# Relaxations of the Cutting Plane Method for Quadratic Semi-Infinite Programming

Shu-Cherng Fang*
Operations Research and Industrial Engineering
North Carolina State University
Raleigh, NC 27695-7906, U.S.A.
Email: fang@eos.ncsu.edu

Chih-Jen Lin†
Department of Computer Science and Information Engineering
National Taiwan University
Taipei, Taiwan
Email: cjlin@csie.ntu.edu.tw

Soon-Yi Wu
Institute of Applied Mathematics
National Cheng-Kung University
Tainan 700, Taiwan
Email: soonyi@mail.ncku.edu.tw

**Abstract**

This paper studies the cutting plane method for solving quadratic semi-infinite programming problems. Several relaxation techniques and their combinations are proposed and discussed. A flexible convergence proof is provided to cover different settings. The implementation issues are addressed with some numerical experiments to illustrate the computational behavior of each different combination scheme.

*Partially supported by the the North Carolina Supercomputing Center, Cray Research Grant, and the National Textile Center Research Grant.

†Part of this work was done while the author worked for the Mathematics and Computer Science Division, Argonne National Laboratory, supported by U.S. Department of Energy, under Contract W-31-109-Eng-38.

# 1 Introduction

Consider the following (convex) quadratic semi-infinite programming problem:

$$(QSIP) \qquad \begin{aligned} \min \sum_{j=1}^{n} c_j x_j + \frac{1}{2}\mathbf{x}^T Q \mathbf{x} & \\ \text{s.t.} \sum_{j=1}^{n} x_j f_j(t) \ &\geq\ g(t),\ \forall t \in T \qquad (1) \\ x_j \ &\geq 0, \qquad j = 1, \cdots, n \qquad (2) \end{aligned}$$

where $T$ is a compact metric space with an infinite cardinality, $Q$ is an $n \times n$ symmetric positive semidefinite matrix, $f_j$, $j = 1, \cdots, n$, and $g$ are real valued continuous functions defined on $T$.

Its dual problem can be formulated in the following form:

$$(DQSIP) \qquad \begin{aligned} \max \int_T g(t) d\mu(t) - \frac{1}{2}\mathbf{x}^T Q \mathbf{x} & \\ \text{s.t.} \int_T f_j(t) d\mu(t) - (Q\mathbf{x})_j \ &\leq\ c_j, j = 1, \cdots, n \qquad (3) \\ \mu \ &\in\ M^+(T) \qquad (4) \end{aligned}$$

where $(Q\mathbf{x})_j$ is the $j$th component of the vector $Q\mathbf{x}$ and $M^+(T)$ is the space of all non-negative bounded regular Borel measures on $T$.

Under some regularity conditions, it can be shown that there is no duality gap between $(QSIP)$ and $(DQSIP)$. Also notice that when $Q$ becomes the zero matrix, the problem reduces to a linear semi-infinite programming problem $(LSIP)$ with a corresponding dual problem $(DLSIP)$.

Hettich and Kortanek [10] provided a general survey of semi-infinite programming in 1993. A more focused review on the numerical methods for semi-infinite programming can be found in the recent work of Rememtsen and Görner [18]. Survey shows that the cutting plane method [1, 5, 7] is one of the major techniques used for solving linear, quadratic, and convex semi-infinite programming problems. When the cutting plane method is applied to solve a quadratic semi-infinite programming problem, basically, it solves a sequence of finite dimensional quadratic programs and shows that the corresponding solution sequence converges to an optimal solution of $(QSIP)$.

To be more precise, in the $k$th iteration, let $T_k = \{t_1, t_2, \cdots, t_k\} \subset T$ and consider the following quadratic subproblem:

$$
(QP_k) \qquad \min \sum_{j=1}^{n} c_j x_j + \frac{1}{2}\mathbf{x}^T Q \mathbf{x}
$$

$$
\text{s.t.} \sum_{j=1}^{n} x_j f_j(t_i) \ \geq \ g(t_i), \ i = 1, 2, \cdots, k \tag{5}
$$

$$
x_j \ \geq 0, \qquad j = 1, 2, \cdots, n. \tag{6}
$$

We solve $(QP_k)$ for an optimal solution $\mathbf{x}^k = (x_1^k, x_2^k, \cdots, x_n^k)^T$ and define

$$
\phi_k(t) = \sum_{j=1}^{n} f_j(t) x_j^k - g(t). \tag{7}
$$

Then we find an optimizer:

$$
t_{k+1} \in \ arg \min_{t \in T} \phi_k(t). \tag{8}
$$

If $\phi_k(t_{k+1}) = \sum_{j=1}^{n} f_j(t_{k+1}) x_j^k - g(t_{k+1}) \geq 0$, then $\mathbf{x}^k$ must be an optimal solution to $(QSIP)$, because the feasible domain of $(QSIP)$ is contained in that of $(QP_k)$. Otherwise, we let $T_{k+1} = T_k \bigcup \{t_{k+1}\}$ to construct $(QP_{k+1})$ and repeat the process. The convergence proof of $\{\mathbf{x}^1, \mathbf{x}^2, \cdots, \mathbf{x}^k, \cdots\}$ to an optimal solution of $(QSIP)$ can be obtained by using the compactness of $T$ and the continuity of $f$ and $g$.

In this approach, one constraint is added at a time and the major computational work in each iteration involves (a) solving a quadratic program $(QP_k)$ and (b) finding a global minimizer $t_{k+1}$ of $\phi_k(t)$. To reduce the computational burden, several relaxation techniques have been investigated for the improvement of the original algorithm:

(i) Using the information obtained in the $k$th iteration to help solving $(QP_{k+1})$.

(ii) Relaxing the work of solving $(QP_k)$.

(iii) Relaxing the work of finding a global minimizer $t_{k+1}$ of $\phi_k(t)$.

(iv) Dropping unnecessary constraints while a new constraint is added in each iteration.

(v) Exploiting the special structure of a given problem.

Intuitively, each of the above relaxation techniques seems to be useful and a combination of several techniques may produce even better results. However, in reality, the combination of relaxation techniques has seldom been studied, because these relaxation techniques were designed under very different settings and their numerical behaviors are quite different. It is not a simple task to design an effective computational scheme with combined relaxations and come up with a convergence proof.

For example, when the traditional simplex method is employed as the solution engine, for issue (i), the whole sequence of $(QP_k)$ can be reformulated in such a way that the cutting plane method is implicitly performed by executing only one simplex procedure [5, 11]. On the other hand, when the interior-point method is chosen to be the solution engine, explicitly solving a sequence of subproblems becomes unavoidable. For issue (i), a dual affine scaling based cutting plane method was proposed [15], and for issue (ii), a primal-dual interior-point based method was considered [21].

When the dimensionality of the compact metric space $T$ becomes high, finding a global minimizer of the function $\phi_k(t)$ over $T$ could be extremely time-consuming, in particular, when $f_j(t)$ and $g(t)$ are highly nonlinear and non-convex. Several authors [8, 9, 6, 24] have tried to address issue (iii) by constructing subproblems $(QP_{k+1})$ without exactly solving the global minimization problems. The "one-phase" algorithm proposed by Hu [12] only requires that $\phi_k(t_{k+1}) < -\delta$ or $\phi_k(t_{k+1}) < \delta \min_t \phi_k(t)$. For the same purpose, the authors of [16] replaced the global minimization problem by multidimensional integration in their unconstrained convex programming approach.

