ANZIAM J. 45(2004), 477–494

A SOLUTION METHOD FOR COMBINED SEMI-INFINITE AND SEMI-DEFINITE PROGRAMMING

S. J. LI[1], X. Q. YANG[2], K. L. TEO[2] and S. Y. WU[3]

(Received 10 June, 2003; revised 24 July, 2003)

Abstract

In this paper, we develop a discretisation algorithm with an adaptive scheme for solving a class of combined semi-infinite and semi-definite programming problems. We show that any sequence of points generated by the algorithm contains a convergent subsequence; and furthermore, each accumulation point is a local optimal solution of the combined semi-infinite and semi-definite programming problem. To illustrate the effectiveness of the algorithm, two specific classes of problems are solved. They are relaxations of quadratically constrained semi-infinite quadratic programming problems and semi-infinite eigenvalue problems.

1. Introduction

Let S^n denote the set of real symmetric $n \times n$ matrices. The standard inner product on S^n is $A \bullet B = \mathrm{tr}\{AB\} = \sum_{i,j} a_{ij} b_{ij}$. By $X \succeq 0$, where $X \in S^n$, we mean that the matrix X is positive semi-definite and $\|X\|_F$, or simply $\|X\|$, is the Frobenius norm of matrix X : $\|X\|_F = (X \bullet X)^{1/2}$. Here S^n_+ denotes the set of all positive semi-definite matrices in S^n.

We consider the following combined semi-infinite and semi-definite programming problem (SISDP):

$$\sup C \bullet X \quad \text{s.t.} \begin{cases} A_i \bullet X = a_i, & i = 1, 2, \ldots, l, \\ B(t) \bullet X \leq b(t), & t \in T, \\ X \succeq 0. \end{cases}$$

[1]Department of Information and Computer Sciences, College of Sciences, Chongqing University, Chongqing, 400044, China; e-mail: lisj@cqu.edu.cn.
[2]Department of Applied Mathematics, The Hong Kong Polytechnic University, Kowloon, Hong Kong; e-mail: mayangxq@polyu.edu.hk and mateokl@polyu.edu.hk.
[3]Institute of Applied Mathematics, National Cheng-Kung University, Tainan 700, Taiwan; e-mail: soonyi@mail.ncku.edu.tw.

Here, T is a nonempty compact set in $\mathscr{R}$ and $C, A_i, i = 1, 2, \ldots, l$, and $B(t), t \in T$, are all fixed matrices in S^n. Let $a_i, i = 1, 2, \ldots, l$, $b(t) \in \mathscr{R}, t \in T$, be fixed real numbers, and let $X \in S^n$ be the decision matrix to be optimised upon.

Three important special cases of problem (SISDP) are given as follows.

EXAMPLE 1.1. Suppose that T is a finite set. Then problem (SISDP) becomes a general semi-definite programming problem (SDP):

$$\sup C \bullet X \quad \text{s.t.} \begin{cases} A_i \bullet X = a_i, & i = 1, 2, \ldots, l, \\ B(t_j) \bullet X \leq b(t_j), & t_j \in T = \{t_1, \ldots, t_m\}, \\ X \succeq 0. \end{cases}$$

Some relevant references for (SDP) are [1, 27, 15] and [6].

EXAMPLE 1.2. Suppose that

$$C = \begin{pmatrix} c_1 & & 0 \\ & \ddots & \\ 0 & & c_n \end{pmatrix}, \quad A_i = \begin{pmatrix} g_1^i & & 0 \\ & \ddots & \\ 0 & & g_n^i \end{pmatrix}, \quad i = 1, 2, \ldots, l,$$

$$B(t) = \begin{pmatrix} h_1(t) & & 0 \\ & \ddots & \\ 0 & & h_n(t) \end{pmatrix}, \quad \forall\, t \in T, \quad \text{and} \quad X = \begin{pmatrix} x_{11} & \cdots & x_{1n} \\ & \cdots & \\ x_{n1} & \cdots & x_{nn} \end{pmatrix}.$$

Then problem (SISDP) is reduced to a semi-infinite linear programming problem (SILP):

$$\sup c^T x \quad \text{s.t.} \begin{cases} (g^i)^T x = a_i, & i = 1, 2, \ldots, l, \\ (h(t))^T x \leq b(t), & t \in T, \\ x \geq 0, \end{cases}$$

where $x = (x_{11}, \ldots, x_{nn})^T$, $c = (c_1, c_2, \ldots, c_n)^T$, $g^i = (g_1^i, \ldots, g_n^i)^T$, $i = 1, \ldots, l$, and $h(t) = (h_1(t), h_2(t), \ldots, h_n(t))^T$.

Obviously, if $x^* = (x_{11}^*, \ldots, x_{nn}^*)^T \in \mathscr{R}^n$ is a solution of (SILP), then the matrix

$$X^* = \begin{pmatrix} x_{11}^* & & 0 \\ & \ddots & \\ 0 & & x_{nn}^* \end{pmatrix} \in S^n$$

is a solution of (SISDP). Conversely, if $X^* = (x_{ij}^*)_{n \times n} \in S^n$ is a solution of (SISDP), then the point $x^* = (x_{11}^*, x_{22}^*, \ldots, x_{nn}^*)^T \in \mathscr{R}^n$ is a solution of (SILP).

https://doi.org/10.1017/S1446181100013511 Published online by Cambridge University Press

EXAMPLE 1.3. We consider a quadratically constrained semi-infinite quadratic programming problem (Q^2P):

$$\begin{aligned} &\max x^T Q_0 x + 2g_0^T x + \alpha_0 \\ &\text{s.t.}\, x^T Q(t) x + 2g(t)^T x + \alpha(t) \le b(t), \quad t \in T, \end{aligned}$$

where T is a nonempty compact set in $\mathscr{R}$, $\alpha(t)$ and $b(t)$, $t \in T$, are real numbers in $\mathscr{R}$ and Q_0 and $Q(t)$, $t \in T$, are all symmetric matrices in S^n. Here the objective function and the constraints are not necessarily convex. Clearly, the feasible set of (Q^2P) can be a very "nasty" set, and hence this problem is, in general, very hard to solve. For more details, see [17, 28].

