{jj} = 0$ then $t_j \in \Theta$,

i.e.,

$$\mathcal{T} = \left\{ j \in \{1, \ldots, n_1\} \mid \Lambda_{jj} > 0 \text{ and } \exists k \in \mathbb{Z}^1_+ \text{ s.t. } kt_j \in \Theta \right\} \bigcup \left\{ j \in \{1, \ldots, n_1\} \mid \Lambda_{jj} = 0 \text{ and } t_j \in \Theta \right\}.$$

Next we show that columns $j \notin \mathcal{T}$ can be deleted when computing $\psi(\beta)$ for $\beta \in \Theta$. For any $\beta \in \mathbb{Z}_+^{m_2}$, define

$$\psi'(\beta) = \max\left\{ \frac{1}{2}\sum_{j\in\mathcal{T}} \Lambda_{jj}x_j^2 + \sum_{j\in\mathcal{T}} c_j x_j \;\Big|\; x \in S_1'(\beta) \right\}, \tag{34}$$

where

$$S_1'(\beta) = \left\{ x \in \mathbb{Z}_+^{|\mathcal{T}|} \;\Bigg|\; \sum_{j\in\mathcal{T}} a_j x_j \le b,\ \sum_{j\in\mathcal{T}} t_j x_j \le \beta \right\}. \tag{35}$$

Then the reduced superadditive dual reformulation is

$$\max_{\beta\in\Theta} \left\{ \psi'(\beta) + \mathbb{E}_\xi \phi(h(\omega) - \beta) \right\}. \tag{36}$$

Lemma 6.2 *For $\beta \in \Theta$, $\psi'(\beta) = \psi(\beta)$.*

Proof: The result is trivial for $\beta = \mathbf{0}$. For any $\beta \in \Theta \setminus \{\mathbf{0}\}$ it follows directly from Corollary 6.1 and Proposition 6.3.

Theorem 6.3 *There exists an optimal solution to $(P2)$ that is an optimal solution to (36). That is,*

$$\max_{\beta\in\Theta} \left\{ \psi'(\beta) + \mathbb{E}_\xi \phi(h(\omega) - \beta) \right\} = \max_{\beta\in\mathbf{B}^1} \left\{ \psi'(\beta) + \mathbb{E}_\xi \phi(h(\omega) - \beta) \right\}.$$

Proof: Follows from Lemma 6.2 and Theorem 6.2.

Corollary 6.2 *Let β^* be an optimal solution to (36). Then*

$$\hat{x} \in \text{argmax}\left\{ \frac{1}{2}x^T\Lambda x + c^T x \,\Big|\, Ax \le b,\ Tx \le \beta^*,\ x \in \mathbb{Z}_+^{n_1} \right\}.$$

is an optimal solution to $(P1)$. Furthermore, the optimal objective values of the two problems are equal.

Corollary 6.3 *There exists an optimal solution x^* to $(P1)$ where $x_j^* = 0$ for $j \notin \mathcal{T}$.*

7 Computational Experiments

We perform our computational experiments on randomly generated instances under the assumptions **A1** $-$ **A4**. We do not consider the first-stage constraints $Ax \leq b$ as they can be embedded into the technology matrix T by setting the corresponding rows of the recourse matrix W to $\mathbf{0}$. To test all four algorithms for finding the value function in the first phase, and both approaches for finding the optimal tender in the second phase we assume that $\mathbf{B}^k = \mathcal{B}^k \cap \mathbb{Z}_+^{m_2}$ where $\mathcal{B}^k = \Pi_{i=1}^{m_2}[0, b_i^k]$, $k = 1, 2$.

We consider SQIP instances whose first- and second-stage quadratic objective functions are given by $(\chi_1^T \bar{x})(\sigma_1^T x) + c^T x$ and $(\chi_2^T \bar{y})(\sigma_2^T y) + d^T y$, respectively. We have that $\sigma_1 \in \mathbb{Z}^{n_1}, \sigma_2 \in \mathbb{Z}^{n_2}, \chi_1 \in \mathbb{Z}^{a_1}$ for some $a_1 \leq n_1$, and $\chi_2 \in \mathbb{Z}^{a_2}$ for some $a_2 \leq n_2$. Note that $\bar{x} \in \mathbb{Z}^{a_1}$ and $\bar{y} \in \mathbb{Z}^{a_2}$ are variable vectors whose elements are composed of a_1 and a_2 dimensional subsets of the indices in x and y, respectively.

We randomly generate two different testbeds. Testbed 1 in Table 1 has 40 instance classes and Testbed 2 in Table 2 has 10 instance classes. There are more instances in Testbed 1 as we vary the dimensions of $\bar{x} \in \mathbb{Z}_+^{a_1}$ and $\bar{y} \in \mathbb{Z}_+^{a_2}$ between 10 and 30; whereas in Testbed 2 we always set $a_1 = n_1$ and $a_2 = n_2$. Note that quadratic objective functions of Testbed 1 instances are specially structured so that the Sparse-Fixing and Sparse-Enumeration Algorithms (i.e., Algorithms 3 and 4) apply. However, instances in Testbed 2 do not exhibit any special structure.

The instances are named IC$m-KX$, where $m = 1, 2$ is the testbed index, $K = 1, \ldots, 40$ is the instance class index, and $X \in \{S, L\}$ is the instance size index. For a particular instance class K in testbed m, size index S denotes the instance that has smaller numbers of variables, i.e., n_1 and n_2; whereas size index L denotes the larger instance.

In Tables 1 and 2, the numbers listed under $h(\omega)$ are the lower and upper bounds of the uniform distribution that is used to generate stochastic right-hand side vectors in $\mathbb{Z}_+^{m_2}$. We denote by υ the density of technology matrix T and recourse matrix W, which is modeled by a Bernoulli distribution. Elements of matrix T are generated from $U[1, 4]$, i.e., uniformly distributed integers between 1 and 4, and elements of matrix W are generated from $U[2, 5]$. Moreover, deterministic parameters c_j, χ_{1j} are generated from $U[1, 5]$; d_j, χ_{2j} are generated from $U[2, 6]$; and σ_{1j}, σ_{2j} are generated from $U[1, 3]$. We choose these parameters as well as the number of scenarios and constraints similar to that of test instances of two-stage linear integer programs with stochastic right-hand sides used for computational experiments in Kong et al. [43].