For issue (iv), the first paper considering the possibility of dropping unnecessary constraints in each subprogram may be referred to Roleff [19]. But the proposed procedure lacks a convergence proof. Tang [22] filled in with a convergence proof for a special class of problems. A general convergence proof of dropping unnecessary constraints for linear semi-infinite programming can be found in [14].

Following the work of [13], the authors of [25] recently proposed a framework which allows the relaxation in finding the global minimizer of $\phi_k(t)$ and dropping unnecessary constraints for solving linear semi-infinite programming problems. In this paper we would like to further extend this framework for solving convex quadratic semi-infinite programming problems and study the effect of combining different relaxation techniques. In particular, we shall modify the framework to include relaxation techniques (ii), (iii) and (iv), accommodate the difference between linear and quadratic cases, present a convergence proof of the modified cutting plane method for the quadratic case (Section 2), show that our convergence proof is flexible enough for using the primal-dual infeasible interior-point algorithm as a solution engine (Section 3), and discuss how to combine different relaxation techniques with some numerical experiments (Section 4). Concluding remarks will be made in Section 5.

## 2 Relaxation Scheme and Convergence Proof

Let $T' = \{t_1, ..., t_m\}$ be a subset with $m$ elements in $T$. A convex quadratic program with $m$ explicit constraints induced by $T'$ is denoted as

$$(QP_{T'}) \qquad \min \sum_{j=1}^{n} c_j x_j + \frac{1}{2}\mathbf{x}^T Q\mathbf{x}$$

$$\text{s.t.} \sum_{j=1}^{n} x_j f_j(t_i) \geq g(t_i), \quad i = 1, 2, \cdots, m \tag{9}$$

$$x_j \geq 0, \qquad j = 1, 2, \cdots, n. \tag{10}$$

Its dual problem can be written as

$$(DQP_{T'}) \qquad \max -\frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \sum_{i=1}^{m} g(t_i) y_i$$

$$\text{s.t.} \sum_{i=1}^{m} f_j(t_i) y_i - (Q\mathbf{x})_j \leq c_j, \quad j = 1, 2, \cdots, n \tag{11}$$

$$y_i \geq 0, \quad i = 1, 2, \cdots, m. \tag{12}$$

For a primal solution $\mathbf{x}$, its primal objective value is denoted by $V(QP_{T'}(\mathbf{x}))$. Similarly, for a dual solution $(\mathbf{x}, \mathbf{y})$, its dual objective value is denoted by $V(DQP_{T'}(\mathbf{x}, \mathbf{y}))$. Notice that if both problems $(QP_{T'})$ and $(DQP_{T'})$ are solvable, when a primal-dual algorithm (no matter simplex-based or interior-based , e.g., see [3, 20]) is applied, usually a primal feasible solution $\mathbf{x}$ and a dual feasible solution $(\mathbf{x}, \mathbf{y})$ can be generated simultaneously.

Given that $\delta > 0$ is a prescribed small number, under the assumption that $(QP_{T_k})$ and its dual $(DQP_{T_k})$ are both solvable in each iteration, we now state a relaxed cutting plane method for solving convex quadratic semi-infinite programming problems as follows.

**Relaxation Scheme of Cutting Plane Method for QSIP**

Step 0 : Let $k = 1$, choose any $t_1^0 \in T$, $\Delta_1 > 0$, $\epsilon > 0$, set $T_1 = \{t_1^0\}$, and $m_0 = 0$.

Step 1 : Find an approximate (inexact) primal solution $\mathbf{x}^k$ for $(QP_{T_k})$ and an approximate (inexact) dual solution $(\mathbf{x}^k, \mathbf{y}^k)$ for $(DQP_{T_k})$ such that $V(QP_{T_k}(\mathbf{x}^k)) - V(DQP_{T_k}(\mathbf{x}^k, \mathbf{y}^k)) < \Delta_k$.

Define $\phi_k(t)$ according to Eq. (7).

Step 2 : Denote $\mathbf{y}^k = (y_1^k, \cdots, y_{m_{k-1}+1}^k)^T$.

Define a discrete measure $\mu_k$ on $T$ with

$$\mu_k(t) = \begin{cases} y_i^k (\geq 0) & \text{if } t = t_i^{k-1} \in T_k, \\ 0 & \text{if } t \in T \backslash T_k. \end{cases}$$

Set $E_k = \{t \in T_k \mid \mu_k(t) > 0\} = \{t_1^k, \cdots, t_{m_k}^k\}$.

Step 3 : Find any $t_{m_k+1}^k \in T$ such that $\phi_k(t_{m_k+1}^k) < -\delta$.

If such $t_{m_k+1}^k$ does not exist and $\Delta_k < \delta$, stop and output $\mathbf{x}^k$ as a solution.

If $k = 1$ or $V(DQP_{T_k}(\mathbf{x}^k, \mathbf{y}^k)) \geq V(DQP_{T_{k-1}}(\mathbf{x}^{k-1}, \mathbf{y}^{k-1}))$ but $t_{m_k+1}^k$ exists, set $T_{k+1} = E_k \cup \{t_{m_k+1}^k\}$; otherwise, set $\Delta_k = (1-\epsilon)\Delta_k$ and go to Step 1.

Step 4 : Update $k$ by $k+1$ and go to Step 1.

Notice that the proposed scheme is more general than the algorithm proposed in [25, Section 4]. In addition to the relaxation in finding a global minimizer and dropping unnecessary constraints in each iteration, the proposed scheme also permits the relaxation in solving subproblems $(QP_{T_k})$ and $(DQP_{T_k})$. In Step 3, since $\mathbf{x}^k$ and $\mathbf{y}^k$ are no longer (exact) optimal solutions, when $\Delta_k$ is not small enough, the algorithm is required to stay in the $k$th iteration to find finer solutions of the subproblems.

Also notice that, as long as the inexact solutions $\mathbf{x}^k$ and $(\mathbf{x}^k, \mathbf{y}^k)$ obtained in Step 1 are feasible to $(QP_{T_k})$ and $(DQP_{T_k})$, respectively, the following complementary slackness condition is satisfied:

$$
\begin{aligned}
0 &< \sum_{j=1}^{n} c_j x_j^k + (\mathbf{x}^k)^T Q(\mathbf{x}^k) - \sum_{i=1}^{m_{k-1}+1} g(t_i^{k-1}) y_i^k \\
&= \sum_{i=1}^{m_{k-1}+1} (\sum_{j=1}^{n} f_j(t_i^{k-1}) x_j^k - g(t_i^{k-1})) y_i^k + \sum_{j=1}^{n} (c_j + (Q\mathbf{x}^k)_j - \sum_{i=1}^{m_{k-1}+1} f_j(t_i^{k-1}) y_i^k) x_j^k \\
&\leq \Delta_k.
\end{aligned} \tag{13}
$$

In order to provide a convergence proof of the proposed relaxation algorithm, we need a few definitions and a technical lemma. Recalling the definition of $\phi_k(t)$, on the dual side, we define

$$
\phi_j^k \equiv \sum_{i=1}^{m_k} f_j(t_i^k)\mu_k(t_i^k) - (Q\mathbf{x}^k)_j - c_j, \quad \text{for } j = 1, \cdots, n. \tag{14}
$$

We also define

$$
s_j^k \equiv \sum_{i=1}^{n} f_i(t_j^k) x_i^{k+1} - g(t_j^k), \quad \text{for } \ j = 1, \cdots, m_k.
$$

