Separated Continuous Conic Programming: Strong Duality and an Approximation Algorithm*

Xiaoqing Wang† Shuzhong Zhang‡ David D. Yao§

January, 2006; revised December 2007

Abstract

Motivated by recent applications in robust optimization and in sign-constrained linear-quadratic control, we study in this paper a new class of optimization problems called *separated continuous conic programming* (SCCP). Focusing on a symmetric primal-dual pair, we develop a strong duality theory for the SCCP. Our idea is to use discretization to connect the SCCP and its dual to two ordinary conic programs. We show if the latter are strongly feasible and with finite optimal values, a condition that is readily verifiable, then the strong duality holds for the SCCP. This approach also leads to a polynomial-time approximation algorithm that solves the SCCP to any required accuracy.

Keywords: continuous optimization, conic programming, duality, approximation algorithm.

MSC subject classification: 90C25, 49M25.

*This research is supported in part by Hong Kong RGC Earmarked Grant CUHK4242/04E.

†Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Shatin, Hong Kong (xqwang@se.cuhk.edu.hk).

‡Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Shatin, Hong Kong (zhang@se.cuhk.edu.hk).

§Department of Industrial Engineering and Operations Research, Columbia University, New York, USA (yao@columbia.edu).

1 Introduction

The object of our study here is the following optimization problem, which we shall call *separated continuous conic programming* (SCCP):

$$
\begin{array}{lll}
(SCCP) & \max & \int_0^T [(\gamma + (T-t)c)' u(t) + d' x(t)] dt \\
& \text{s.t.} & \alpha + ta - \int_0^t Gu(s) ds - Fx(t) \in \mathcal{K}_1, \\
& & b - Hu(t) \in \mathcal{K}_2, \\
& & u(t) \in \mathcal{K}_3,\ x(t) \in \mathcal{K}_4,\ t \in [0, T],
\end{array} \tag{1}
$$

where the control and state variables (both are decision variables), $u(t)$ and $x(t)$, are vectors of bounded measurable functions of time $t \in [0, T]$, with $T > 0$ being a given constant; $\mathcal{K}_i,\ i = 1, 2, 3, 4$, are closed convex cones in the Euclidean space with appropriate dimensions; γ, c, d, α, a and b are vectors; G, F, H are matrices; and the superscript $'$ denotes the transpose operation.

In the special case when all the closed convex cones are non-negative orthants, the SCCP reduces to what is known as *separated continuous linear programming* (SCLP):

$$
\begin{array}{lll}
(SCLP) & \max & \displaystyle\int_0^T [(\gamma + (T-t)c)' u(t) + d' x(t)] dt \\
& \text{s.t.} & \displaystyle\int_0^t Gu(s) ds + Fx(t) \le \alpha + ta, \\
& & Hu(t) \le b, \\
& & u(t) \ge 0,\ x(t) \ge 0,\ t \in [0, T].
\end{array} \tag{2}
$$

The SCLP is a special kind of *continuous linear program* (CLP). Specifically (for both SCLP and SCCP), the qualifier, “separated,” alludes to the fact that there are two main constraints in the problem, with the first one involving an integration on the control variable $u(t)$ over time, whereas the second one exercising an instantaneous restraint on the control. In applications, the first constraint typically accounts for some type of “flow balance”: total output cannot exceed total input; whereas the second constraint represents the capacity limit of resources (refer to the example in §5.3). Note that in both SCCP and SCLP, $x(0)$ can be either a decision variable or a given quantity (in the latter case, it must satisfy the required constraints). Regardless, it has no impact on the optimal objective value; refer to Remark 4.1.

The SCLP was first introduced by Anderson in [1] as a model for job-shop scheduling. In recent years, SCLP has attracted renewed research interests mainly due to what is called the fluid network model (see §5.3 for a simple example), which is known to be a functional strong-law-of-large-numbers limit of many stochastic networks (see, [14]; also, [11]), and the dynamic control of resources in a fluid network can be formulated and solved as instances of the SCLP (e.g., [10, 15]).