Let $P_0 = \begin{bmatrix} \alpha_0 & g_0^T \\ g_0 & Q_0 \end{bmatrix}$ and $P(t) = \begin{bmatrix} \alpha(t) & g(t)^T \\ g(t) & Q(t) \end{bmatrix}$. Then an equivalent formulation of (Q^2P) is $(Q^2P)_y$:

$$\max y^T P_0 y \quad \text{s.t.} \begin{cases} y^T P(t) y \le b(t), & t \in T, \\ y_0^2 = 1, \\ y = (y_0, x^T)^T \in \mathscr{R}^{n+1}. \end{cases}$$

It is clear that if $(1, (x^*)^T)^T$ is optimal to $(Q^2P)_y$, then x^* is optimal to (Q^2P) and the values of (Q^2P) at x^* and $(Q^2P)_y$ at $(1, (x^*)^T)^T$ are equal. Since the values of $(Q^2P)_y$ at $(1, (x^*)^T)^T$ and $(-1, -(x^*)^T)^T$ are equal, we may consider that the optimal values of (Q^2P) and $(Q^2P)_y$ are equal. The equivalent formulation simplifies the notion and opens the way to the semi-definite relaxation since we can rewrite $(Q^2P)_y$ in the form of (SISDP) as

$$\max P_0 \bullet Y \quad \text{s.t.} \begin{cases} E_{00} \bullet Y = 1, \\ P(t) \bullet Y \le b(t), & t \in T, \\ \operatorname{rank}(Y) = 1, \quad Y \succeq 0, \end{cases}$$

where E_{00} is the zero matrix with 1 in the top left corner. Dropping the constraint of $\operatorname{rank}(Y) = 1$ leads to a relaxation of (Q^2P):

$$\max P_0 \bullet Y \quad \text{s.t.} \begin{cases} E_{00} \bullet Y = 1, \\ P(t) \bullet Y \le b(t), & t \in T, \\ Y \succeq 0. \end{cases} \qquad (\bar{P})$$

Thus we see that problem (SISDP) includes the semi-definite programming problems and the semi-infinite linear programming problems as special cases. Moreover, the problem (SISDP) can also be considered as a relaxation of a class of quadratically constrained semi-infinite quadratic programming problems.

https://doi.org/10.1017/S1446181100013511 Published online by Cambridge University Press

Conversely, let V_j, $j = 1, \ldots, \bar{n}$ be an orthonormal basis for the linear space of $n \times n$ symmetric matrices, where $\bar{n} = (n/2) \times n$. Let (see [13])

$$C = \sum_{j=1}^{\bar{n}} c_j V_j \equiv c^T V, \qquad A_i = \sum_{j=1}^{\bar{n}} h_j^i V_j \equiv (h^i)^T V, \qquad i = 1, \ldots, l,$$

$$X = \sum_{j=1}^{\bar{n}} x_j V_j \equiv x^T V, \qquad B(t) = \sum_{j=1}^{\bar{n}} g_j(t) V_j \equiv g(t)^T V, \quad t \in T,$$

and $v(s) = (v_1(s), \ldots, v_{\bar{n}}(s))^T$, with $v_j(s) = -s^T V_j s$, where $s \in \mathscr{R}^n$. Because of the fact that X is positive semi-definite if and only if $s^T X s \geq 0$ for any $s \in \mathscr{R}^n$, the problem (SISDP) is the same as the following semi-infinite programming problem:

$$\sup c^T x \quad \text{s.t.} \begin{cases} (h^i)^T x = a_i, & i = 1, 2, \ldots, l, \\ g(t)^T x \leq b(t), & t \in T, \\ v(s)^T x \leq 0, & s \in T_1 = \{s \in \mathscr{R}^n \mid \|s\| = 1\}, \end{cases}$$

that is, the problem (SISDP) can be transformed into a semi-infinite linear programming problem. However, the dimension of the index set of the semi-infinite linear programming problem is $(n + 1)$, where n is the dimension of the decision variable of the original problem (SISDP) with the index set of dimension 1. In practice, n can be large, say $n = 10$. For this situation, it does not appear possible to develop a practically feasible method, based on currently available results and approaches, for solving this semi-infinite programming problem with an index set of such a large dimension. On the other hand, the original problem (SISDP) with $n = 10$ is only a small scale problem, which can easily be solved by the methods developed in this paper. Thus it is much more natural to consider the problem (SISDP) as formulated in this paper.

There are many good reasons to study (SDP). First, (SDP) problems directly arise in a number of important applications, for example, structural optimisation, discrete (combinatorial) optimisation, robust filter design in signal processing, systems and control problems. Second, many convex optimisation problems, such as linear programming problems and convex quadratic programming problems, can be cast as semi-definite programming problems. We know that (SDP) is a generalisation of linear programming problems. In [2], simplex-like methods for (SDP) were discussed. In [2] and [5], simplex-like methods were obtained to solve the problem (SIP), which includes (SDP) as a special case. It is known that the primal (SDP) and its dual (SDP) may have a nonzero duality gap and that the interior point method is an efficient method to solve semi-definite programming problems up to any tolerance ϵ, in a polynomial number of arithmetic operations. See [1, 6, 8, 16, 21, 24] and [29].

Semi-infinite programming (SIP) arises in various fields of engineering such as control system design, general approximation, resource allocation in decentralised systems, decision making under competition, multi-objective optimisation, and optimum filter design in signal processing. See [7] and [23]. There are many solution methods (see [30, 31] and [7]) for solving (SIP) problems. Some outer approximation methods were obtained in [22] for nonlinear (SIP) problems with an adaptive scheme. Relevant convergence results were also established in [22]. In [25], an inner approximation method is developed for solving nonlinear (SIP) problems.

Motivated by the work reported in [22] and [25], we develop a discretisation algorithm with an adaptive scheme for solving the problem (SISDP), where each (SDP) subproblem is solved by an interior point method. We establish the convergence of the algorithm. Finally, we apply this algorithm to solve two important specific classes of problems: relaxation problems of quadratically constrained semi-infinite quadratic programming problems and semi-infinite eigenvalue problems.

The rest of the paper is organised as follows. In Section 2, the Lagrangian dual problem is introduced for the problem (SISDP). An algorithm for solving the problem (SISDP) is given. In Section 3, the convergence property of the algorithm is established. In Section 4, numerical results are presented.