With the above notations, we have the following technical lemma:

**Lemma 2.1** *Let* $\Delta_k' \equiv V(QP_{T_k}(\mathbf{x}^k)) - V(DQP_{T_k}(\mathbf{x}^k, \mathbf{y}^k))$ *and* $\mathbf{d}^k \equiv \mathbf{x}^{k+1} - \mathbf{x}^k$, *then*

$$
V(DQP_{T_{k+1}}(\mathbf{x}^{k+1}, \mathbf{y}^{k+1})) - V(DQP_{T_k}(\mathbf{x}^k, \mathbf{y}^k)) \tag{15}
$$

$$= \sum_{i=1}^{m_k} s_i^k \mu_k(t_i^k) - \sum_{j=1}^{n} \phi_j^k x_j^{k+1} + \frac{1}{2}(\mathbf{d}^k)^T Q \mathbf{d}^k - \Delta'_{k+1}.$$

**Proof**: By the definition of $\phi_j^k$, we have

$$\begin{aligned}
\sum_{j=1}^{n} \phi_j^k x_j^{k+1} &= \sum_{j=1}^{n}(\sum_{i=1}^{m_k} f_j(t_i^k)\mu_k(t_i^k) - (Q\mathbf{x}^k)_j - c_j)x_j^{k+1} \\
&= \sum_{i=1}^{m_k}(\sum_{j=1}^{n} f_j(t_i^k)x_j^{k+1})\mu_k(t_i^k) - (\mathbf{x}^{k+1})^T Q\mathbf{x}^k - \sum_{j=1}^{n} c_j x_j^{k+1} \\
&= \sum_{i=1}^{m_k}(s_i^k + g(t_i^k))\mu_k(t_i^k) - (\mathbf{x}^{k+1})^T Q\mathbf{x}^k - \sum_{j=1}^{n} c_j x_j^{k+1} \\
&= \sum_{i=1}^{m_k} s_i^k \mu_k(t_i^k) + \sum_{i=1}^{m_k} g(t_i^k)\mu_k(t_i^k) - (\mathbf{x}^{k+1})^T Q\mathbf{x}^k - \sum_{j=1}^{n} c_j x_j^{k+1}. \qquad (16)
\end{aligned}$$

Since $\mathbf{d}^k = \mathbf{x}^{k+1} - \mathbf{x}^k$, we have

$$\begin{aligned}
(\mathbf{x}^{k+1})^T Q\mathbf{x}^k &= \frac{1}{2}(\mathbf{x}^{k+1})^T Q\mathbf{x}^k + \frac{1}{2}(\mathbf{x}^{k+1})^T Q\mathbf{x}^k \\
&= \frac{1}{2}(\mathbf{x}^k + \mathbf{d}^k)^T Q\mathbf{x}^k + \frac{1}{2}(\mathbf{x}^{k+1})^T Q(\mathbf{x}^{k+1} - \mathbf{d}^k) \\
&= \frac{1}{2}(\mathbf{x}^k)^T Q\mathbf{x}^k + \frac{1}{2}(\mathbf{d}^k)^T Q\mathbf{x}^k + \frac{1}{2}(\mathbf{x}^{k+1})^T Q\mathbf{x}^{k+1} - \frac{1}{2}(\mathbf{x}^{k+1})^T Q\mathbf{d}^k. \quad (17)
\end{aligned}$$

Notice that $Q$ is symmetric and $(\mathbf{x}^{k+1})^T Q\mathbf{d}^k = (\mathbf{d}^k)^T Q\mathbf{x}^{k+1}$, we further have

$$\begin{aligned}
(\mathbf{x}^{k+1})^T Q\mathbf{x}^k &= \frac{1}{2}(\mathbf{x}^k)^T Q\mathbf{x}^k + \frac{1}{2}(\mathbf{x}^{k+1})^T Q\mathbf{x}^{k+1} + \frac{1}{2}(\mathbf{d}^k)^T Q\mathbf{x}^k - \frac{1}{2}(\mathbf{d}^k)^T Q\mathbf{x}^{k+1} \\
&= \frac{1}{2}(\mathbf{x}^k)^T Q\mathbf{x}^k + \frac{1}{2}(\mathbf{x}^{k+1})^T Q\mathbf{x}^{k+1} - \frac{1}{2}(\mathbf{d}^k)^T Q\mathbf{d}^k.
\end{aligned}$$

It follows from Eq. (16) that

$$\begin{aligned}
\sum_{j=1}^{n} \phi_j^k x_j^{k+1} &= \sum_{j=1}^{m_k} s_j^k \mu_k(t_i^k) + \sum_{j=1}^{m_k} g(t_i^k)\mu_k(t_i^k) - \frac{1}{2}(\mathbf{x}^k)^T Q\mathbf{x}^k - \frac{1}{2}(\mathbf{x}^{k+1})^T Q\mathbf{x}^{k+1} + \\
&\quad \frac{1}{2}(\mathbf{d}^k)^T Q\mathbf{d}^k - \sum_{j=1}^{n} c_j x_j^{k+1} \\
&= \sum_{j=1}^{m_k} s_j^k \mu_k(t_i^k) + V(DQP_{T_k}(\mathbf{x}^k, \mathbf{y}^k)) - \Delta'_{k+1} - V(DQP_{T_{k+1}}(\mathbf{x}^{k+1}, \mathbf{y}^{k+1})) + \\
&\quad \frac{1}{2}(\mathbf{d}^k)^T Q\mathbf{d}^k.
\end{aligned}$$

Thus we have

$$
\begin{aligned}
& V(DQP_{T_{k+1}}(\mathbf{x}^{k+1}, \mathbf{y}^{k+1})) - V(DQP_{T_k}(\mathbf{x}^k, \mathbf{y}^k)) \\
= & \sum_{j=1}^{m_k} s_j^k \mu_k(t_i^k) - \sum_{j=1}^{n} \phi_j^k x_j^{k+1} + \frac{1}{2} \mathbf{d}^{k^T} Q \mathbf{d}^k - \Delta'_{k+1}.
\end{aligned}
$$

$\Xi$

The above lemma is similar to the case in linear semi-infinite programming [25, Theorem 2.2] but different. First, an additional quadratic term $\frac{1}{2}(\mathbf{d}^k)^T Q \mathbf{d}^k$, where $\mathbf{d}^k = \mathbf{x}^{k+1} - \mathbf{x}^k$, appears. Second, because $\mathbf{x}^k$ and $\mathbf{y}^k$ are not optimal to $(QP_{T_k})$ and $(DQP_{T_k})$, respectively, an error term $\Delta'_k = V(QP_{T_k}(\mathbf{x}^k)) - V(DQP_{T_k}(\mathbf{x}^k, \mathbf{y}^k))$ is involved. With Lemma 2.1, we now provide a convergence proof of the proposed relaxation scheme.

Let us start with a simpler case with strictly convex quadratic semi-infinite programming problems.