At each iteration of the Exact-Superadditive Algorithm (i.e., Algorithm 1), we use a dynamic programming algorithm to solve the quadratic integer programs arising in Step 1. In the global branch-and-bound algorithm we set the initial global lower bound to $\max\{\mathbb{E}_\xi \phi(h(\omega)), \psi(b^1) + \mathbb{E}_\xi \phi(h(\omega) - b^1)\}$, the maximum between the objective function values of $(P2)$ with respect to $\mathbf{0}$ and b^1. With the minimal tender approach, we do not explore the possible computational benefits of constructing the reduced formulation (36). After obtaining the first-stage value function, we simply check if each $\beta \in \mathbf{B}^1$ is a minimal tender by definition, and then form the minimal tender set Θ. An optimal solution to $(P2)$ is obtained by evaluating the objective function with respect to each $\beta \in \Theta$. All computational experiments are conducted on an Intel Xeon PC with 3GHz CPU and 3GB of RAM.

7.1 Finding the Value Function

In this section we present computational results of the algorithms proposed in Section 5 for constructing the value functions of the first- and second-stage QIPs. The Exact-Superadditive Algorithm is taken as a baseline for all comparisons since it does not require a particular structure of the quadratic objective function other than superadditivity. Recall that the Sparse-Fixing Algorithm enumerates all possible $(\chi_1^T \bar{x})$ values; whereas the Sparse-Enumeration Algorithm iterates over a_1 dimensional candidate solution vectors to $\bar{x}$. In the vast majority of test instances that we consider, the Sparse-Enumeration Algorithm runs faster than the Sparse-Fixing Algorithm, sometimes by several orders of magnitude. Therefore, we do not report computational results for the Sparse-Fixing Algorithm. However, we expect that this algorithm will outperform the Sparse-Enumeration Algorithm when $(\chi_1^T \bar{x})$ has a small range and large domain, since it will make less iterations in this case.

In Section 7.1.1 we consider the Exact-Superadditive and the Sparse-Enumeration Algorithms and perform the following tests :

1. Our implementation of the Exact-Superadditive Algorithm can solve instances with $n_1 \leq 400$. Therefore, to compare performances of the Exact-Superadditive and the Sparse-Enumeration Algorithms, we use small instances of Testbed 1 in Table 1.

Table 1: Characteristics of instances in Testbed 1, e.g., instance $\mathrm{IC1}-1S$ has $n_1 = 400$, $\mathrm{IC1}-1L$ has $n_1 = 1000$.

$\mathrm{IC1}-KX$	$\mathrm{IC1}-KS$				$\mathrm{IC1}-KL$				Common parameters			
	a_1	a_2	n_1	n_2	a_1	a_2	n_1	n_2	v	m_2	$h(\omega)$	Scenarios
$\mathrm{IC1}-1X$	10	10	400	400	10	10	1000	100	0.5	6	$[5, 9]$	7812
$\mathrm{IC1}-2X$	10	10	200	200	10	10	500	500	0.4	6	$[5, 9]$	7812
$\mathrm{IC1}-3X$	10	10	300	200	10	10	300	500	0.5	6	$[5, 9]$	7812
$\mathrm{IC1}-4X$	10	10	300	300	10	10	500	500	0.5	6	$[5, 9]$	7812
$\mathrm{IC1}-5X$	10	10	400	300	10	10	500	500	0.6	6	$[5, 9]$	7812
$\mathrm{IC1}-6X$	10	10	300	300	10	10	500	500	0.8	6	$[5, 9]$	7812
$\mathrm{IC1}-7X$	10	10	200	300	10	10	1000	500	0.7	7	$[5, 10]$	279936
$\mathrm{IC1}-8X$	10	10	300	200	10	10	500	500	0.7	7	$[5, 10]$	279936
$\mathrm{IC1}-9X$	10	10	400	200	10	10	300	500	0.7	7	$[5, 10]$	279936
$\mathrm{IC1}-10X$	10	10	400	400	10	10	1000	500	0.3	7	$[5, 10]$	279936
$\mathrm{IC1}-11X$	10	30	400	400	10	30	1000	100	0.5	6	$[5, 9]$	7812
$\mathrm{IC1}-12X$	10	30	200	200	10	30	500	500	0.4	6	$[5, 9]$	7812
$\mathrm{IC1}-13X$	10	30	300	200	10	30	300	500	0.5	6	$[5, 9]$	7812
$\mathrm{IC1}-14X$	10	30	300	300	10	30	500	500	0.5	6	$[5, 9]$	7812
$\mathrm{IC1}-15X$	10	30	400	300	10	30	500	500	0.6	6	$[5, 9]$	7812
$\mathrm{IC1}-16X$	10	30	300	300	10	30	500	500	0.8	6	$[5, 9]$	7812
$\mathrm{IC1}-17X$	10	30	200	300	10	30	1000	500	0.7	7	$[5, 10]$	279936
$\mathrm{IC1}-18X$	10	30	300	200	10	30	500	500	0.7	7	$[5, 10]$	279936
$\mathrm{IC1}-19X$	10	30	400	200	10	30	300	500	0.7	7	$[5, 10]$	279936
$\mathrm{IC1}-20X$	10	30	400	400	10	30	1000	500	0.3	7	$[5, 10]$	279936
$\mathrm{IC1}-21X$	15	15	400	400	15	15	1000	100	0.5	6	$[5, 7]$	7812
$\mathrm{IC1}-22X$	15	15	200	200	15	15	500	500	0.4	6	$[5, 7]$	7812
$\mathrm{IC1}-23X$	15	15	300	200	15	15	300	500	0.5	6	$[5, 7]$	7812
$\mathrm{IC1}-24X$	15	15	300	300	15	15	500	500	0.5	6	$[5, 7]$	7812
$\mathrm{IC1}-25X$	15	15	400	300	15	15	500	500	0.6	6	$[5, 7]$	7812
$\mathrm{IC1}-26X$	15	15	300	300	15	15	500	500	0.8	6	$[5, 7]$	7812
$\mathrm{IC1}-27X$	15	15	200	300	15	15	1000	500	0.7	7	$[5, 8]$	279936
$\mathrm{IC1}-28X$	15	15	300	200	15	15	500	500	0.7	7	$[5, 8]$	279936
$\mathrm{IC1}-29X$	15	15	400	200	15	15	300	500	0.7	7	$[5, 8]$	279936
$\mathrm{IC1}-30X$	15	15	400	400	15	15	1000	500	0.3	7	$[5, 8]$	279936
$\mathrm{IC1}-31X$	30	10	400	400	30	10	1000	100	0.5	6	$[5, 9]$	7812
$\mathrm{IC1}-32X$	30	10	200	200	30	10	500	500	0.4	6	$[5, 9]$	7812
$\mathrm{IC1}-33X$	30	10	300	200	30	10	300	500	0.5	6	$[5, 9]$	7812
$\mathrm{IC1}-34X$	30	10	300	300	30	10	500	500	0.5	6	$[5, 9]$	7812
$\mathrm{IC1}-35X$	30	10	400	300	30	10	500	500	0.6	6	$[5, 9]$	7812
$\mathrm{IC1}-36X$	30	10	300	300	30	10	500	500	0.8	6	$[5, 9]$	7812
$\mathrm{IC1}-37X$	30	10	200	300	30	10	1000	500	0.7	7	$[5, 10]$	279936
$\mathrm{IC1}-38X$	30	10	300	200	30	10	500	500	0.7	7	$[5, 10]$	279936
$\mathrm{IC1}-39X$	30	10	400	200	30	10	300	500	0.7	7	$[5, 10]$	279936
$\mathrm{IC1}-40X$	30	10	400	400	30	10	1000	500	0.3	7	$[5, 10]$	279936