2. Discretisation algorithms

Let us first introduce some notation. For a nonempty compact interval T in $\mathscr{R}$, let $\mathscr{R}^T = \prod_T \mathscr{R}$ denote the product space equipped with the product topology, which is a locally convex Hausdorff topological vector space; see [12]. Then the topological dual space of $\mathscr{R}^T$ is the generalised finite sequence space consisting of all functions $g : T \to \mathscr{R}$ with a finite support. The set $\mathscr{R}^T_+ = \prod_T \mathscr{R}_+$ denotes the convex cone of all nonnegative functions on T. Then the dual cone of $\mathscr{R}^T_+$ is defined by

$$\Lambda_T = \left\{ y = \{y(t)\}_{t\in T} \;\middle|\; \begin{array}{l} (\exists \text{ a finite set } F \subseteq T)(\forall t \in T\backslash F)\ y(t) = 0 \\ \text{and } (\forall t \in F)\ y(t) > 0 \end{array} \right\}.$$

For this result, see [9].

For the programming problem (SISDP), we introduce the Lagrangian dual problem (DSISDP) as follows:

$$\inf a^T z + \sum_{t\in T} y(t)b(t) \quad \text{s.t.} \begin{cases} \sum_{i=1}^{l} z_i A_i + \sum_{t\in T} y(t)B(t) - Z = C, & y \in \Lambda_T, \\ z \in \mathscr{R}^l, \quad Z \succeq 0, \end{cases}$$

where $a = (a_1, \ldots, a_l)^T$ and $z = (z_1, \ldots, z_l)^T$.

https://doi.org/10.1017/S1446181100013511 Published online by Cambridge University Press

When the parameter set T is finite, (SISDP) and (DSISDP) become a pair of primal and dual semi-definite programming problems. See [29] for reference. In this paper, we assume throughout that the problem (SISDP) and its dual problem (DSISDP) have optimal solutions and that their optimal values are equal. This assumption is satisfied under the situations considered in the following remark.

REMARK 2.1. Let $\tilde{B}(t) = \begin{bmatrix} B(t) & 0 \\ 0^T & -b(t) \end{bmatrix}$. Suppose that $\{\tilde{B}(t) \mid t \in T\}$ is a compact set, the Slater condition holds (that is, there exists an $X_0 \succeq 0$ such that $B(t) \bullet X_0 < b(t)$, $\forall t \in T$ and $A_i \bullet X_0 = a_i$, $i = 1, \ldots, l$), (SISDP) has an optimal solution and the optimal objective function value is finite. Then by [4, Theorem 2.2] and relevant parts in the proof of [4, Theorem 2.3], it follows that (SISDP) and (DSISDP) have optimal solutions and their optimal values are equal.

In view of Remark 2.1, we see that under appropriate constraint qualifications, we can guarantee that the optimal values of (SISDP) and (DSISDP) are equal. For more detailed discussion, see [3, 21] and [26].

Assume that $T = [T_1, T_2]$. Then the problem (SISDP) becomes the following combined semi-infinite and semi-definite programming problems:

$$\sup C \bullet X \quad \text{s.t.} \begin{cases} A_i \bullet X = a_i, & i = 1, 2, \ldots, l, \\ B(t) \bullet X \leq b(t), & t \in [T_1, T_2], \\ X \succeq 0. \end{cases} \tag{P_0}$$

In this section, we develop a discretisation method with an adaptive scheme for solving problem (P_0). A sequence of discretised subproblems is obtained, and each semi-definite programming subproblem is solved by an interior point method [6].

The feasible set of (P_0) is denoted by

$$\mathscr{F} = \{X \in S_+^n : A_i \bullet X = a_i,\ i = 1, 2, \ldots, l,\ B(t) \bullet X \leq b(t),\ t \in [T_1, T_2]\}.$$

We consider the following discretisation scheme: Given an integer $N > 0$, let

$$\Omega_N = \left\{ t_i = T_1 + \frac{i(T_2 - T_1)}{2^N} : i = 0, 1, \ldots, 2^N \right\}.$$

We introduce the following discretised problem:

$$\sup C \bullet X \quad \text{s.t.} \begin{cases} A_i \bullet X = a_i, & i = 1, 2, \ldots, l, \\ B(t_j) \bullet X \leq b(t_j), & t_j \in \Omega_N, \\ X \succeq 0. \end{cases} \tag{$\bar{P}_N$}$$

https://doi.org/10.1017/S1446181100013511 Published online by Cambridge University Press

The feasible set of $(\bar{P}_N)$ is denoted by

$$\mathscr{F}_N = \{X \in S_+^n : A_i \bullet X = a_i,\ i = 1, 2, \ldots, l,\ B(t_j) \bullet X \leq b(t_j),\ t_j \in \Omega_N\}.$$

We have the following lemma.

LEMMA 2.1. *Consider the problems* (P_0) *and* $(\bar{P}_N)$. *Then* $\mathscr{F} \subset \mathscr{F}_N$.

A direct method for solving the problem (P_0) is to solve a sequence of discretised problems $(\bar{P}_N)$. The solution X_N of $(\bar{P}_N)$ is used as an approximate solution of the problem (P_0). However, the above discretised problem $(\bar{P}_N)$ is, in general, a good approximation of the original (P_0) only if the integer N is large enough. Obviously, such a simple approximation of $[T_1, T_2]$ by the discretised subset Ω_N with a large number N leads to the problem $(\bar{P}_N)$ with a large number of inequality constraints. In order to overcome the difficulty of solving the discretised problem $(\bar{P}_N)$ with a large number of inequality constraints, we introduce an adaptive scheme strategy. More specifically, in each iteration, we add only one additional constraint. Moreover, in solving each subproblem, we need only to obtain an inexact optimal solution instead of an exact solution of the subproblem. The implemented algorithm for solving the problem (SISDP) is stated as follows.

DISCRETISATION ALGORITHM. Let $\{\epsilon_k\}$ be a strictly monotone decreasing sequence with $\epsilon_k \to 0$ (as $k \to \infty$) and let $\{N_m\}$ be a strictly monotone increasing integer sequence with $N_m \to \infty$ (as $m \to \infty$).

Step 1. $E_1 = \Omega_1$, $M_1 = \mathscr{F}_1$, $k = \bar{k} = 1$, $m = 1$.