**Theorem 2.2** *Given any $\delta > 0$, assume that*
*(A1) the set $\{\mathbf{x}^1, \mathbf{x}^2, \cdots\}$ is bounded,*
*(A2) there exists $\bar{\delta} > 0$ such that $\mu_{k+1}(t_{m_k+1}^k) \geq \bar{\delta}$, for each $k$.*
*Then the proposed scheme terminates in a finite number of iterations, if the matrix $Q$ in $(QSIP)$ is positive definite.*

**Proof**:

Suppose that the proposed scheme does not stop in a finite number of iterations. Step 3 implies that

$$V(DQP_{T_1}(\mathbf{x}^1, \mathbf{y}^1)) \leq V(DQP_{T_2}(\mathbf{x}^2, \mathbf{y}^2)) \leq \cdots \leq V(QSIP).$$

Consequently, $\lim_{r\to\infty} V(DQP_{T_r}(\mathbf{x}^r, \mathbf{y}^r)) = \alpha \leq V(QSIP)$, for some $\alpha$. We claim this is impossible.

By the assumption $(A1)$, the infinite sequences $\{\mathbf{x}^k\}$ and $\{\mathbf{d}^k\}$ are confined in a compact set $C$ in $R^n$. Hence there exists a subsequence $\{\mathbf{x}^{k_r}\}$ of $\{\mathbf{x}^k\}$ such that $\{\mathbf{x}^{k_r}\}$ converges to $\mathbf{x}^*$, the subsequence $\{\mathbf{d}^{k_r}\}$ converges to some point $\mathbf{d}^*$, and the subsequence $\{t_{m_{k_r}+1}^{k_r}\}$ converges to some point $t_*$ as $r \to \infty$. Let

$$\phi_*(t) \equiv \sum_{i=1}^{n} f_i(t) x_i^* - g(t). \tag{18}$$

Then $\phi_{k_r}(t_{m_{k_r}+1}^{k_r})$ converges to $\phi_*(t_*)$. Since $\phi_{k_r}(t_{m_{k_r}+1}^{k_r}) < -\delta$, for each $r$, we know $0 \neq \phi_*(t_*) \leq -\delta$.

Now let $\varepsilon \in (0, \delta)$ be an arbitrary number, we can find a large integer $N \in \{k_r\}_{r=1}^{\infty}$ such that

$$| V(DQP_{T_N}(\mathbf{x}^N, \mathbf{y}^N)) - \alpha | \leq \varepsilon^2, \|\mathbf{d}^N - \mathbf{d}^*\| \leq \varepsilon^2, \text{ and } | \phi_N(t^N_{m_N+1}) - \phi_*(t_*) | \leq \varepsilon^2. \quad (19)$$

By Lemma 2.1, we have

$$\begin{aligned} & |V(DQP_{T_{N+1}}(\mathbf{x}^{N+1}, \mathbf{y}^{N+1})) - V(DQP_{T_N}(\mathbf{x}^N, \mathbf{y}^N))| \\ = \; & |\sum_{j=1}^{m_N} s_j^N \mu_N(t_i^N) - \sum_{j=1}^{n} \phi_j^N x_j^{N+1} - \Delta'_{N+1} + \frac{1}{2}(\mathbf{d}^N)^T Q \mathbf{d}^N| \leq \varepsilon^2. \end{aligned} \quad (20)$$

Recall that $\phi_j^N \leq 0$, for $j = 1, 2, \cdots, n$, each term in the first summation sign of (20) is nonnegative, and each term in the second summation sign of (20) is nonpositive. It follows that

$$0 \leq \frac{1}{2}(\mathbf{d}^N)^T Q \mathbf{d}^N \leq \varepsilon^2 + \Delta'_{N+1} - \sum_{j=1}^{m_N} s_j^N \mu_N(t_i^N) + \sum_{j=1}^{n} \phi_j^N x_j^{N+1} \leq \varepsilon^2 + \Delta'_{N+1}$$

and hence

$$0 \leq |\frac{1}{2}(\mathbf{d}^N)^T Q \mathbf{d}^N| \leq \varepsilon^2 + \Delta'_{N+1}.$$

Thus $(\mathbf{d}^*)^T Q \mathbf{d}^* = 0$ which implies that $\mathbf{d}^* = 0$ when $Q$ is positive definite. Then from $\|\mathbf{d}^N - \mathbf{d}^*\| \leq \varepsilon^2$ of (13), we have

$$\|\mathbf{d}^N\| = \|\mathbf{x}^{N+1} - \mathbf{x}^N\| < \varepsilon^2. \quad (21)$$

By the assumption (A2) and (13), we further have

$$0 \leq \phi_{N+1}(t^N_{m_N+1}) < \frac{\Delta'_{N+1}}{\bar{\delta}}. \quad (22)$$

It follows from (21) and (22) that

$$\phi_N(t^N_{m_N+1}) \to 0, \text{ as } N \to \infty. \quad (23)$$

But $\phi_N(t^N_{m_N+1}) \to \phi_*(t_*) \neq 0$, as $N \to \infty$. Hence (23) cannot be true and we draw a contradiction. $\Xi$

The positive definiteness for matrix $Q$ is the key to conclude that $\mathbf{d}^* = 0$ in the above proof. When the matrix $Q$ is only positive semidefinite, we need some stronger assumptions for a convergence proof.

Again we need some more notations. Let $B_k = \{j_1^k, \cdots, j_{p_k}^k\}$ be an index set such that $x_j^k > 0$ if and only if $j \in B_k$. Also let $M_k$ be a $p_k \times m_k$ matrix with its $j$th row vector being

$$(f_j(t_1^k), \cdots, f_j(t_{m_k}^k)), \text{ for } j \in B_k. \quad (24)$$

Similar to Theorem 2.3 of [25], we have the following result:

**Theorem 2.3** *Given any $\delta > 0$, assume that, in each iteration,*
*(A1) the set $\{\mathbf{x}^1, \mathbf{x}^2, \cdots\}$ is bounded,*
*(A2') there exists $\bar{\delta} > 0$ such that $\mu_k(t_i^k) \geq \bar{\delta}$, $\forall i = 1, ..., m_k$, and $t_{m_k+1}^k \in E_{k+1}$,*
*(A3) $\phi_j^k < -\bar{\delta}$, $\forall j \notin B_k$,*
*(A4) $M_k^T$ has a square submatrix $D_k$ with rank $p_k$ $(= |B_k|)$ and $|det(D_k)| > \bar{\delta}$.*
*Then the proposed scheme terminates in a finite number of iterations, if the matrix $Q$ in $(QSIP)$ is positive semidefinite.*

Several comments on Theorems 2.2 and 2.3 can be made here. (1) Since the logic used for the proof of Theorem 2.3 follows the same logic as used in the proof of [25, Theorem 2.3], except for some technical details, we omit the proof here. (2) Note that in Theorem 2.2, the assumption (A2) is used for obtaining (22). If $\mathbf{x}^k$ and $\mathbf{y}^k$ are chosen to be optimal solutions of $(QP_{T_k})$ and $(DQP_{T_k})$, instead of being approximate solutions, when matrix $Q$ is positive definite, this assumption (A2) could be lifted. In this case, the duality gap (13) is zero which implies that $\phi_{N+1}(t_{m_N+1}^N) = 0$ and hence (22) holds. Otherwise, $\mu_{N+1}(t_{m_N+1}^N) = 0$ and hence $V(DQP_{T_{N+1}}(\mathbf{x}^{N+1}, \mathbf{y}^{N+1})) = V(DQP_{T_N}(\mathbf{x}^N, \mathbf{y}^N))$. This means that both $\mathbf{x}^{N+1}$ and $\mathbf{x}^N$ are optimal solutions to $(QP_{T_N})$. However, $\phi_{N+1}(t_{m_N+1}^N) \geq 0$ and $\phi_N(t_{m_N+1}^N) < 0$ imply that $\mathbf{x}^{N+1} \neq \mathbf{x}^N$, which contradicts the fact that $(QP_{T_N})$ has a unique solution given $Q$ is positive definite. This also explains that when $Q$ is only positive semidefinite, we have to specially include $t_{m_k+1}^k \in E_{k+1}$ in the assumption (A2') for Theorem 2.3. (3) In the case of linear semi-infinite programming, $t_{m_k+1}^k \in E_{k+1}$ is always true under the assumption that $(DLP_{T_k})$ is nondegenerate (see Theorem 2.3 of [25]).