Table 2: Characteristics of instances in Testbed 2, e.g., instance $\text{IC2} - 1S$ has $a_1 = n_1 = 400$, $\text{IC2} - 1L$ has $a_1 = n_1 = 1000$.

$\text{IC2} - KX$	$\text{IC2} - KS$				$\text{IC2} - KL$				Common parameters			
	a_1	a_2	n_1	n_2	a_1	a_2	n_1	n_2	v	m_2	$h(\omega)$	Scenarios
$\text{IC2} - 1X$	400	400	400	400	1000	100	1000	100	0.5	6	$[5, 9]$	7812
$\text{IC2} - 2X$	200	200	200	200	500	500	500	500	0.5	6	$[5, 9]$	7812
$\text{IC2} - 3X$	300	200	300	200	300	500	300	500	0.5	6	$[5, 9]$	7812
$\text{IC2} - 4X$	300	300	300	300	500	500	500	500	0.5	6	$[5, 9]$	7812
$\text{IC2} - 5X$	400	300	400	300	500	500	500	500	0.5	6	$[5, 9]$	7812
$\text{IC2} - 6X$	300	300	300	300	500	500	500	500	0.8	6	$[5, 9]$	7812
$\text{IC2} - 7X$	200	300	200	300	1000	500	1000	500	0.7	7	$[5, 10]$	279936
$\text{IC2} - 8X$	300	200	300	200	500	500	500	500	0.7	7	$[5, 10]$	279936
$\text{IC2} - 9X$	400	200	400	200	300	500	300	500	0.7	7	$[5, 10]$	279936
$\text{IC2} - 10X$	400	400	400	400	1000	500	1000	500	0.3	7	$[5, 10]$	279936

2. The Exact-Superadditive Algorithm does not require any special structure of the quadratic objective function other than superadditivity. That is, parameters a_1 and a_2 can be as large as n_1 and n_2. We test the performance of the Exact-Superadditive Algorithm on these type of instances by using small instances of Testbed 2 in Table 2.

3. The size of the instances, i.e., the number of variables, that can be solved efficiently using the Sparse-Enumeration Algorithm is much larger than that of the Exact-Superadditive Algorithm as long as a_1 and a_2 do not exceed 30. We demonstrate this by testing the Sparse-Enumeration Algorithm on large instances of Testbed 1.

In Section 7.1.2 we compare performances of the Exact-Superadditive and the Diagonal-Q Algorithms on small instances of Testbed 2, where we delete off-diagonal elements of the quadratic objective function to obtain diagonal instances. The size of the instances, i.e., the number of variables, that can be solved efficiently using the Diagonal-Q Algorithm is much larger than that of the Exact-Superadditive Algorithm. We demonstrate this by testing the Diagonal-Q Algorithm on large instances of Testbed 2, again by deleting the off-diagonal elements of the quadratic objective function.

7.1.1 Experiments with instances that have sparse quadratic matrices

We run the Exact-Superadditive and the Sparse-Enumeration Algorithms for five hours on small instances of Testbed 1 and report the number iterations as well as the total solution time required for constructing the value function in each stage in Table 3. If the computer memory is insufficient for an instance, we indicate this with a "-" mark in the associated time column.

The Sparse-Enumeration Algorithm often outperforms the Exact-Superadditive Algorithm in the first (second) stage when $a_1(a_2)$ is small such as 10, see Table 3. However, if we increase $a_1(a_2)$ up to 30, then the Exact-Superadditive Algorithm is often faster than the Sparse-Enumeration Algorithm. This result is intuitive because in any stage the number of iterations of the Sparse-Enumeration Algorithm grows exponentially with $a_1(a_2)$; whereas there is no direct relation between these parameters and the running time of the Exact-Superadditive Algorithm.

We further test the Exact-Superadditive Algorithm on small instances of Testbed 2 in which a_1 and a_2 parameters increase up to 400. We repeat this experiment three times using randomly generated instances and report the average values in Table 4. Note that the Sparse-Enumeration Algorithm takes longer than five hours to solve any one of these instances.

Finally we test the Sparse-Enumeration Algorithm on large instances of Testbed 1 in which the number of variables n_1 and n_2 increase up to 1000. The goal of this test is to demonstrate that the size of the instances (i.e., the number of variables) that can be solved efficiently using the Sparse-Enumeration Algorithm is much larger than that of the Exact-Superadditive Algorithm. We present the computational results in Table 5. Note that the Exact-Superadditive Algorithm takes longer than five hours to solve any one of these instances.

Considering the running times of the Sparse-Enumeration Algorithm on large and small instances of Testbed 1 in Tables 3 and 5, we note that this algorithm is relatively insensitive to the increase in the number of variables.

7.1.2 Experiments with instances that have diagonal quadratic matrices

In this section we test the Exact-Superadditive and Diagonal-Q Algorithms. To obtain diagonal instances we simply delete the off-diagonal elements of the quadratic objective function. We first run both algorithms three times on randomly generated small instances of Testbed 2. We present the average computational results in Table 6.