Step 2. Find a feasible solution $X_k \in M_k$ of the following problem (P_k) (that is, X_k satisfies the constraints of problem (P_k)) and a feasible solution (z^k, y_k, Z_k) of the dual problem (D_k) (that is, (z^k, y_k, Z_k) satisfies the constraints of the dual problem (D_k)) such that

$$\left(a^T z^k + \sum_{t \in E_k} y_k(t) b(t)\right) - C \bullet X_k \leq \epsilon_k, \tag{2.1}$$

where

$$\sup C \bullet X \quad \text{s.t. } X \in M_k, \tag{P_k}$$

$$\inf a^T z + \sum_{t \in E_k} y(t) b(t) \quad \text{s.t.} \begin{cases} \sum_{i=1}^{l} z_i A_i + \sum_{t \in E_k} y(t) B(t) - Z = C, \\ z \in \mathscr{R}^l, \quad y(t) \geq 0, \quad t \in E_k, \quad Z \succeq 0. \end{cases} \tag{D_k}$$

Increase $\bar{k}$ to $\bar{k} + 1$ and construct $\Omega_{\bar{k}+1}$. Go to Step 3.

https://doi.org/10.1017/S1446181100013511 Published online by Cambridge University Press

Note that the constraint (2.1) is to ensure that the difference between the two objective function values is less than or equal to the tolerance ϵ_k. The solutions X_k and (z^k, y_k, Z_k) of the respective problems (P_k) and (D_k) with the condition (2.1) being satisfied up to the tolerance ϵ_k are called inexact optimal solutions up to the tolerance ϵ_k.

Step 3. Find a t_k such that $B(t_k) \bullet X_k - b(t_k) = \max_{t \in \Omega_{\bar{k}+1}} (B(t) \bullet X_k - b(t))$.

Step 4 If $B(t_k) \bullet X_k - b(t_k) > 0$, go to Step 5.

If $B(t_k) \bullet X_k - b(t_k) \leq 0$ and $\bar{k} < N_m$, set $\bar{k} = \bar{k} + 1$. Increase $\bar{k}$ to $\bar{k} + 1$ and construct $\Omega_{\bar{k}+1}$. Go to Step 3.

If $B(t_k) \bullet X_k - b(t_k) \leq 0$ and $\bar{k} \geq N_m$, increase m to $m + 1$ and k to $k + 1$ and give N_{m+1} and ϵ_{k+1}, respectively. Set $M_{k+1} = M_k$. Go to Step 6.

Step 5 Set

$$M_{k+1} = \left\{ X \in S_+^n \mid A_i \bullet X = a_i,\ i = 1, \ldots, l,\ B(t) \bullet X \leq b(t),\ t \in E_{k+1} \right\},$$

where $E_{k+1} = E_k \bigcup \{t_k\}$.

Increase m to $m + 1$ and k to $k + 1$ and give N_{m+1} and ϵ_{k+1}, respectively. Go to Step 6.

Step 6. Set $k = k + 1$, $\bar{k} = \bar{k} + 1$, $m = m + 1$ and go to Step 2.

For practical implementation, we will include a stopping criterion: We choose an integer N and a real number $\epsilon > 0$, and we will terminate the algorithm when $N_m \geq N$ and $\epsilon_n \leq \epsilon$. For example, we can take $N = 11$ and $\epsilon = 0.0001$.

Now we discuss the idea of the discretisation algorithm in detail as follows. Suppose that $k = \bar{k} = 1$ and $m = 1$. For the kth iteration, we assume that the optimal value of (P_k) is equal to that of (D_k). By a primal and dual interior point method for semi-definite programming, we find feasible solutions X_k and (z^k, y_k, Z_k) of the respective problems (P_k) and (D_k) with the condition (2.1) being satisfied up to the tolerance ϵ_k, where we take $\epsilon_k = 10^{-k}$. They are referred to as inexact optimal solutions of (P_k) and (D_k) up to the tolerance ϵ_k. Next, we find a $t_k \in \Omega_{\bar{k}+1}$ such that

$$B(t_k) \bullet X_k - b(t_k) = \max_{t \in \Omega_{\bar{k}+1}} (B(t) \bullet X_k - b(t)).$$

There are two cases to consider:

Case 1. $B(t_k) \bullet X_k - b(t_k) > 0$, that is, $X_k \notin \mathscr{F}_{\bar{k}+1}$. Take $E_{k+1} = E_k \bigcup \{t_k\}$ and construct M_{k+1}, which is the feasible set of (P_{k+1}). Set $k = k + 1$, $\bar{k} = \bar{k} + 1$ and $m = m + 1$. Then we solve the problems (P_k) and (D_k) until the condition (2.1) is satisfied up to the tolerance ϵ_k by a primal and dual interior point method for semi-definite programming.

Case 2. $B(t_k) \bullet X_k - b(t_k) \leq 0$, that is, $X_k \in \mathscr{F}_{\bar{k}+1}$. Suppose that we also construct M_{k+1} by taking $E_{k+1} = E_k \bigcup \{t_k\}$ and then solve the problems (P_{k+1}) and (D_{k+1}). Then X_k and (z^k, y_k, Z_k) are also inexact optimal solutions of the respective problems (P_{k+1}) and (D_{k+1}) with the condition (2.1) being satisfied up to the same tolerance ϵ_k.

https://doi.org/10.1017/S1446181100013511 Published online by Cambridge University Press

Clearly, it does not give us any further information. Therefore, for this situation, we need to increase the partition number $\bar{k}$ of T, that is, $\bar{k} = \bar{k} + 1$. Then we find t_k from a finer discretisation set $\Omega_{\bar{k}+1}$ such that

$$B(t_k) \bullet X_k - b(t_k) = \max_{t \in \Omega_{\bar{k}+1}} (B(t) \bullet X_k - b(t)).$$

If $B(t_k) \bullet X_k - b(t_k) > 0$, repeat Case 1.

If $B(t_k) \bullet X_k - b(t_k) \leq 0$, repeat Case 2. If the algorithm repeats Case 2 continuously, it means that the partition number $\bar{k}$ of T increases continuously. When the partition number is suitably large (for example, $\bar{k} \geq N_m$, where we may take $N_m = 4m$), we shall decrease the tolerance and solve the problems (P_{k+1}) and (D_{k+1}) up to the new tolerance. Namely, when $\bar{k} \geq N_m$, take $k = k + 1$, $\bar{k} = \bar{k} + 1$ and $m = m + 1$. Then we solve the problems (P_k) and (D_k) until the condition (2.1) is satisfied up to the new tolerance ϵ_k. By repeating this process, a sequence $\{X_k\}$ is obtained.

REMARK 2.2. Many computational methods are now available for solving general semi-infinite programming problems. These methods are based on exchange methods, cutting plane methods or discretisation methods. For details, see [5, 7] and [11]. Specific features of our algorithm, which is a discretisation method, are as follows.