The above theorems assure that the proposed relaxation scheme terminates in a finite number of iterations, say $k^*$ iterations, with an solution $\mathbf{x}^{k^*} = (x_1^{k^*}, \cdots, x_n^{k^*})^T$, such that $x_j^{k^*} > 0$, for $j \in B_{k^*}$. In this case, $\mathbf{x}^{k^*}$ is of course feasible for $(QP_{T_{k^*}})$. Actually the following theorem guarantees that $\mathbf{x}^{k^*}$ is a good approximate solution of $(QSIP)$, if $\delta$ is chosen to be small enough.

**Theorem 2.4** *For any given $\delta > 0$, if there exists $\bar{\mathbf{x}} = (\bar{x}_1, \cdots, \bar{x}_n)^T$ with $\bar{x}_j \geq -x_j^{k^*}/\delta$, $\forall j \in B_{k^*}$, and $\bar{x}_j \geq 0$, $\forall j \notin B_{k^*}$, such that*

$$\sum_{j=1}^{k} \bar{x}_j f_j(t) \geq 1, \quad \forall t \in T, \tag{25}$$

*then*

$$|V(QP_{T_{k^*}}(\mathbf{x}^{k^*})) - V(QSIP)| \leq \delta|\bar{\mathbf{x}}^T Q \mathbf{x}^{k^*} + \frac{1}{2}\delta\bar{\mathbf{x}}^T Q \bar{\mathbf{x}} + \sum_{j=1}^{n} c_j \bar{x}_j| + 2\Delta_{k^*} \tag{26}$$

*where $\Delta_{k^*}$ is obtained in Step 3 of the relaxation scheme.*

**Proof**:

Since

$$0 \le V(QP_{T_{k^*}}(\mathbf{x}^{k^*})) - V(DQP_{T_{k^*}}(\mathbf{x}^{k^*}, \mathbf{y}^{k^*})) < \Delta_{k^*}$$

and

$$V(DQP_{T_{k^*}}(\mathbf{x}^{k^*}, \mathbf{y}^{k^*})) \le V(DQSIP) = V(QSIP),$$

we have

$$V(QP_{T_{k^*}}(\mathbf{x}^{k^*})) - \Delta_{k^*} < V(QSIP).$$

By (25) and our assumption, it is clear to see that $\mathbf{x}^{k^*} + \delta\bar{\mathbf{x}}$ is feasible to $(QSIP)$. Moreover,

$$\begin{aligned}
& |V(QP_{T_{k^*}}(\mathbf{x}^{k^*})) - \Delta_{k^*} - V(QSIP)| \\
\le\ & |\frac{1}{2}(\mathbf{x}^{k^*})^T Q \mathbf{x}^{k^*} + \mathbf{c}^T \mathbf{x}^{k^*} - \Delta_{k^*} - \frac{1}{2}(\mathbf{x}^{k^*} + \delta\bar{\mathbf{x}})^T Q (\mathbf{x}^{k^*} + \delta\bar{\mathbf{x}}) - \mathbf{c}^T(\mathbf{x}^{k^*} + \delta\bar{\mathbf{x}})| \\
\le\ & \delta|\frac{1}{2}\bar{\mathbf{x}}^T Q \mathbf{x}^{k^*} + \frac{1}{2}(\mathbf{x}^{k^*})^T Q \bar{\mathbf{x}} + \frac{1}{2}\delta\bar{\mathbf{x}}^T Q \bar{\mathbf{x}} + \mathbf{c}^T \bar{\mathbf{x}}| + \Delta_{k^*} \\
\le\ & \delta|\bar{\mathbf{x}}^T Q \mathbf{x}^{k^*} + \frac{1}{2}\delta\bar{\mathbf{x}}^T Q \bar{\mathbf{x}} + \mathbf{c}^T \bar{\mathbf{x}}| + \Delta_{k^*}.
\end{aligned}$$

Consequently, the theorem follows. $\Xi$

# 3 Flexibility of the Scheme

The proposed relaxation scheme is very general, in the sense that it allows the relaxation techniques (ii), (iii), and (iv) to be integrated in one algorithm with a convergence proof. In this section, we show that the convergence proof is flexible for the proposed scheme to employ different solution engines, in particular, the infeasible (start) interior-point method (e.g., see [20]), for its subproblems.

Note that no matter a simplex-based or interior-point based method is used for solving subproblems $(QP_{T_k})$ and $(DQP_{T_k})$ in Step 1 of the proposed scheme, the (approximate) solutions $\mathbf{x}^k$ and $(\mathbf{x}^k, \mathbf{y}^k)$ are required stay feasible. Therefore some initialization procedure such as the phase-1 method or big-M method is required. This certainly builds up the computational burden. To relieve this burden, one may consider using the infeasible primal-dual interior-point method for solving $(QP_{T_k})$ and $(DQP_{T_k})$. In this case, the Step 1 of the proposed scheme could be further relaxed as follows.

Step 1: Find an approximate (maybe infeasible) solution $\mathbf{x}^k$ of $(QP_{T_k})$ and an approximate (maybe infeasible) solution $(\mathbf{x}^k, \mathbf{y}^k)$ of $(DQP_{T_k})$ such that

$$\sum_{i=1}^{m_k} |\min(s_i^k, 0)\mu_k(t_i^k)| < \Delta_k, \tag{27}$$

$$\sum_{j=1}^{n} |\max(0, \phi_j^{k-1})x_j^k| < \Delta_{k-1} \ \ (\text{if } k > 2), \tag{28}$$

$$\sum_{i=1}^{m_{k-1}} |\min(s_i^{k-1}, 0)\mu_k(t_i^{k-1})| < \Delta_k \ \ (\text{if } k > 2), \tag{29}$$

$$\sum_{j=1}^{n} |\max(0, \phi_j^{k})x_j^k| < \Delta_k, \tag{30}$$

and

$$0 \leq V(QP_{T_k}(\mathbf{x}^k)) - V(DQP_{T_k}(\mathbf{x}^k, \mathbf{y}^k)) < \Delta_k.$$

Since the solutions could be infeasible, the inequality $V(DQP_{T_k}\ (\mathbf{x}^k, \mathbf{y}^k)) \geq V(DQP_{T_{k-1}}$ $(\mathbf{x}^{k-1}, \mathbf{y}^{k-1}))$ may fail in Step 3. Hence Step 3 is modified correspondingly as follows.

Step 3 : Find any $t_{m_k+1}^k \in T$ such that $\phi_k(t_{m_k+1}^k) < -\delta$.
If such $t_{m_k+1}^k$ does not exist and $\Delta_k < \delta$, stop and output $\mathbf{x}^k$ as a solution.
If $k = 1$ or $V(DQP_{T_k}(\mathbf{x}^k, \mathbf{y}^k)) \geq V(DQP_{T_{k-1}}(\mathbf{x}^{k-1}, \mathbf{y}^{k-1}))$, but $t_{m_k+1}^k$ exists, set $T_{k+1} = E_k \cup \{t_{m_k+1}^k\}$; otherwise, set $\Delta_k = (1-\epsilon)\Delta_k$ and go to Step 1.