In Table 6 the Diagonal-Q Algorithm is often faster than the Exact-Superadditive Algorithm, which is expected as the this algorithm is specially

Table 3: Evaluating the value function of small instances in Testbed 1 using the Exact-Superadditive and the Sparse-Enumeration Algorithms[*]

IC1 − KS	First stage				Second stage			
	Exact-Superadditive		Sparse-Enumeration		Exact-Superadditive		Sparse-Enumeration	
	Iters	Time	Iters	Time	Iters	Time	Iters	Time
IC1 − 1S	724	10530.2	35	2.4	87	6.9	11	3.5
IC1 − 2S	168	245.3	37	1.2	1	<1 s	11	2.1
IC1 − 3S	771	1595.1	70	3.4	75	1.6	11	2.8
IC1 − 4S	599	1621.8	82	2.1	32	1.8	11	3.8
IC1 − 5S	-	>5 hrs	51	5.2	14	1.2	11	3.7
IC1 − 6S	2191	131.8	14	1.5	95	1.9	11	2.5
IC1 − 7S	7168	217.6	25	1.7	496	34.0	11	39.0
IC1 − 8S	7596	380.0	11	2.7	304	29.5	12	30.2
IC1 − 9S	10355	3622.9	19	3.0	321	35.5	11	28.8
IC1 − 10S	-	>5 hrs	31	8.6	42	17.3	19	91.7
IC1 − 11S	-	>5 hrs	24	1.0	26	2.2	16	4.1
IC1 − 12S	420	191.4	89	1.9	28	1.2	31	313.3
IC1 − 13S	445	6647.2	56	3.1	60	1.6	31	308.8
IC1 − 14S	534	1141.4	48	2.4	69	3.3	31	309.3
IC1 − 15S	-	>5 hrs	54	3.1	24	<1 s	31	309.0
IC1 − 16S	2493	259.1	16	<1 s	124	2.7	31	309.0
IC1 − 17S	6424	186.9	18	1.1	377	30.2	31	381.0
IC1 − 18S	11899	994.4	11	1.7	324	31.1	32	562.1
IC1 − 19S	10974	3186.5	17	3.5	203	23.9	31	379.6
IC1 − 20S	-	>5 hrs	126	22.0	66	437.9	138	5818.2
IC1 − 21S	-	>5 hrs	87	3.8	58	3.8	16	1.3
IC1 − 22S	1126	461.1	52	2.3	46	<1 s	16	<1 s
IC1 − 23S	1889	2909.4	44	1.7	40	<1 s	16	1.0
IC1 − 24S	209	2750.9	51	2.4	17	<1 s	16	1.0
IC1 − 25S	-	>5 hrs	292	15.0	17	<1 s	16	1.2
IC1 − 26S	2158	248.6	38	1.2	79	1.3	16	1.0
IC1 − 27S	6917	192.1	67	2.8	300	11.2	16	11.1
IC1 − 28S	10700	1575.4	31	3.8	190	7.2	16	8.2
IC1 − 29S	8397	2118.8	30	3.1	176	6.6	16	9.4
IC1 − 30S	-	>5 hrs	104	11.6	42	17.3	16	13.4
IC1 − 31S	285	17991.9	298	1777.5	30	2.4	11	3.0
IC1 − 32S	1914	1304.1	380	1106.1	36	1.1	11	2.0
IC1 − 33S	1808	10378.0	583	1614.1	54	1.2	11	3.5
IC1 − 34S	2230	7665.2	352	2504.6	13	1.3	11	3.8
IC1 − 35S	5	498.4	242	1494.3	31	1.8	11	3.1
IC1 − 36S	1093	83.7	231	2088.2	105	1.6	11	3.1
IC1 − 37S	6283	146.3	147	414.3	438	35.4	11	37.2
IC1 − 38S	5581	484.5	176	1281.1	395	34.3	11	30.4
IC1 − 39S	15442	4878.4	77	371.7	267	23.4	11	29.0
IC1 − 40S	-	>5 hrs	1157	9751.3	53	397.2	21	84.4

[*]Time is in seconds.

Table 4: Evaluating the value function of small instances in Testbed 2 using the Exact-Superadditive Algorithm. Reported values are the average of three runs on randomly generated instances*

$\text{IC2} - KS$	First stage		Second stage	
	Iters	Time	Iters	Time
$\text{IC2} - 1S$[‡]	-	>14990.1	40	6.1
$\text{IC2} - 2S$	2770	2491.6	47	1.7
$\text{IC2} - 3S$	1971	9861.4	54	1.8
$\text{IC2} - 4S$[†]	-	>9407.4	48	3.8
$\text{IC2} - 5S$	-	>5 hrs	40	3.0
$\text{IC2} - 6S$	4077	602	100	3.1
$\text{IC2} - 7S$	9243	474.1	403	47.6
$\text{IC2} - 8S$	17285	6892.6	337	32.6
$\text{IC2} - 9S$	13333	7133.8	353	30.8
$\text{IC2} - 10S$	-	>5 hrs	35	274.6

*Time is in seconds.

†One of the runs took more than 5 hours to solve.

‡Two of the runs took more than 5 hours to solve.

Table 5: Evaluating the value function of large instances in Testbed 1 using the Sparse-Enumeration Algorithm*

$\text{IC1} - KL$	First stage		Second stage		$\text{IC1} - KL$	First stage		Second stage	
	Iters	Time	Iters	Time		Iters	Time	Iters	Time
$\text{IC1} - 1L$	82	2.5	11	1.4	$\text{IC1} - 21L$	25	1.3	16	<1 s
$\text{IC1} - 2L$	30	<1 s	11	2.9	$\text{IC1} - 22L$	137	3.1	16	<1 s
$\text{IC1} - 3L$	51	<1 s	11	2.9	$\text{IC1} - 23L$	61	<1 s	16	<1 s
$\text{IC1} - 4L$	33	<1 s	11	2.9	$\text{IC1} - 24L$	75	1.1	16	<1 s
$\text{IC1} - 5L$	45	1.4	11	3	$\text{IC1} - 25L$	264	3.9	16	<1 s
$\text{IC1} - 6L$	12	<1 s	11	2.3	$\text{IC1} - 26L$	34	<1 s	16	<1 s
$\text{IC1} - 7L$	25	4.3	11	45.4	$\text{IC1} - 27L$	58	5	16	13
$\text{IC1} - 8L$	27	2.4	11	46	$\text{IC1} - 28L$	38	2.4	16	14
$\text{IC1} - 9L$	11	1	11	46.5	$\text{IC1} - 29L$	42	1.5	16	12.9
$\text{IC1} - 10L$	159	25.9	31	87.8	$\text{IC1} - 30L$	148	20.8	16	11.6
$\text{IC1} - 11L$	35	2	16	1.5	$\text{IC1} - 31L$	332	1648.8	11	1.5
$\text{IC1} - 12L$	51	1.1	31	226.8	$\text{IC1} - 32L$	275	1038.5	11	2.9
$\text{IC1} - 13L$	52	<1 s	31	226.7	$\text{IC1} - 33L$	558	4893.6	11	3.1
$\text{IC1} - 14L$	54	1.1	31	226.7	$\text{IC1} - 34L$	292	853	11	3
$\text{IC1} - 15L$	15	<1 s	31	226.2	$\text{IC1} - 35L$	303	1097.5	11	3
$\text{IC1} - 16L$	21	<1 s	31	225.6	$\text{IC1} - 36L$	190	834.6	11	2.3
$\text{IC1} - 17L$	14	3.5	33	310.8	$\text{IC1} - 37L$	158	482.4	13	43.8
$\text{IC1} - 18L$	16	2.5	32	444.5	$\text{IC1} - 38L$	74	270.9	11	46.4
$\text{IC1} - 19L$	19	1.4	31	318	$\text{IC1} - 39L$	124	428.4	11	45.3
$\text{IC1} - 20L$	26	7.5	142	5104.1	$\text{IC1} - 40L$	839	4835	13	83.5