(1) In view of the Introduction, we can transform the problem (SISDP) into a semi-infinite programming problem. However, if the problem (SISDP) contains decision variables of n dimensions and the index set is of 1 dimension, the index set of the semi-infinite programming problem is of $(n + 1)$ dimensions. For large n, the discretisation of the new index set is computationally a formidable task. Our algorithm solves the problem (SISDP) directly. Thus, we only need to partition the one-dimensional index set T.

(2) When the set $\Omega_{\bar{k}}$ of the discretisation points changes to $\Omega_{\bar{k}+1}$, the number of points in E_{k+1} does not necessarily increase. Only when $B(t_k) \bullet X_k - b(t_k) > 0$ do we increase a point in E_{k+1}, that is, $E_{k+1} = E_k \bigcup \{t_k\}$. In Step 3, the optimisation problem is solved within the finite set $\Omega_{\bar{k}+1}$, rather than within the infinite set T. Therefore we only check the feasibility of X_k in the final iteration.

(3) In solving each subproblem, we obtain only inexact optimal solutions of (P_k) and (D_k) up to the tolerance ϵ_k, where $\epsilon_k \to 0$, as $k \to \infty$.

REMARK 2.3. In this paper, we only consider a solution method for the problem (SISDP), in which T is a one-dimensional interval. When T is a box interval in $\mathscr{R}^m$, in which the dimension m is small, for example, $m = 4$ or 5, we can still construct grids of the box interval T. Regarding the set of these grid points as the discretisation set Ω of T, our algorithm can also be used to solve this problem (SISDP) with such

a multi-dimensional index set. However, when the dimension m of T is large, for example, $m \geq 10$, the method is no longer an efficient solution method for solving the problem (SISDP).

3. Convergence analysis

In this section, we shall study the convergence property of the discretisation algorithm.

THEOREM 3.1. *Suppose that $\mathscr{F}_1$ is a compact set. Then there exists an accumulation point of the sequence $\{X_k\}$ generated by the discretisation algorithm and any accumulation point is an optimal solution of (P_0).*

PROOF. Since $X_k \in M_k$, by the compactness of $\mathscr{F}_1$, there exists an accumulation point of the sequence $\{X_k\}$. Let $\bar{X}$ be an accumulation point of the sequence $\{X_k\}$. Then there exists a subsequence $\{X_{k_j}\}$ of $\{X_k\}$ such that $\{X_{k_j}\}$ converges to a point $\bar{X}$. It follows from the closedness of S_+^n that $\bar{X} \in S_+^n$. Suppose that X^* is an optimal solution of (P_0) and $v(P_k)$ and $v(D_k)$ are optimal values of (P_k) and (D_k), respectively. It follows that $X^* \in M_k$ and $v(P_k) \geq C \bullet X^*$. Since

$$\left(a^T z^k + \sum_{t \in E_K} y_k(t) b(t)\right) \geq v(D_k) = v(P_k) \geq C \bullet X_k$$

and

$$\left(a^T z^k + \sum_{t \in E_K} y_k(t) b(t)\right) - C \bullet X_k \leq \epsilon_k,$$

we have $C \bullet X_k + \epsilon_k \geq v(P_k) \geq C \bullet X^*, \forall k$. Thus $C \bullet X_{k_j} + \epsilon_{k_j} \geq C \bullet X^*, \forall j$. As $j \to \infty$, we have

$$C \bullet \bar{X} \geq C \bullet X^*. \tag{3.1}$$

There are two cases to be considered.

Case 1. There exists a subsequence $\{X_{k_{l_j}}\}$ of $\{X_{k_j}\}$ such that

$$B(t_{k_{l_j}}) \bullet X_{k_{l_j}} - b(t_{k_{l_j}}) > 0,$$

that is, the algorithm goes to Step 5 from Step 3 as an infinite loop. Suppose that the algorithm goes to Step 5 at $\bar{k} + 1 = \bar{k}_j$ (as $j \to \infty$). Since $\Omega_{\bar{k}_j} \to T$ (as $j \to \infty$), for each $\xi \in T$, we can find $\xi_{\bar{k}_j} \in \Omega_{\bar{k}_j}$ with $\xi_{\bar{k}_j} \to \xi$ (as $j \to \infty$). Thus

$$B(t_{k_{l_j}}) \bullet X_{k_{l_j}} - b(t_{k_{l_j}}) \geq B(\xi_{\bar{k}_j}) \bullet X_{k_{l_j}} - b(\xi_{\bar{k}_j}).$$

https://doi.org/10.1017/S1446181100013511 Published online by Cambridge University Press

By the compactness of T, we can assume, without loss of generality, that the sequence $\{t_{k_j}\}$ is a convergent one with the limiting point $\bar{t}$. Therefore we obtain

$$B(\bar{t}) \bullet \bar{X} - b(\bar{t}) \geq B(\xi) \bullet \bar{X} - b(\xi).$$

By the construction of $E_{k_{j+1}}$ and $M_{k_{j+1}}$, we have $B(t_{k_j}) \bullet X_{k_{j+1}} - b(t_{k_j}) \leq 0$. So

$$B(\bar{t}) \bullet \bar{X} - b(\bar{t}) \leq 0 \quad \text{and} \quad B(\xi) \bullet \bar{X} - b(\xi) \leq 0.$$

By $X_{k_j} \in M_{k_j}$, we have $A_i \bullet X_{k_j} = a_i$, for $i = 1, 2, \ldots, l$. It follows that $A_i \bullet \bar{X} = a_i$, for $i = 1, 2, \ldots, l$, $\bar{X} \in \mathscr{F}$ and

$$C \bullet X^* \geq C \bullet \bar{X}. \tag{3.2}$$

From (3.1) and (3.2), we have $C \bullet X^* = C \bullet \bar{X}$.