Now we show that the convergence proof of this modified scheme can be obtained accordingly. First we can check that Lemma 2.1 is still valid. In order to prove Theorem 2.2, after Eq. (20), i.e.,

$$\begin{aligned} & |V(DQP_{T_{N+1}}(\mathbf{x}^{N+1}, \mathbf{y}^{N+1})) - V(DQP_{T_N}(\mathbf{x}^N, \mathbf{y}^N))| \\ = \ & |\sum_{j=1}^{m_N} s_j^N \mu_N(t_i^N) - \sum_{j=1}^{n} \phi_j^N x_j^{N+1} - \Delta'_{N+1} + \frac{1}{2}(\mathbf{d}^N)^T Q \mathbf{d}^N| \leq \varepsilon^2, \end{aligned}$$

we do not have $s_j^N \geq 0$ and $\phi_j^N \leq 0$ any more. However, from (27) and (28), we have

$$\Delta_k^{P'} \equiv \sum_{i=1}^{m_k} |\min(s_i^k, 0)\mu_k(t_i^k)| < \Delta_k, \text{ and } \Delta_k^{D'} \equiv \sum_{j=1}^{n} |\max(0, \phi_j^k)x_j^{k+1}| < \Delta_k, k = 1, 2, \cdots,$$

and

$$
\begin{aligned}
0 \leq \frac{1}{2}(\mathbf{d}^N)^T Q \mathbf{d}^N &\leq \varepsilon^2 + \Delta'_{N+1} - \sum_{j=1}^{m_N} s_j^N \mu_N(t_i^N) + \sum_{j=1}^{n} \phi_j^N x_j^{N+1} \\
&\leq \varepsilon^2 + \Delta'_{N+1} + \sum_{i=1}^{m_N} |\min(s_i^N, 0)\mu_N(t_i^N)| + \sum_{j=1}^{n} |\max(0, \phi_j^N) x_j^{N+1}| \\
&\leq \varepsilon^2 + \Delta'_{N+1} + \Delta_N^{P'} + \Delta_N^{D'}. \qquad (31)
\end{aligned}
$$

Similarly, from (29) and (30), we can have

$$
\bar{\Delta}_k^{P'} \equiv \sum_{i=1}^{m_{k-1}} |\min(s_i^{k-1}, 0)\mu_k(t_i^{k-1})| < \Delta_k \text{ and } \bar{\Delta}_k^{D'} \equiv \sum_{j=1}^{n} |\max(0, \phi_j^k) x_j^k| < \Delta_k, k = 2, 3, \cdots.
$$

Consequently, (22) becomes

$$
0 \leq \phi_{N+1}(t_{m_N+1}^N) < \frac{(\Delta'_{N+1} + \bar{\Delta}_{N+1}^{P'} + \bar{\Delta}_{N+1}^{D'})}{\bar{\delta}} \qquad (32)
$$

and (23) is obtained. The corresponding proof of Theorem 2.2 follows.

When the matrix $Q$ is positive semidefinite, as discussed in Section 2, parallel to the work of [25], a corresponding convergence proof of Theorem 2.3 can be derived. As to Theorem 2.4, since the relation between $V(QSIP)$, and $V(QP_{T_{k^*}}(\mathbf{x}^{k^*}))$ becomes unclear, we are not able to provide a proof. However, in general, $x^{k^*}$ may still be an approximate solution.

## 4 Numerical Experiments

In this section, we study the effects of combined relaxation techniques by using the following problem setting:

$$
\begin{aligned}
\min \quad & \sum_{j=1}^{n} \frac{x_j}{j} + \frac{1}{2}\mathbf{x}^T Q \mathbf{x} \\
\text{s.t.} \quad & \sum_{j=1}^{n} x_j t^{j-1} \geq g(t), \quad t \in [0,1] \\
& x_j \geq 0, \quad j = 1, 2, ..., n
\end{aligned}
$$

where $Q$ is a diagonal matrix with $Q_{jj} = \frac{1}{j}$, for $j = 1, ..., n$. Four types of $g(t)$, namely, $sin(t)$, $\frac{1}{2-t}$, $e^t$, and $t^2$, were used for testing purpose. We call them Problem 1, Problem 2, Problem 3, and Problem 4, respectively.

Note that a version of Problem 2 with $n = 8$ free variables was studied in [4], and Problems 1, 2, and 3 with different $n$ were tested in [2]. In our computational experimentation, we fixed $n = 8$ for all testing problems. Also notice that the effect of dropping unnecessary constraints to reduce the computational work in each iteration was studied before (for example, see [25]), therefore our experiments were designed to focus only on the effects of combining relaxation techniques (ii) and (iii). Consequently, in the experiments, in the $k$th iteration, $T_k$ can be simplified as $\{t_1, \cdots, t_k\}$.

The MATLAB (version 5.0) [23] was installed on a SUN UltraSPARC1-140 workstation for our study. For a baseline study, the discretization method [10, 18] was applied to solve Problems 1 – 4, with $n = 8$. To be more precise, the interval $T = [0, 1]$ was discretized into 101 evenly spaced points to create a quadratic program with 101 explicit constraints for each case. The optimal objective value of each problem was obtained by using the qp subroutine of MATLAB and listed in the top row of Table 1 for reference.

Four possible combinations of the relaxation techniques (ii) and (iii) are denoted by "method 0-0," "method 1-0," "method 0-1," and "method 1-1," respectively. Method 0-0 is the classical cutting plan algorithm without using any relaxation technique, i.e., exact optimal solutions are found in both Step 2 and Step 3. Method 1-0 uses the relaxation technique (ii) but not (iii), i.e., inexact solutions with $0 \leq V(QP_{T_k}(\mathbf{x}^k)) - V(DQP_{T_k}(\mathbf{x}^k, \mathbf{y}^k)) < \Delta_k$ are used in Step 2. Also, initially we set $\Delta_0 = 0.1$ and update $\Delta_k$ by $\min(0.1\Delta_k, 10^{-5})$ in Step 3. Method 0-1 uses the relaxation technique (iii) but not (ii), i.e., an inexact solution $t_{k+1}$ with $\phi_k(t_{k+1}) < -\delta$ is used in Step 3. Method 1-1 uses both relaxation techniques as stated in Method 1-0 and Method 0-1.

For Method 0-0 and Method 1-0, the fmin subroutine of MATLAB was used for finding the global minimizer $t_{k+1} = \arg\min_{t \in T} \phi_k(t)$. For Method 0-1 and 1-1, $\delta$ was set to be 0.0001 and the interval $T$ was recursively discretized to find $t_{k+1}$. In other words, in the beginning, the interval $[0, 1]$ was discretized by 11 evenly spaced points (in the interval) and we test each point to see if $\phi(t) < -\delta$. If all points failed, then we refine the discretization by 101 points. If this failed again, then refine by 1001 points, etc. (Only in the worst situation, the MATLAB subroutine fmin is activated for finding $t_{k+1}$.) All four methods stopped when $\phi(t_{k+1}) \geq -0.0001$.