*Time is in seconds.

Table 6: Evaluating the value function of diagonal small instances in Testbed 2 using the Exact-Superadditive and Diagonal-Q Algorithms. Reported values are the average of three runs on randomly generated instances*

$\mathrm{IC2}-KS$	First stage				Second stage			
	Exact-Superadditive		Diagonal-Q		Exact-Superadditive		Diagonal-Q	
	Iters	Time	Iters	Time	Iters	Time	Iters	Time
$\mathrm{IC2}-1S$	274	4363.9	943	1.8	49	7.4	800	3.6
$\mathrm{IC2}-2S$	799	611.1	471	1.0	42	2.2	400	2.3
$\mathrm{IC2}-3S$	62	662.9	708	1.1	35	2.0	400	2.4
$\mathrm{IC2}-4S$	75	1755.8	703	1.1	27	3.0	600	3.1
$\mathrm{IC2}-5S$	-	>5 hrs	1124	2.3	54	4.9	600	3.0
$\mathrm{IC2}-6S$	2500	381.0	626	1.0	49	2.8	600	2.5
$\mathrm{IC2}-7S$	6329	373.2	412	1.1	409	65.4	613	37.9
$\mathrm{IC2}-8S$	9422	2822.7	633	2.1	346	41.1	407	26.7
$\mathrm{IC2}-9S$	7487	3661.5	836	3.0	308	31.7	408	27.9
$\mathrm{IC2}-10S$	23	13308.4	1072	4.1	15	49.3	941	78.6

*Time is in seconds.

designed for the diagonal case. Recall that these instances have at most seven rows in the second stage. We expect that the superiority of the Diagonal-Q Algorithm over the Exact-Superadditive Algorithm would end as the number of second-stage constraints increases. This is because $|\mathbf{B}|$ grows exponentially as the number of rows increases, and from Proposition 5.4 the worst case running time estimation of the Diagonal-Q Algorithm is $O(n\mu_{\max}|\mathbf{B}|)$.

Finally we test the Diagonal-Q Algorithm on large instances of Testbed 2. The goal of this test is to demonstrate that the size of diagonal SQIP instances (i.e., the number of variables) that can be solved efficiently using the Diagonal-Q Algorithm is much larger than that of the Exact-Superadditive Algorithm. We run the Diagonal-Q Algorithm three times on randomly generated instances and present the average computational results in Table 7. The Exact-Superadditive Algorithm takes more than five hours to solve for each instance reported in Table 7.

Recall that the Sparse-Enumeration Algorithm is developed for the case when the quadratic objective function can be written as a multiplication of small number of sparse linear functions; whereas the Diagonal-Q Algorithm is designed for the case when $z(\cdot)$ is superadditive with a diagonal quadratic matrix. Not surprisingly, in these specific cases, the Exact-Superadditive

Table 7: Evaluating the value function of diagonal large instances in Testbed 2 using the Diagonal-Q Algorithm. Reported values are the average of three runs on randomly generated instances*

IC2 − KL	First stage		Second stage	
	Iters	Time	Iters	Time
IC2 − 1L	2380	7.6	200	1.5
IC2 − 2L	1183	2.5	1000	4.1
IC2 − 3L	691	1.1	1000	4.0
IC2 − 4L	1149	2.5	1000	4.1
IC2 − 5L	1371	3.0	1000	4.2
IC2 − 6L	1047	2.4	1000	3.5
IC2 − 7L	2089	12.0	1018	52.5
IC2 − 8L	1049	4.3	1019	54.6
IC2 − 9L	625	2.1	1025	57.5
IC2 − 10L	2660	10.5	1183	87.1

*Time is in seconds.

Algorithm is slower than the associated specialized algorithms.

7.2 Finding the Optimal Tender

Table 8 presents the computational results for the second phase of the proposed solution framework. These experiments are conducted on large instances of Testbed 1 after computing the value functions in both stages. Note that performances of the algorithms presented in this section do not depend on the number of variables. Therefore, we do not repeat our experiments neither with small instances of Tesbed 1 nor with any instance from Testbed 2.

The time required for finding an optimal tender is recorded for the branch-and-bound algorithm (B&B), exhaustive search method (ES), and the minimal tender approach (MT). We also report the size of the minimal tender set $|\Theta|$, and the sizes of the first- and second-stage feasible right-hand side sets $|\mathbf{B}^1|$ and $|\mathbf{B}^2|$. We do not report the time required for computing the minimal tender set Θ, because once the value functions are constructed, it is very fast to check whether a given right-hand side vector is a minimal tender as the number of rows is at most seven.