Our interest in extending the SCLP to the more general SCCP setting is motivated by two factors. First, the SCCP has more modeling power; e.g., in the fluid network model, if the objective function is quadratic, instead of linear, then the scheduling problem cannot be handled by SCLP: it becomes an SCCP. There are other applications that necessitate the extension from SCLP to SCCP, e.g., in robust optimization; refer to §5.1. The second factor that motivates our study is computational: we want to develop practical algorithms that can efficiently solve SCCP problems. Notably, such algorithms are relatively rare even in the special case of SCLP; refer to the literature review that follows.

In searching for a practical and efficient algorithm, we resort to, quite naturally, (a) discretization — discretize the SCCP into an ordinary conic program that can be solved by standard algorithms, and (b) closing the duality gap — between the original SCCP and its dual. In pursuing (b), we must develop a strong duality theory for the SCCP.

Here is a quick summary of our main results. Through discretization, we connect the SCCP and its dual to two ordinary conic programs; and some mild (and verifiable) conditions on the latter is shown to imply that strong duality holds for the SCCP. Furthermore, the two conic programs also provide an explicit bound on the duality gap, which can be closed by increasing the granularity of the discretization. This leads to a polynomial-time approximation algorithm that solves the SCCP to any required accuracy.

The paper is organized as follows. In the rest of this introduction, we break into two subsections, overviewing the related literature and the preliminaries of conic programming. In §2, we discretize the SCCP and its dual into two ordinary conic programs, and bring out their relations. In §3, we establish the strong duality result for the SCCP. This leads to a polynomial-time approximation algorithm with an explicit error bound, detailed in §4. In §5, we present two new applications of SCCP, in robust optimization and in sign-constrained linear quadratic control, followed by a numerical example. We conclude in §6 pointing out some possible extensions and unresolved issues.

1.1 Related Literature

Bellman [7, 8] first introduced the so-called continuous linear programming, in which the decision variables (control and states) are all functions of time, over a planning horizon $[0, T]$, subject to polyhedral constraints, to be satisfied at every time instant. The model has wide-ranging applications, but is notoriously difficult to solve in general. Later, Anderson [1] introduced SCLP, as highlighted above in (2), a special case of continuous linear programming. Anderson, Nash and Perold [3] studied the properties of the extreme solutions of the SCLP, based on which Anderson and Philpott [5] developed a simplex type of algorithm for a network-based SCLP. Refer to Anderson and Nash [4] for a comprehensive survey on these and related topics.

Pullan ([20, 21, 22, 23, 24, 25, 26]) in a series of papers systematically developed a duality theory and solution algorithms for the SCLP. Notably, Pullan chose to focus on a *non-symmetric* primal-dual pair — while the primal is an SCLP, the dual is not. With this choice of a non-symmetric dual, strong duality readily follows. Furthermore, Pullan showed there exists an optimal solution — provided optimal solutions do exist — that is piecewise constant in the control variable with a finite set of breakpoints over the interval $[0, T]$ under some conditions. Based on this, Pullan designed algorithms that attempt to find the correct (optimal) set of breakpoints. A related work is due to Luo and Bertsimas [18], who incorporated finding such a partition of $[0, T]$ as a part of the decision. This results in a (non-convex) quadratic program. Another related work is Shapiro [27], where a duality theory is developed for conic linear program in general, including continuous linear programming as a special case. Sufficient conditions are provided so as to guarantee the strong duality relation in this general framework. As the setup is quite abstract, the duality theory does not immediately lead to any implementable algorithms.

Two recent algorithmic developments in SCLP relate closely to our work. Weiss [30] developed a simplex-based algorithm to solve the SCLP. Assuming a non-degeneracy condition holds, the approach always finds an exact optimal solution in finite steps, albeit requiring typically a large amount of computation. In contrast to Pullan's non-symmetric dual, Weiss focused on a symmetric dual — the dual is itself an SCLP; and via the algorithm, he constructively established the strong duality for the SCLP. Fleischer and Sethuraman [16] developed a polynomial-time approximation algorithm for the SCLP in a network flow context. They used a fixed partition of $[0, T]$, specifically designed to meet the accuracy requirement on the solution.