In each iteration, the subproblem $(QP_{T_k})$ was transformed into the following standard form:

$$\begin{aligned} \min \quad & \frac{1}{2}\bar{\mathbf{x}}^T \bar{Q} \bar{\mathbf{x}} + \bar{\mathbf{c}}^T \bar{\mathbf{x}} \\ s.t. \quad & A\bar{\mathbf{x}} = \mathbf{b} \\ & \bar{\mathbf{x}} \geq 0 \end{aligned}$$

where

$$A = \begin{bmatrix} f_1(t_1) & \cdots & f_n(t_1) & | & -1 & & \\ & \vdots & & | & & \ddots & \\ f_1(t_k) & \cdots & f_n(t_k) & | & & & -1 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} g(t_1) \\ \vdots \\ g(t_k) \end{bmatrix},$$

$$\bar{Q} = \begin{bmatrix} Q & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \bar{\mathbf{c}} = \begin{bmatrix} \mathbf{c} \\ \mathbf{0} \end{bmatrix},$$

and $\bar{\mathbf{x}}$ is a vector which includes $\mathbf{x}$ as its first $n$ components. Then the corresponding KKT system becomes

$$\begin{aligned} A\bar{\mathbf{x}} &= \mathbf{b}, && (33) \\ A^T\mathbf{y} + \mathbf{s} &= \bar{Q}\bar{\mathbf{x}} + \bar{\mathbf{c}}, && (34) \\ \bar{x}_i s_i &= 0,\ i = 1, \cdots, n+k, && (35) \end{aligned}$$

where $\mathbf{y}$ and $\mathbf{s}$ are dual variables.

The predictor-corrector method [17] was implemented to solve the KKT system (33) - (35) with the right-hand of (35) being replaced by a small scalar $\mu > 0$. To test the ideas of Section 2, $\mathbf{x}^k$ and $\mathbf{y}^k$ were kept to be primal and dual feasible in each iteration. For our test problems, since $0 \leq g(t) < 3$ for each form of $g(t)$, an initial solution pair were set to be

$$\bar{\mathbf{x}} = [n, \cdots, n]^T \text{ and } \mathbf{y} = [\frac{1}{k}, \cdots, \frac{1}{k}]^T.$$

For $n = 8$, it can be easily checked that, for the primal,

$$\begin{bmatrix} 1 & \cdots & t_1^{n-1} \\ & \vdots & \\ 1 & \cdots & t_k^{n-1} \end{bmatrix} \begin{bmatrix} n \\ \vdots \\ n \end{bmatrix} \geq \begin{bmatrix} n \\ \vdots \\ n \end{bmatrix} > \begin{bmatrix} g(t_1) \\ \vdots \\ g_n(t_k) \end{bmatrix}$$

and for the dual,

$$\begin{bmatrix} 1 & & \\ & \ddots & \\ & & \frac{1}{n} \end{bmatrix} \begin{bmatrix} n \\ \vdots \\ n \end{bmatrix} + \begin{bmatrix} 1 \\ \vdots \\ \frac{1}{n} \end{bmatrix} - \begin{bmatrix} 1 & \cdots & 1 \\ & \vdots & \\ t_1^{n-1} & \cdots & t_k^{n-1} \end{bmatrix} \begin{bmatrix} \frac{1}{k} \\ \vdots \\ \frac{1}{k} \end{bmatrix} \geq \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}.$$

The numerical results obtained for the experiments are shown in Table 1. In the table, the "obj." column represents the final objective value, column "cp" indicates the number of cutting-plane iterations each method took, and column "in-pt" shows the total number of interior-point iterations required.

By comparing the results of Method 0-0 with Method 1-0, as well as Method 0-1 with Method 1-1, we see that the relaxation technique (ii) works well in every case. However,

Table 1: Comparison of different methods

| method | Problem 1 | | | Problem 2 | | | Problem 3 | | | Problem 4 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | cp | in-pt | obj. | cp | in-pt | obj. | cp | in-pt | obj. | cp | in-pt | obj. |
| discretization | | | 0.5944 | | | 0.8370 | | | 2.5126 | | | 0.4003 |
| method 0-0 | 6 | 123 | 0.5944 | 5 | 108 | 0.8370 | 6 | 169 | 2.5126 | 7 | 128 | 0.4003 |
| method 1-0 | 6 | 102 | 0.5944 | 5 | 86 | 0.8370 | 6 | 135 | 2.5126 | 7 | 117 | 0.4003 |
| method 0-1 | 2 | 44 | 0.5944 | 10 | 214 | 0.8370 | 11 | 298 | 2.5125 | 6 | 112 | 0.4003 |
| method 1-1 | 2 | 22 | 0.5945 | 10 | 192 | 0.8370 | 11 | 266 | 2.5125 | 6 | 99 | 0.4003 |

by comparing the results of Method 0-0 with Method 0-1, as well as Method 1-0 with Method 1-1, we see that the relaxation technique (iii) could be a good idea, but not always working (e.g., Problems 2 and 3). The major problem we identified is that "when the global minimizer was relaxed by solving $\phi_k(t) < -\delta$, sometimes only a less critical violation point $t_{k+1}$ was found and hence the cutting-plane obtained became less effective." Since our implementation always searches from those discretization points in an ascending order, those points near 0 are selected much too often and too close. Consequently, they are not supportive enough for generating significant cuts as the number of iterations increases. Therefore, the experiments were conducted again by testing those discretization points in a random order. Table 2 shows the results.

Table 2: New results of Method 0-1 and Method 1-1

| method | Problem 1 | | | Problem 2 | | | Problem 3 | | | Problem 4 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | cp | in-pt | obj. | cp | in-pt | obj. | cp | in-pt | obj. | cp | in-pt | obj. |
| discretization | | | 0.5944 | | | 0.8370 | | | 2.5126 | | | 0.4003 |
| method 0-0 | 6 | 123 | 0.5944 | 5 | 108 | 0.8370 | 6 | 169 | 2.5126 | 7 | 128 | 0.4003 |
| method 1-0 | 6 | 102 | 0.5944 | 5 | 86 | 0.8370 | 6 | 135 | 2.5126 | 7 | 117 | 0.4003 |
| method 0-1 | 3 | 57 | 0.5944 | 6 | 118 | 0.8370 | 6 | 143 | 2.5125 | 7 | 135 | 0.4003 |
| method 1-1 | 3 | 44 | 0.5944 | 7 | 121 | 0.8370 | 4 | 65 | 2.5125 | 7 | 117 | 0.4003 |

The results shown in Table 2 confirmed our reasoning and we really observed that $t_k$'s were well distributed between $[0, 1]$ to generate critical cuts. Moreover, the results of comparing Method 0-0 with Method 1-1 in Table 2 suggest that a combination of the relaxation techniques (ii) and (iii) in general works well.

Now we move to another experiment. Recall that, in Section 3, the "feasibility" requirement of $\mathbf{x}^k$ and $(\mathbf{x}^k, \mathbf{y}^k)$ in each iteration was further relaxed without jeopardizing the convergence proof. This relaxation technique is important for practical implementation because the interior-feasible solutions are not always readily available. To study its effect, in each iteration, for $(QP_{T_k})$ and $(DQP_{T_k})$, we took $\bar{\mathbf{x}} = [1, \cdots, 1]^T$ as a starting point of the primal system (33), $\mathbf{y} = [0, \cdots, 0]^T$ and $\mathbf{s} = [1, \cdots, 1]^T$ for the dual system (34). For practicality reason,

$$\frac{\|A\bar{\mathbf{x}} - \mathbf{b}\|}{1 + \|\mathbf{b}\|} < \Delta_k \quad \text{and} \quad \frac{\|A^T\mathbf{y} + \mathbf{s} - Q\bar{\mathbf{x}} - \bar{\mathbf{c}}\|}{1 + \|\bar{\mathbf{c}}\|} < \Delta_k,$$

were used for the feasibility check in our experiments. The results are shown by Table 3.