Not surprisingly, B&B and MT methods are faster than the ES approach for all instances in Table 8. Note that the ES solution times of two instances are very close if the sizes of their feasible first-stage right-hand side sets $|\mathbf{B}^1|$ are equal to each other. This is because ES approach enumerates all right-

Table 8: Finding the optimal tender of large instances in Testbed 1

Instance	B&B	ES	MT	$\lvert\Theta\rvert$	$\lvert\mathbf{B}^1\rvert$	$\lvert\mathbf{B}^2\rvert$
$\mathrm{IC1}-1L$	3.547	41.327	0.093	59	46656	1000000
$\mathrm{IC1}-2L$	0.453	41.203	0.328	242	46656	1000000
$\mathrm{IC1}-3L$	3.828	41.249	0.500	448	46656	1000000
$\mathrm{IC1}-4L$	2.296	41.125	0.125	77	46656	1000000
$\mathrm{IC1}-5L$	0.766	41.328	0.860	794	46656	1000000
$\mathrm{IC1}-6L$	0.328	41.281	0.750	647	46656	1000000
$\mathrm{IC1}-7L$	44.344	23176.800	470.891	5664	279936	19487171
$\mathrm{IC1}-8L$	13.531	23345.400	164.688	1969	279936	19487171
$\mathrm{IC1}-9L$	52.687	23272.400	74.203	880	279936	19487171
$\mathrm{IC1}-10L$	293.563	23274.300	27.000	316	279936	19487171
$\mathrm{IC1}-11L$	2.796	41.234	0.547	526	46656	1000000
$\mathrm{IC1}-12L$	2.468	41.282	0.125	82	46656	1000000
$\mathrm{IC1}-13L$	1.000	41.157	0.422	342	46656	1000000
$\mathrm{IC1}-14L$	0.578	41.219	2.094	2057	46656	1000000
$\mathrm{IC1}-15L$	1.391	41.219	0.140	106	46656	1000000
$\mathrm{IC1}-16L$	0.141	41.251	1.438	1302	46656	1000000
$\mathrm{IC1}-17L$	39.172	23129.500	372.052	4486	279936	19487171
$\mathrm{IC1}-18L$	52.079	23218.800	136.971	1639	279936	19487171
$\mathrm{IC1}-19L$	11.157	23245.700	98.685	1176	279936	19487171
$\mathrm{IC1}-20L$	634.102	23231.700	10.531	124	279936	19487171
$\mathrm{IC1}-21L$	4.078	32.703	0.047	25	46656	262144
$\mathrm{IC1}-22L$	2.844	32.719	0.047	34	46656	262144
$\mathrm{IC1}-23L$	2.343	32.656	0.312	408	46656	262144
$\mathrm{IC1}-24L$	0.219	32.813	0.204	247	46656	262144
$\mathrm{IC1}-25L$	0.750	32.704	0.422	575	46656	262144
$\mathrm{IC1}-26L$	0.188	32.688	0.750	1012	46656	262144
$\mathrm{IC1}-27L$	11.453	11615.900	202.891	4841	279936	4782969
$\mathrm{IC1}-28L$	21.359	11584.900	38.703	918	279936	4782969
$\mathrm{IC1}-29L$	14.000	11586.500	63.000	1497	279936	4782969
$\mathrm{IC1}-30L$	1174.840	11611.600	2.297	51	279936	4782969
$\mathrm{IC1}-31L$	0.485	41.219	0.156	104	46656	1000000
$\mathrm{IC1}-32L$	0.828	41.235	0.250	206	46656	1000000
$\mathrm{IC1}-33L$	0.110	41.267	1.422	1358	46656	1000000
$\mathrm{IC1}-34L$	0.375	41.282	0.125	84	46656	1000000
$\mathrm{IC1}-35L$	0.157	41.485	0.844	735	46656	1000000
$\mathrm{IC1}-36L$	0.125	41.236	1.141	1016	46656	1000000
$\mathrm{IC1}-37L$	59.969	23034.800	346.192	4168	279936	19487171
$\mathrm{IC1}-38L$	74.235	23297.400	75.251	893	279936	19487171
$\mathrm{IC1}-39L$	14.469	23186.200	105.798	1265	279936	19487171
$\mathrm{IC1}-40L$	427.021	23262.600	1.828	20	279936	19487171

*Time is in seconds.

hand side vectors in $\mathbf{B}^1$. On the other hand, MT method first eliminates the first-stage right-hand sides that are not minimal tenders as they can not be an optimal tender by Theorem 6.2. Then it enumerates over all right-hand sides in the minimal tender set Θ.

There is no general conclusion we could draw in terms of the comparison between B&B and MT methods from the computational results in Table 8. MT method tends to outperform B&B method when $|\Theta|$ is small such as in $\text{IC1} - 10L$, $\text{IC1} - 20L$, $\text{IC1} - 30L$, and $\text{IC1} - 40L$ instances. On the other hand, B&B method tends to outperform MT method as $|\Theta|$ gets larger. This result is intuitive, since the solution time of MT method increases linearly by $|\Theta|$; whereas there is no direct relation between the solution time of B&B method and $|\Theta|$.

The overall computational results show that our approach is relatively insensitive to the number of decision variables in both stages but sensitive to the number of second-stage constraints and the numbers of feasible right-hand sides in $\mathbf{B}^1$ and $\mathbf{B}^2$. We note that the portion of the total running time spent in the first and second phase varies depending on the algorithms used in those phases.

7.3 Observations from computational experiments

The largest QIPs that have been solved so far are diagonal instances and they have no more than 2000 columns and 2000 rows [62]. Extensive forms of the (possibly indefinite) SQIP instances (3) that we can solve are hundreds of orders of magnitude larger than those instances solved previously in the literature.

Usually integer programs with quadratic objective are substantially harder to solve than their linear objective counterparts. For instance, the linear assignment problem is polynomially solvable; whereas the quadratic assignment problem is NP-hard [17]. However, extensive forms of the SQIP instances (3) that we solve in this paper have the similar order of magnitude as those of Kong et al. [43], which are the largest stochastic linear integer programs solved in the literature. From this observation we conclude that our proposed two-phase solution framework overcomes the difficulties arising due to (possibly indefinite) quadratic objective functions for the class of problems considered in this paper.

8 Concluding Remarks

We present an algorithmic framework for a class of two-stage stochastic quadratic integer programs where the uncertainty only appears in the second-stage right-hand sides. The main contribution of the paper is two fold. First, we derive some theoretical properties of the value functions of QIPs. Second, we use these properties as well as superadditivity to develop efficient algorithms for computing the value functions of QIPs. We then apply a dual reformulation and resort to a generic global branch-and-bound algorithm, full enumeration, or a level-set approach to find a global optimal solution.

The Exact-Superadditive Algorithm presented in Section 5.1 provides the flexibility for improvements that would be interesting for further investigation. We use a dynamic programming algorithm to solve the quadratic integer programs arising in Step 1, which limits the number of variables that can be handled. Various objectives regarding the computational preference between solving quadratic integer programs and applying superadditive dual properties will lead to different procedural selections.