Like Weiss's approach, we focus on a symmetric dual. Like the Fleischer-Sethuraman algorithm, ours is an approximation algorithm, and we use a fixed partition of $[0, T]$ — indeed, we use the simplest possible partition, namely, an even partition (i.e., the segments are all equal in length). On the other hand, the problem we solve is the more general SCCP. We establish the strong duality for the SCCP, as Weiss did for the SCLP, but using a very different approach, via discretization and an explicit bound on the duality gap. Furthermore, our approximation algorithm readily solves the SCLP (as a special case) requiring neither the non-degeneracy condition of Weiss's simplex-based algorithm nor the network flow formalism of Fleischer and Sethuraman. Indeed, what we require is some very minimal and easily verifiable condition on the SCCP, that two related ordinary conic programs are strongly feasible, with finite optimal solutions; and we provide easily computable error bounds for the solutions from our approximation algorithm.

1.2 Preliminaries on Conic Programming

Since our study concerns continuous-time optimization models beyond the confines of polyhedra, for the benefit of the reader we summarize here some basic facts of conic programming, most of which will be used later in the paper, often without explicit reference. For further details of the subject, refer to Nesterov and Nemirovski [19].

1. A set $\mathcal{K} \subseteq \Re^n$ is a *cone* if for any $x \in \mathcal{K}$ and scalar $\lambda \geq 0$, it follows that $\lambda x \in \mathcal{K}$. In this paper, we are primarily interested in closed convex cones. We use 'int $\mathcal{K}$' to denote the interior of the cone $\mathcal{K}$.

2. For a cone $\mathcal{K}$, its dual is:

$$\mathcal{K}^* = \{y \in \Re^n \mid x'y \geq 0, \ \forall x \in \mathcal{K}\}.$$

3. For a convex cone $\mathcal{K}$, its dual cone $\mathcal{K}^*$ is a closed convex cone; and the dual of $\mathcal{K}^*$ is the closure of $\mathcal{K}$, i.e., $\mathcal{K}^{**} = \mathrm{cl}\ \mathcal{K}$, which is the celebrated bi-polar theorem for convex cones.

4. The following three types of closed convex cones are popular due to their appealing properties (e.g., the so-called self-scaledness property) and their versatility in modeling: (i) the non-negative orthants, e.g. $\Re^n_+ = \{x \in \Re^n \mid x \geq 0\}$; (ii) the second-order cones, e.g. $\{(t,x) \in \Re^n \mid t \geq \|x\|\}$; (iii) the cone of positive semidefinite matrices, e.g. $\{X \in \mathcal{S}^{n\times n} \mid X \succeq 0\}$, where $\mathcal{S}^{n\times n}$ denotes the space of n-by-n real symmetric matrices.

5. The following are elementary properties regarding nonempty closed convex cones $\mathcal{K}$, $\mathcal{K}_1$ and $\mathcal{K}_2$.

(i) If $\alpha \in \mathrm{int}\ \mathcal{K}$, scalar $\lambda > 0$, then $\lambda\alpha \in \mathrm{int}\ \mathcal{K}$.

(ii) $\mathrm{int}\ (\mathcal{K}_1 \times \mathcal{K}_2) = \mathrm{int}\ \mathcal{K}_1 \times \mathrm{int}\ \mathcal{K}_2$.

(iii) If $\alpha_1 \in \mathcal{K}, \alpha_2 \in \mathrm{int}\ \mathcal{K}, \alpha_3 = \lambda\alpha_1 + (1-\lambda)\alpha_2, \lambda \in [0,1)$, then $\alpha_3 \in \mathrm{int}\ \mathcal{K}$.

(iv) $\mathcal{K}_1 \times \mathcal{K}_2$ is also convex and closed.

(v) $(\mathcal{K}_1 \times \mathcal{K}_2)^* = \mathcal{K}_1^* \times \mathcal{K}_2^*$.

6. A standard conic programming (CP) problem can be expressed as follows:

$$\begin{array}{lll} (CP) & \min & c'x \\ & \text{s.t.} & Ax - b \in \mathcal{K} \end{array}$$

where A is an $m \times n$ matrix, $b \in \Re^m$, $c \in \Re^n$ and $\mathcal{K}$ is a closed convex cone in $\Re^m$.