Table 3: Relaxation of the feasibility requirement

| method | Problem 1 | | | Problem 2 | | | Problem 3 | | | Problem 4 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | cp | in-pt | obj. | cp | in-pt | obj. | cp | in-pt | obj. | cp | in-pt | obj. |
| method 1-1 | 6 | 116 | 0.5944 | 5 | 97 | 0.8370 | 6 | 160 | 2.5126 | 7 | 115 | 0.4003 |
| method 1-0 | 6 | 96 | 0.5944 | 5 | 75 | 0.8370 | 6 | 125 | 2.5126 | 7 | 108 | 0.4003 |
| method 0-1 | 2 | 34 | 0.5944 | 10 | 195 | 0.8370 | 11 | 277 | 2.5125 | 6 | 99 | 0.4003 |
| method 1-1 | 2 | 17 | 0.5944 | 10 | 173 | 0.8370 | 11 | 246 | 2.5125 | 6 | 85 | 0.4003 |

Table 3 supports similar conclusions drawn from Table 1. Comparing with Table 1, Table 3 also indicates that the relaxation of "feasibility" requirement indeed reduces the computational work. Therefore using an infeasible-start method as the solution engine for solving subproblems becomes a valid argument. We did not generate a corresponding table to compare with Table 2 because of the randomness involved in generating discretization points for selecting $t_k$. But similar results can be expected.

# 5 Concluding Remarks

1. In this paper, we have proposed a new relaxation scheme of the cutting plane method for solving convex quadratic semi-infinite programming problems. The proposed scheme is very general and flexible. It allows us to explore the possibility of combining the use of "approximate solutions" for each subproblem, the use of "inexact" minimizer for generating new cuts, the idea of "dropping unnecessary constraints" in each iteration, and the use of "infeasible-start" interior-point method as the solution engine.

2. The results obtained in this paper have generalized most, if not all, known relaxation results for both linear and convex quadratic semi-infinite programming problems.

3. Although we have only addressed the convex quadratic semi-infinite programming problems in this paper, all convergence proofs derived here can be used as the basis for designing relaxed cutting plane methods for solving convex semi-infinite programming problems.

## References


[1] E. J. Anderson and P. Nash. *Linear Programming in Infinite–dimensional Spaces.* Wiley, Chichester, 1987.

[2] S.-C. Fang, C.-J. Lin, and S.-Y. Wu. On solving convex quadratic semi-infinite programming problems. *Optimization*, 31:107–125, 1994.

[3] S.-C. Fang and S. Puthenpura. *Linear Optimization and Extensions: Theory and Algorithms.* Kluwer Academic Publishers, Norwell, MA, 1993.

[4] M. C. Ferris and A. B. Philpott. An interior point algorithm for semi–infinite linear programming. *Mathematical Programming*, 43:257–276, 1989.

[5] K. Glashoff and S. Å. Gustafson. *Linear Optimization and Approximation.* Springer Verlag, New York, New York, 1982.

[6] P. R. Gribik. A central cutting plane algorithm for SIP. In R. Hettich, editor, *Semi-infinite Programming*, *Lecture Notes in Control and Information Sciences*, vol. 15, pages 66–82, Springer-Verlag, Berlin, 1979.

[7] S. Å. Gustafson and K. O. Kortanek. Numerical treatment of a class of semi-infinite programming problems. *Naval Research Logistics Quarterly*, 20:477–504, 1973.

[8] S. Å. Gustafson and K. O. Kortanek. Computation of optimal experimental designs for air quality surveillance. In P. J. Jansen, O. Moeschlin, and O. Rentz, editors, *Quantitative Modelle für Ökonomisch-Ökologische Analysen*, pages 43–60. Verlag Anton Hain, Meisenheim, Germany, 1976.

[9] S. Å. Gustafson and K. O. Kortanek. Computational schemes for semi-infinite programs. Technical report, Department of Mathematics, Carnegie-Mellon University, Pittsburgh, Pennsylvania, 1977.

[10] R. Hettich and K. Kortanek. Semi-infinite programming: theory, method and application. *SIAM Review*, 35:380–429, 1993.

[11] K.-H. Hoffmann and A. Klostermair. A semi-infinite linear programming procedure and applications to approximation problems in optimal control. In G. G. Lorentz, C. K. Chui, and L. L. Schumaker, editors, *Approximation Theory II*, pages 379–389, Academic Press, 1976.

[12] H. Hu. A one-phase algorithm for semi-infinite linear programming. *Mathematical Programming*, 46:85–103, 1990.

[13] H.-C. Lai and S.-Y. Wu. Extremal points and optimal solutions for general capacity problems. *Mathematical Programming*, 54:87–113, 1992.

[14] H.-C. Lai and S.-Y. Wu. On linear semi-infinite programming problems: an algorithm. *Numerical Functional Analysis and Optimization*, 13:287–304, 1992.

[15] C.-J. Lin, S.-C. Fang, and S.-Y. Wu. A dual affine scaling based algorithm for solving linear semi-infinite programming problems. In D. Z. Du and J. Sun, editors, *Advances in Optimization and Application*, pages 217–234. Kluwer Academic Publishers, Boston, 1994.

[16] C.-J. Lin, S.-C. Fang, and S.-Y. Wu. An unconstrained convex programming approach to linear semi-infinite programming. *SIAM Journal on Optimization*, 8(2):443–456, 1998.

[17] S. Mehrotra. On the implementation of a (primal dual) interior point methods. *SIAM Journal on Optimization*, 2:575–601, 1992.

[18] R. Reemtsen and S. Görner. Numerical methods for semi-infinite programming: a survey. In R. Reemtsen and J-J. Rueckmann, editors, *Semi-Infinite Programming*, pages 195–275, Kluwer Academic Publishers, Boston, 1998.

[19] K. Roleff. A stable multiple exchange algorithm for linear SIP. In R. Hettich, editor, *Semi-infinite Programming, Lecture Notes in Control and Information Sciences*, vol. 15, pages 83–96, Springer-Verlag, Berlin, 1979.

[20] R. Saigal. *Linear Programming: a Modern Integrated Analysis.* Kluwer Academic Publishers, Norwell, MA, 1995.

[21] R.-L. Sheu, S.-Y. Wu, and S.-C. Fang. A primal-dual infeasible interior-point algorithm for linear semi-infinite programming. *Computers and Mathematics with Applications*, 29(8):7–18, 1995.

[22] P. T. P. Tang. A fast algorithm for linear complex Chebyshev approximations. *Mathematics of Computation*, 51(184):721–739, 1988.

[23] The MathWorks Inc. *MATLAB User Guide.* Natick, MA 01760-1500, 1996.

[24] R. Tichatschke and V. Nebeling. A cutting plane algorithm for solving quadratic semi-infinite programs. *Optimization*, 19:803–817, 1988.

[25] S.-Y. Wu, S.-C. Fang, and C.-J. Lin. Relaxed cutting plane method for solving linear semi-infinite programming problems. To appear in *Journal of Optimization Theory and Applications*, 99(3), December, 1998.