The major limitation of our two-phase solution approach is the explicit storage of value functions in computer memory. This is why our computations are based on instances that have tremendous number of columns and scenarios but relatively few rows. One approach to overcome this limitation is to seek more efficient ways to store value functions, such as using generating functions [48]. Another approach is to incorporate a global branch-and-bound scheme to calculate the solution on a subset of right-hand sides so that only a portion of the value function needs to be stored at any time.

References

[1] W. P. Adams and R. Forrester. Linear forms of nonlinear expressions: New insights on old ideas. *Operations Research Letters*, 35(4):510–518, 2007.

[2] W. P. Adams, R. Forrester, and F. Glover. Comparisons and enhancement strategies for linearizing mixed 0-1 quadratic programs. *Discrete Optimization*, 1(2):99–120, 2004.

[3] W. P. Adams and H. D. Sherali. A tight linearization and an algorithm for zero-one quadratic programming problems. *Management Science*, 32(10):1274–1290, 1986.

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

[5] S. C. Agrawal. On integer solutions to quadratic programs by a branch-and-bound technique. *Trabajos de Estadìstica y de Investigaciòn Operativa*, 25(1–2):65–70, 1974.

[6] S. C. Agrawal. On mixed-integer quadratic programs. *Naval Research Logistics Quarterly*, 21(2):289–297, 1974.

[7] S. C. Agrawal. An alternative method on integer solutions to quadratic programs by a branch-and-bound technique. *Trabajos de Estadìstica y de Investigaciòn Operativa*, 27(1–3):185–192, 1976.

[8] S. Ahmed and R. Garcia. Dynamic capacity acquisition and assignment under uncertainty. *Annals of Operational Research*, 124(1-4):267–283, 2003.

[9] S. Ahmed and A. Shapiro. The sample average approximation method for stochastic programs with integer recourse. 2002. ISyE Technical Report. Available from http://www2.isye.gatech.edu/˜sahmed/publications.html. Retrieved September 8, 2009.

[10] S. Ahmed, M. Tawarmalani, and N. V. Sahinidis. A finite branch and bound algorithm for two-stage stochastic integer programs. *Mathematical Programming*, 100(2):355–377, 2004.

[11] F. A. Al-Khayyal and J. E. Falk. Jointly constrained biconvex programming. *Mathematics of Opertions Research*, 8(2):273–286, 1983.

[12] F. A. Al-Khayyal and C. Larsen. Global optimization of a quadratic function subject to a bounded mixed integer constraint set. *Annals of Opertions Research*, 25(1–4):169–180, 1990.

[13] E. Balas. Duality in discrete programming II: The quadratic case. *Management Science*, 16(1):14–32, 1969.

[14] B. Bank and R. Hansel. Stability of mixed-integer quadratic programming problems. *Mathematical Programming Study*, 21:1–17, 1984.

[15] C. E. Blair and R. G. Jeroslow. The value function of an integer program. *Mathematical Programming*, 23(1):237–273, 1982.

[16] K. M. Bretthauer and B. Shetty. The nonlinear resource allocation problem. *Operations Research*, 43(4):670–683, 1995.

[17] R. Burkard, M. Dell'Amico, and S. Martello. *Assignment Problems.* Society for Industrial and Applied Mathematics, Philadelphia, 2009.

[18] A. Capara, D. Pisinger, and P. Toth. Exact solution of the quadratic knapsack problem. *INFORMS Journal on Computing*, 11(2):125–137, 1999.

[19] C. C. Carøe and R. Schultz. Dual decomposition in stochastic integer programming. *Operations Reseach Letters*, 24(1–2):37–45, 1999.

[20] C. C. Carøe and J. Tind. L-shaped decomposition of two-stage stochastic programs with integer recourse. *Mathematical Programming*, 83(1–3):451–464, 1998.

[21] W. Cook, A. M. H. Gerards, A. Schrijver, and E. Tardos. Sensitivity results in integer linear programming. *Mathematical Programming*, 34(3):251–264, 1986.

[22] J. W. Daniel. Stability of the solution of definite quadratic programs. *Mathematical Programming*, 5(1):41–53, 1973.

[23] V. Dua, K. P. Papalexandri, and E. N. Pistikopoulos. Global optimization issues in multiparametric continuous and mixed-integer optimization problems. *Journal of Global Optimization*, 30(1):59–89, 2004.

[24] B. C. Eaves. On quadratic programming. *Management Science*, 17(11):698–711, 1971.

[25] S. S. Erenguc and H. P. Benson. An algorithm for indefinite quadratic integer programming. *INFORMS Journal on Computing*, 11(2):125–137, 1999.

[26] G. Gallo, P. L. Hammer, and B. Simeone. Quadratic knapsack problems. *Mathematical Programming Study*, 12:132–149, 1980.

[27] P. C. Gilmore and R. E. Gomory. The theory and computation of knapsack functions. *Operations Research*, 14(6):1045–1074, 1966.

[28] F. Glover. Improved linear integer programming formulations of nonlinear integer problems. *Management Science*, 22(4):445–460, 1975.

[29] R. E. Gomory. Some polyhedra related to combinatorial problems. *Linear Algebra and its Applications*, 2(4):451–558, 1969.

[30] F. Granot and J. Skorin-Kapov. Some proximity and sensitivity results in quadratic integer programming. *Mathematical Programming*, 47(2):259–268, 1990.

[31] W. K. Klein Haneveld and M. H. van der Vlerk. Stochastic integer programming: General models and algorithms. *Annals of Operations Research*, 85(1):39–57, 1999.

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

[33] C. Helmberg, F. Rendl, and R. Weismantel. A semidefinite programming approach to the quadratic knapsack problem. *Journal of Combinatorial Optimization*, 4(2):197–215, 2000.

[34] R. Horst and H. Tuy. *Global Optimization: Deterministic Approaches, 3rd edition.* Springer-Verlag, Berlin, 1996.

[35] H.-X. Huang, P. M. Pardalos, and O. A. Prokopyev. Lower bound improvement and forcing rule for quadratic binary programming. *Computational Optimization and Applications*, 33(2–3):187–208, 2006.

[36] ILOG. Cplex, 2009. Available from http://www.ilog.com/products/cplex/. Retrieved August 15, 2009.

[37] E. L. Johnson. Cyclic groups, cutting planes and shortest paths. pages 185–211, 1973. In: Hu, T.C., Robinson, S.M. (eds.) Mathematical Programming. Academic Press, NewYork, NY.