It is well known that any convex optimization problem can be formulated in the conic form as in the above. The dual of (CP) is:

$$\begin{array}{lll} (CP^*) & \max & b'y \\ & \text{s.t.} & A'y = c \\ & & y \in \mathcal{K}^*. \end{array}$$

7. Problem (CP) is said to be *strongly feasible* if there exists x, such that $Ax - b \in \text{int } \mathcal{K}$. Such x is called a *strongly feasible* solution of (CP). Similarly, (CP^*) is strongly feasible if there exists y, such that $A'y = c, y \in \text{int } \mathcal{K}^*$, and such y is called a strongly feasible solution for (CP^*). A strongly feasible conic program is also said to satisfy the *Slater condition*.

8. The following facts are well known:

(i) The dual of (CP^*) is (CP).

(ii) The *weak duality* holds between (CP) and (CP^*). That is, if x is a feasible solution of (CP), and y is a feasible solution of (CP^*), then $c'x \geq b'y$.

(iii) If (CP) and (CP^*) are both strongly feasible, then they are solvable and their optimal solution sets are non-empty and bounded. Moreover, their optimal objective values coincide. In other words, in this case, the *strong duality* holds.

2 A Symmetric Dual and Discretization

The dual of $(SCCP)$ in (1) that we shall focus on is the following problem, which is itself an SCCP:

$$\begin{array}{lll} (SCCP^*) & \min & \int_0^T [(\alpha + (T-t)a)'p(t) + b'q(t)]dt \\ & \text{s.t.} & \int_0^t G'p(s)ds + H'q(t) - (\gamma + tc) \in \mathcal{K}_3^*, \\ & & F'p(t) - d \in \mathcal{K}_4^*, \\ & & p(t) \in \mathcal{K}_1^*, \ q(t) \in \mathcal{K}_2^*, t \in [0,T], \end{array} \tag{3}$$

where $\mathcal{K}_i^*$ is the dual cone of $\mathcal{K}_i$, $i = 1, 2, 3, 4$, and the dual variables, $p(t)$ and $q(t)$, are bounded measurable functions.

The above dual can be derived in a similar way as in linear programming (e.g., Chvátal [13]). Strictly speaking, the time in the dual problem should run backward, i.e., from T to 0. In (3), however, we have redefined t to be $T-t$, so as to achieve a certain symmetry. In particular, observe that the dual problem in (3) is also an SCCP, and its dual is exactly the primal problem in(1). In other words, the SCCP primal-dual pair is just like what's in linear programming. The following weak duality is readily shown.

Proposition 2.1 *Weak duality holds for SCCP. That is, if $(u(t), x(t))$ is a feasible solution of $(SCCP)$ and $(p(t), q(t))$ is a feasible solution of $(SCCP^*)$, then*

$$\int_0^T [(\gamma + (T-t)c)'u(t) + d'x(t)]dt \leq \int_0^T [(\alpha + (T-t)a)'p(t) + b'q(t)]dt.$$

What we want to accomplish in this paper is to develop the strong duality theory for SCCP, parallel to point 8(iii) of §1.2 in the case of conic programming. In particular, we will show that the strong duality between $(SCCP)$ and $(SCCP^*)$ is determined by two ordinary conic programs, which are discretized versions of $(SCCP)$ and $(SCCP^*)$. But first, we need the following notation and conventions (which mostly follow what's used in [20]).