Case 2. There does not exist any subsequence $\{X_{k_{l_j}}\}$ of $\{X_{k_j}\}$ such that

$$B(t_{k_{l_j}}) \bullet X_{k_{l_j}} - b(t_{k_{l_j}}) > 0.$$

Then by the algorithm and the convergence of X_{k_j}, there exists a subsequence $\{X_{k_{q_j}}\}$ such that $B(t_{k_{q_j}}) \bullet X_{k_{q_j}} - b(t_{k_{q_j}}) \leq 0$, that is, the algorithm goes to Step 6 from Step 3 as an infinite loop. Suppose that the algorithm goes to Step 6 at $\bar{k} + 1 = \bar{k}_j$. Since $\Omega_{\bar{k}_j} \to T$ (as $j \to \infty$), for each $\xi \in T$, we can find $\xi_{\bar{k}_j} \in \Omega_{\bar{k}_j}$ with $\xi_{\bar{k}_j} \to \xi$ (as $j \to \infty$). Thus $B(t_{k_{q_j}}) \bullet X_{k_{q_j}} - b(t_{k_{q_j}}) \geq B(\xi_{\bar{k}_j}) \bullet X_{k_{q_j}} - b(\xi_{\bar{k}_j})$. By the compactness of T, we can assume, without loss of generality, that the sequence $\{t_{k_{q_j}}\}$ is a convergent one with the limiting point $\tilde{t}$. Therefore we obtain

$$B(\tilde{t}) \bullet \bar{X} - b(\tilde{t}) \geq B(\xi) \bullet \bar{X} - b(\xi).$$

By the conditions of Case 2, we have $B(t_{k_{q_j}}) \bullet X_{k_{q_j}} - b(t_{k_{q_j}}) \leq 0$. So

$$B(\tilde{t}) \bullet \bar{X} - b(\tilde{t}) \leq 0 \quad \text{and} \quad B(\xi) \bullet \bar{X} - b(\xi) \leq 0.$$

Similarly, we have $\bar{X} \in \mathscr{F}$ and $C \bullet X^* = C \bullet \bar{X}$. Thus, by the conclusions obtained for Cases 1 and 2, the proof is complete.

REMARK 3.1. If the conditions in Remark 2.1 hold, then (SISDP) and (DSISDP) have optimal solutions and their optimal objective function values are equal. It is clear that the discretisation problems (P_k) and (D_k) also satisfy these conditions. Thus their duality gap also vanishes.

https://doi.org/10.1017/S1446181100013511 Published online by Cambridge University Press

4. Numerical results

The discretisation algorithm is implemented under the MATLAB environment and run on a GenuineIntel PIII 800M. We will solve a relaxation of a quadratically constrained semi-infinite quadratic programming problem and a semi-infinite eigenvalue problem. Before presenting numerical results, we make a remark on the partition of the parameter interval $T = [T_1, T_2]$. The Nth partition scheme of the parameter interval $[T_1, T_2]$ is

$$\Omega_N = \left\{ t_i = T_1 + \frac{i(T_2 - T_1)}{2^N} : i = 0, 1, \ldots, 2^N \right\}.$$

It is worth noting that the set of partitions in one iteration is included in the set of partitions in the next iteration.

4.1. A quadratically constrained semi-infinite quadratic programming problem

We show how the combined semi-infinite and semi-definite relaxation problems of quadratically constrained semi-infinite quadratic programming problems can be efficiently solved by using a combination of the discretisation method and the interior point method.

Consider the (Q^2P) problem:

$$\begin{aligned} &\max x^T Q_0 x + 2g_0^T x + \alpha_0 \\ &\text{s.t. } x^T Q(t)x + 2g(t)^T x + \alpha(t) \le b(t), \quad t \in T, \end{aligned}$$

and its equivalent formulation $(Q^2P)_y$:

$$\max y^T P_0 y \quad \text{s.t.} \begin{cases} y^T P(t) y \le b(t), & t \in T, \\ y_0^2 = 1, \\ y = (y_0, x^T)^T \in \mathscr{R}^{n+1}. \end{cases}$$

Recall that the relaxation of (Q^2P) is $(\bar{P})$:

$$\max P_0 \bullet Y \quad \text{s.t.} \begin{cases} E_{00} \bullet Y = 1, \\ P(t) \bullet Y \le b(t), & t \in T, \\ Y \succeq 0. \end{cases} \qquad (\bar{P})$$

REMARK 4.1. Note that problem $(\bar{P})$ is a relaxation of (Q^2P). However, suppose that the following conditions are satisfied: T is a compact set, Q_0 and $Q(t)$, $t \in T$, are positive semi-definite matrices and X^* is an optimal solution of $(\bar{P})$. Then by Theorems 3.1 and 3.2 and Corollary 3.1 of [14], it follows that $X^* e_1$ is an optimal solution of (Q^2P), where e_1 is a unit vector in $\mathscr{R}^{n+1}$ with its first component being 1 and other components zero.

https://doi.org/10.1017/S1446181100013511 Published online by Cambridge University Press

The Lagrangian function of $(Q^2P)_y$ is

$$L(y, \mu, \lambda) = y^T P_0 y + \mu(1 - y_0^2) + \sum_{t \in T} \lambda(t)(b(t) - y^T P(t) y), \quad \lambda \in \Lambda_T.$$

The Lagrangian dual problem of $(Q^2P)_y$ is

$$\min_{\lambda \in \Lambda_T, \mu \in \mathscr{R}} \max_{y \in \mathscr{R}^{n+1}} (y^T P_0 y + \mu(1 - y_0^2) + \sum_{t \in T} \lambda(t)(b(t) - y^T P(t) y). \tag{4.1}$$

For detailed discussion, see [10] and [32]. Note that the inner minimisation over y is bounded from above only if

$$P_0 - \mu E_{00} - \sum_{t \in T} \lambda(t) P(t) \preceq 0 \quad \text{and} \quad \lambda \in \Lambda_T.$$

Thus the Lagrangian dual problem (4.1) of $(Q^2P)_y$ is equivalent to the problem

$$\min \mu + \sum_{t \in T} \lambda(t) b(t) \quad \text{s.t.} \begin{cases} \mu E_{00} + \sum_{t \in T} \lambda(t) P(t) \succeq P_0, \\ \lambda \in \Lambda_T. \end{cases} \tag{$D\,Q^2P$}$$

Obviously, the dual problem for $(\bar{P})$ is also the $(D\,Q^2P)$ problem. Therefore problem $(\bar{P})$ is a relaxation of $(Q^2P)_y$. Since all principal minors of a positive semi-definite matrix are positive semi-definite, the positive semi-definite constraint of $(D\,Q^2P)$ implies

$$\sum_{t \in T} \lambda(t) Q(t) - Q_0 \succeq 0. \tag{4.2}$$

This is where the duality gap may arise, since the standard necessary optimality conditions for (Q^2P) do not require that condition (4.2) holds. Therefore it is, in general, not true that the optimal value of $(D\,Q^2P)_y$ is equal to that of (Q^2P). It is only an upper bound.