[38] E. L. Johnson. *Integer Programming: Facets, Subadditivity, and Duality for Group and Semi-Group Problems.* SIAM Publications, Philadelphia, PA, 1980.

[39] E. L. Johnson. Subadditive lifting methods for partitioning and knapsack problems. *Journal of Algorithms*, 1(1):75–96, 1980.

[40] E. L. Johnson. Characterization of facets for multiple right-hand choice linear programs. *Mathematical Programming Study*, 14, 1981.

[41] J. Kalvenes, J. Kennington, and E. V. Olinick. Base station location and service assignment in W-CDMA networks. *INFORMS Journal on Computing*, 18(3), 2006.

[42] A. J. Kleywegt, A. Shapiro, and T. Homem-De-Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal of Optimization*, 12(2):479–502, 2001.

[43] N. Kong, A. J. Schaefer, and B. Hunsaker. Two-stage integer programs with stochastic right-hand sides: a superadditive dual approach. *Mathematical Programming*, 108(2):275–296, 2006.

[44] G. Laporte and F. V. Louveaux. The integer L-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters*, 13(3):133–142, 1993.

[45] E. L. Lawler. The quadratic assignment problem. *Management Science*, 9(4):586–599, 1963.

[46] R. Lazimy. Mixed-integer quadratic programming. *Mathematical Programming*, 22(1):332–349, 1982.

[47] R. Lazimy. Improved algorithm for mixed-integer quadratic programs and a computational study. *Mathematical Programming*, 32(1):110–113, 1985.

[48] J. A. De Loera, D. Haws, R. Hemmecke, P. Huggins, B. Strumfels, and R. Yoshida. Short rational functions for toric algebra and applications. *Journal of Symbolic Computation*, 38(2):959–973, 2004.

[49] E. M. Loiola, N. M. M. de Abreu, P. O. Boaventura-Netto, P. Hahn, and T. Querido. A survey for the quadratic assignment problem. *European Journal of Operational Research*, 176(2):657–690, 2007.

[50] J. C. T. Mao and B. A. Wallingford. An extension of Lawler and Bell's method of discrete optimization with examples from capital budgeting. *Management Science*, 15(2):851–860, 1969.

[51] R. D. McBride and J. S. Yormark. An implicit enumeration algorithm for quadratic integer programming. *Management Science*, 26(3):282–296, 1980.

[52] R. D. McBride and J. S. Yormark. Finding all solutions for a class of parametric quadratic integer programming problems. *Management Science*, 26(8):784–795, 1980.

[53] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization.* Wiley, NewYork, NY, 1988.

[54] V. I. Norkin, Y. M. Ermoliev, and A. Ruszczyński. On optimal allocation of indivisibles under uncertainty. *Operations Research*, 46(3):381–395, 1998.

[55] L. Ntaimo. Disjunctive decomposition for two-stage stochastic mixed-binary programs with random recourse. 2009. To appear in Operations Research.

[56] L. Ntaimo and S. Sen. The million-variable 'march' for stochastic combinatorial optimization. *Journal of Global Optimization*, 32(3):385–400, 2004.

[57] M. Oral and O. Kettani. A linearization procedure for quadratic and cubic mixed-integer problems. *Operations Research*, 40(S1):109–116, 1990.

[58] M. Oral and O. Kettani. Reformulating nonlinear combinatorial optimization problems for higher computational efficiency. *European Journal of Operational Research*, 58(2):236–249, 1992.

[59] 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.

[60] J.-C. Picard and H. D. Ratliff. A cut approach to a class of quadratic integer programming problems. *Networks*, 10(4):363–370, 1980.

[61] D. Pisinger. The quadratic knapsack problem–a survey. *Discrete Applied Mathematics*, 155(5):623–648, 2006.

[62] D. Quadri, E. Soutif, and P. Tolla. Exact solution method to solve large scale integer quadratic multidimensional knapsack problems. *Journal of Combinatorial Optimization*, 17(2):157–167, 2009.

[63] R. Schultz. On structure and stability in stochastic programs with random technology matrix and complete integer recourse. *Mathematical Programming*, 70(1):73–89, 1995.

[64] R. Schultz. Stochastic programming with integer variables. *Mathematical Programming*, 97(1–2):285–309, 2003.

[65] R. Schultz, L. Stougie, and M. H. van der Vlerk. Solving stochastic programs with integer recourse by enumeration: A framework using Gröbner basis reductions. *Mathematical Programming*, 83(1–3):229–252, 1998.

[66] S. Sen and J. L. Higle. The C^3 theorem and a D^2 algorithm for large scale stochastic mixed-integer programming: Set convexification. *Mathematical Programming*, 104(1):1 – 20, 2005.

[67] S. Sen and H. D. Sherali. Decomposition with branch-and-cut approaches for two-stage stochastic mixed-integer programming. *Mathematical Programming*, 106(2):203 – 223, 2006.

[68] A. Shapiro and T. Homem de Mello. A simulation-based approach to two-stage stochastic programming with recourse. *Mathematical Programming*, 81(3):301–325, 1998.

[69] H. D. Sherali and B. M. P. Fracticelli. A modification of Benders' decomposition algorithm for discrete subproblems: an approach for stochastic programs with integer recourse. *Journal of Global Optimization*, 22(1–4):319 – 342, 2002.

[70] J. C. Smith, A. Schaefer, and J. W. Yen. A stochastic integer programming approach to solving a synchronous optical network ring design problem. *Networks*, 44(1):12–26, 2004.

[71] L. V. Snyder, M. S. Daskin, and C.-P. Teo. The stochastic location model with risk pooling. *European Journal of Operational Research*, 179(3):1221–1238, 2007.

[72] L. Stougie. Design and analysis of algorithms for stochastic integer programming, 1987. Ph.D. dissertation, Center for Mathematics and Computer Science, Amsterdam.

[73] N. V. Thoai. Global optimization techniques for solving the general quadratic integer programming problem. *Computational Optimization and Applications*, 10(2):149–163, 1998.

[74] R. Van Slyke and R. J-B. Wets. L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, 17(4):638–663, 1969.

[75] L. G. Watters. Reduction of integer polynomial problems to zero-one linear programming problems. *Operations Research*, 15(6):1171–1174, 1967.

[76] R. J-B. Wets. Stochastic programs with fixed recourse: The equivalent deterministic problem. *SIAM Review*, 16:309–339, 1974.

[77] L. A. Wolsey. Integer programming duality: Price functions and sensitivity analysis. *Mathematical Programming*, 20(2):173–195, 1981.