We apply the algorithm to solve $(\bar{P})$. Suppose that $T = [1, 3]$, P_0 and $P(t)$, $t \in T$, are symmetric matrices of size n. Each element of P_0 is a number randomly generated between -3 and 3. Each element of $P(t)$ is a qth order polynomial of variable t, each coefficient of the polynomial is a number randomly generated between -3 and 3. Take $N = 11$, that is, when the partition point number of the interval T is greater than $2^{11} = 2048$, we stop the algorithm. Ten test problems for each n are randomly generated and computed. The results on average are summarised in Table 1:

- Time (max) — the average (maximum) time in seconds to compute 10 test problems.

https://doi.org/10.1017/S1446181100013511 Published online by Cambridge University Press

TABLE 1. Numerical results.

n	Time (max)	No. of SDP (max)	Iteration (max)
10	0039.3(0040.2)	09.8(14.0)	08.4(11.3)
20	0154.5(0156.8)	10.6(15.0)	09.7(13.2)
40	0618.6(0635.4)	11.7(15.0)	11.8(20.3)
60	1419.5(1441.4)	11.8(15.0)	13.0(15.9)
80	2544.9(2720.1)	09.6(15.0)	13.2(28.8)
100	3975.2(4016.0)	10.5(12.0)	15.4(18.2)

- No. of (SDP) (max) — the average (maximum) number of (SDP)s solved in 10 test problems.
- Iteration (max) — the average (maximum) number of the average iteration numbers to compute each (SDP) in 10 test problems.

For $q = 6$, we obtain the following results.

4.2. A semi-infinite eigenvalue problem Many problems in mechanics and systems analysis can often be expressed as

$$\frac{d}{dt}X(t) = u(t)A(t) \quad \text{s.t.} \begin{cases} X(0) = A_0, \\ u(t) \geq 0, \quad t \in T, \end{cases}$$

where T is an interval in $\mathscr{R}$ and A_0 and $A(t)$, $t \in T$, are all fixed matrices in S^n. We can calculate the solution of the differential equation as follows:

$$X(u) = A_0 + \int_T u(t)A(t)\,dt.$$

Our aim is to seek a function $u(t)$, $t \in T$, such that the maximum eigenvalue of $X(u)$ is minimised, that is,

$$\lambda^* = \min_{u(t)\geq 0, t\in T} \lambda_{\max}\{X(u)\}, \tag{4.3}$$

where $\lambda_{\max}\{X(u)\}$ denotes the maximum eigenvalue of $X(u)$.

It is clear that this problem can be reformulated as a combined semi-infinite and semi-definite programming problem (P_λ):

$$\min \lambda \quad \text{s.t.} \begin{cases} \lambda I - A_0 - \int_T u(t)A(t)\,dt \succeq 0, \\ u(t) \geq 0, \quad t \in T, \ \lambda \in \mathscr{R}. \end{cases} \tag{P_λ}$$

https://doi.org/10.1017/S1446181100013511 Published online by Cambridge University Press

TABLE 2. Numerical results.

n	Time (max)	No. of SDP(max)	Iteration (max)
10	0038.5(0038.9)	4.4(5.0)	08.0(14.8)
20	0152.3(0153.9)	4.4(6.0)	08.3(23.2)
40	0606.7(0611.7)	4.2(5.0)	11.4(19.0)
60	1402.3((1483.5)	4.1(5.0)	10.5(18.6)
80	2474.1(2482.6)	4.2(5.0)	07.9(12.0)
100	3886.5(3894.2)	4.1(5.0)	07.8(10.0)

Suppose that $A(\cdot)$ is a continuous function of variable t. The Lagrangian function of (P_λ) is

$$L(X, \lambda, \mu) = \lambda - \left[\lambda I - A_0 - \int_T u(t)A(t)\,dt\right] \bullet X.$$

The Lagrangian dual problem is

$$\begin{aligned} d^* &= \max_{X \succeq 0} \min_{u(t)\geq 0, t\in T, \lambda \in \mathscr{R}} \left\{\lambda - \left[\lambda I - A_0 - \int_T u(t)A(t)\,dt\right] \bullet X\right\} \\ &= \max_{X \succeq 0} \min_{u(t)\geq 0, t\in T, \lambda \in \mathscr{R}} \left\{A_0 \bullet X + (1 - I \bullet X)\lambda + \int_T u(t)A(t) \bullet X\,dt\right\}. \end{aligned}$$

Thus, by the continuity of $A(\cdot)$, the Lagrangian dual problem of (P_λ) can be rewritten as (D_λ):

$$\max A_0 \bullet X \quad \text{s.t.} \begin{cases} A(t) \bullet X \geq 0, & t \in T, \\ I \bullet X = 1, \\ X \succeq 0. \end{cases} \qquad (D_\lambda)$$

If the optimal solution $\hat{X}$ of (D_λ) exists, then $\lambda^* = A_0 \bullet \hat{X}$. Thus, the minimax eigenvalue λ^* of the solution $X(u)$ of the differential equation can be obtained by computing the optimal value of the Lagrangian dual problem of (D_λ). It follows from dual feasibility that the eigenvectors for the optimal eigenvalues are found as the columns of $\hat{X}$.

Let $T = [1, 3]$ and let A_0 and $A(t)$, $t \in T$, be symmetric matrices of size n. Each element of A_0 is a number randomly generated between -3 and 3. Each element of $A(t)$ is a qth order polynomial of variable t, of which each coefficient is a number randomly generated between -3 and 3. Take $N = 11$, that is, when the number of partition point of the interval T is greater than $2^{11} = 2048$, we stop the algorithm. For $q = 6$, 10 test problems with each n are randomly generated and computed. The results on average are summarised in Table 2.

https://doi.org/10.1017/S1446181100013511 Published online by Cambridge University Press

From Tables 1 and 2, we know that the average number of (P_k) computed and the average number of iterations to compute (P_k) are independent of the size n of matrices. Obviously, when the size n of matrices increases, the average time to compute each problem also increases. We also observe that the average number of iterations to compute all (P_k) for semi-infinite eigenvalue problems is less than that to compute all (P_k) for relaxations of quadratically constrained semi-infinite quadratic programming problems corresponding to the same n. We note that feasible solutions of semi-infinite eigenvalue problems are required to satisfy the equality equation $I \bullet X = 1$, while feasible solutions of relaxations of quadratically constrained semi-infinite quadratic programming problems are required to satisfy the equality equation $E_{00} \bullet X = 1$.

5. Conclusion

In this paper, we proposed a discretisation algorithm with an adaptive scheme for solving a class of combined semi-infinite and semi-definite programming problems. The proposed scheme is very general and flexible. In the discretisation algorithm, we only find an inexact optimal solution of each subproblem. When the partition number of the parametric interval T increases, it is not necessary to add any inequality constraint. We may regard this case as dropping unnecessary constraints, that is, our discretisation algorithm allows us to explore the possibility of combining the use of "approximate solutions" for each subproblem and the idea of "dropping unnecessary constraints" in each iteration. Semi-infinite eigenvalue problems and relaxation of quadratically constrained semi-infinite quadratic programming problems were effectively solved by the algorithm.

Acknowledgement

This research is partially supported by the Research Committee of The Hong Kong Polytechnic University and the National Natural Science Foundation of China.

References

[1] F. Alizadeh, "Interior point methods in semidefinite programming with applications to combinatorial optimization", *SIAM J. Optim.* **5** (1995) 13–51.

[2] E. J. Anderson and A. S. Lewis, "An extension of the simplex method for semi-infinite programming", *Math. Program.* **44** (1989) 247–269.

[3] E. J. Anderson and P. Nash, *Linear programming in finite-dimensional spaces* (John Wiley & Sons, Chichester, 1987).

[4] R. J. Duffin, R. G. Jeroslow and L. A. Karlovitz, "Duality in semi-infinite linear programming", in *Semi-infinite programming and applications* (eds. A. V. Fiacco and K. O. Kortanek), Lecture Notes in Economics and Mathematical Systems **215**, (Springer, Berlin, 1983) 50–62.

[5] M. A. Goberna and M. A. Lopez, *Linear semi-infinite optimization* (John Wiley & Sons, Chichester, 1998).

[6] C. Helmberg, F. Rendl, R. J. Vanderbei and H. Wolkowicz, "An interior-point method for semidefinite programming", *SIAM J. Optim.* **6** (1996) 342–361.

[7] R. Hettich and K. O. Kortanek, "Semi-infinite programming: theory, methods, and applications", *SIAM Rev.* **35** (1993) 380–429.

[8] F. Jarre, "An interior-point method for minimizing the maximum eigenvalue of a linear combination of matrices", *SIAM J. Control Optim.* **31** (1993) 1360–1377.

[9] V. Jeyakumar and J. Gwinner, "Inequality systems and optimization", *J. Math. Anal. Appl.* **159** (1991) 51–71.

[10] V. Jeyakumar and H. Wolkowicz, "Zero duality gaps in infinite-dimensional programming", *J. Optim. Theory Appl.* **67** (1990) 87–108.

[11] A. Kaplan and R. Tichatschke, "Proximal interior point methods for convex semi-infinite programming", *Optim. Methods Softw.* **15** (2001) 87–119.

[12] J. L. Kelley and I. Namtoka, *Linear topological spaces* (Springer, New York, 1963).

[13] K. O. Kortanek and Q. Zhang, "Perfect duality in semi-infinite and semidefinite programming", *Math. Program.* **91** (2001) 127–144.

[14] S. J. Li, K. L. Teo, X. Q. Yang and S. Y. Wu, "Robust envelope-constrained filter with orthonormal bases and semi-definite and semi-infinite programming", *IEEE Trans. Circuits Sys.-I*, submitted.

[15] M. V. Nayakkankuppam and M. L. Overton, "Conditioning of semidefinite programs", *Math. Program. Series A* **85** (1999) 525–540.

[16] Y. Nesterov and A. Nemirovskii, *Interior-point polynomial algorithms in convex programming* (SIAM, Philadelphia, 1994).

[17] P. M. Pardalos, "Quadratic programming with one negative eigenvalue is NP-hard", *J. Global Optim.* **1** (1991) 15–22.

[18] G. Pataki, "Cone-LP's and semidefinite programs: Geometry and a simplex-type method", in *Integer programming and combinational optimization (Vancouver, BC, 1996)*, (Springer, 1996) 162–174.

[19] E. Polak, *Optimization: algorithm and consistent applications* (Springer, New York, 1997).

[20] F. A. Potra and R. Sheng, "A superlinearly convergent primal-dual infeasible-interior-point algorithm for semidefinite programming", *SIAM J. Optim.* **8** (1998) 1007–1028.

[21] M. Ramana, L. Tuncel and H. Wolkowicz, "Strong duality for semidefinite programming", *SIAM J. Optim.* **7** (1997) 641–662.

[22] R. Reemtsen, "Some outer approximation methods for semi-infinite optimization problems", *J. Comput. Appl. Math.* **53** (1994) 87–108.

[23] R. Reemtsen and J. J. Ruckmann, *Semi-infinite programming* (Kluwer Academic Publisher, Drodrecht, 1998).

[24] F. Rendl, R. J. Vanderbei and H. Wolkowicz, "Max-min eigenvalue problems, primal-dual interior point algorithms, and trust region subproblems", *Optim. Methods Softw.* **5** (1995) 1–16.

[25] K. L. Teo, X. Q. Yang and L. S. Jennings, "Computational discretization algorithms for functional inequality constrained optimization", *Ann. Oper. Res.* **98** (2000) 215–234.

[26] M. J. Todd, "Semidefinite optimization", *Acta Numer.* **10** (2001) 515–560.

[27] L. Vandenberghe and S. Boyd, "Semidefinite programming", *SIAM Rev.* **38** (1996) 49–95.

[28] S. Vavasis, *Nonlinear optimization* (Oxford University Press, New York, 1991).

[29] H. Wolkowicz, R. Saigal and L. Vandenberghe, *Handbook of semidefinite programming theory, algorithms, and applications* (Kluwer Academic Publishers, Dordrecht, 2000).

[30] S. Y. Wu, S. C. Fang and C. J. Lin, "Relaxed cutting plane method for solving linear semi-infinite programming problems", *J. Optim. Theory Appl.* **99** (1998) 759–779.

[31] S. Y. Wu, S. C. Fang and C. J. Lin, "Solving quadratic semi-infinite programming problems by using relaxed cutting plane scheme", *J. Comput. Appl. Math.* **129** (2001) 89–104.

[32] X. Q. Yang and K. L. Teo, "Nonlinear Lagrangian functions and applications to semi-infinite programs", *Ann. Oper. Res.* **103** (2001) 235–250.