Notation and Conventions:

- By default, all vectors are column vectors. One exception is when we denote the solutions to the SCCP and its dual (or their variations) as (u, x) and (p, q), we mean $(u', x')'$ and $(p', q')'$.
- $\pi = \{t_0, \ldots, t_m\}$ denotes a partition of $[0, T]$ into m segments:

$$0 = t_0 < t_1 < \cdots < t_m = T.$$

- Given a partition $\pi = \{t_0, \ldots, t_m\}$, and a vector $\hat{f} := (\hat{f}(t_0), \hat{f}(t_1), \ldots, \hat{f}(t_m))$, where $\hat{f}(\cdot)$ is a right continuous function, the following (continuous) function

$$f(t) = (\frac{t_i - t}{t_i - t_{i-1}})\hat{f}(t_{i-1}) + (\frac{t - t_{i-1}}{t_i - t_{i-1}})\hat{f}(t_i), \text{ for } t \in [t_{i-1}, t_i], i = 1, \ldots, m$$

  is called a piecewise linear extension of $\hat{f}$; whereas the following (right-continuous) function

$$f(t) = \begin{cases} \hat{f}(t_{i-1}), & t \in [t_{i-1}, t_i), \text{ for } i = 1, \ldots, m \\ \hat{f}(t_{m-1}), & t = T, \end{cases}$$

  is called a piecewise constant extension of $\hat{f}$.
- When $(u(t), x(t))$ is a feasible solution to $(SCCP)$, with $u(t)$ being piecewise constant and $x(t)$ piecewise linear, we assume $u(t)$ is right continuous, and $x(t)$ is continuous, and the pieces of both u and x correspond to a common partition.

Nomenclature:

(i) $(SCCP)$ and $(SCCP^*)$: the SCCP problem in (1) and its dual in (3).

(ii) (CP_1) and (CP_2): the two key conic programs in (10) and (11) that determine the strong duality between $(SCCP)$ and $(SCCP^*)$.

(iii) $(CP_1(\pi))$ and $(CP_2(\pi))$: the discretized conic programming problems of $(SCCP)$ and $(SCCP^*)$ under a partition π.

(iv) (P^*) denotes the dual of a problem (P); and $v(P)$ denotes the optimal objective value of (P).

We start with introducing the following discretization of $(SCCP)$ based on a partition of $[0,T]$, $\pi = \{t_0, \ldots, t_m\}$:

$$
\begin{array}{lll}
(CP_1(\pi)) & \max & \sum_{i=1}^{m} \left((\gamma + (T - \frac{t_i + t_{i-1}}{2})c)'\hat{u}(t_{i-1}) + d'[\hat{x}(t_i) + \hat{x}(t_{i-1})]\frac{t_i - t_{i-1}}{2} \right) \\
& \text{s.t.} & \alpha + t_i a - [G\hat{u}(t_0) + \cdots + G\hat{u}(t_{i-1}) + F\hat{x}(t_i)] \in \mathcal{K}_1,\ i = 1, 2, \ldots, m; \\
& & (t_i - t_{i-1})b - H\hat{u}(t_{i-1}) \in \mathcal{K}_2,\ i = 1, \ldots, m; \\
& & \hat{u}(t_{i-1}) \in \mathcal{K}_3,\ \hat{x}(t_i) \in \mathcal{K}_4,\ i = 1, \ldots, m.
\end{array} \tag{4}
$$

In addition, *assume* (i.e., $\hat{x}(t_0)$ is given, not a decision variable):

$$
\alpha - F\hat{x}(t_0) \in \mathcal{K}_1, \qquad \hat{x}(t_0) \in \mathcal{K}_4. \tag{5}
$$

There are some immediate points to note here.

- Since the Riemann integrations in the formulation of $(SCCP)$ can be discretized in different ways, our choice leading to $(CP_1(\pi))$ is a specific one. On the control side, $\hat{u}(t)$ is taken to be the constant $\hat{u}(t_{i-1})$ on the interval $[t_{i-1}, t_i]$, while the state variable $\hat{x}(t)$ is taken to be $(\hat{x}(t_{i-1}) + \hat{x}(t_i))/2$ on the same interval.
- For simplicity of the analysis to follow, $\hat{u}(t_{i-1})$ is equivalent to $u(t_{i-1})(t_i - t_{i-1})$, while $\hat{x}(t_i)$ carries the same identity as $x(t_i)$. Therefore, the scales of $\hat{u}$ and $\hat{x}$ are different.

Clearly, $(CP_1(\pi))$ is a conic program.

Remark 2.2 *Although the value of $x(0)$ can be set either as a decision variable or as a given parameter in $(SCCP)$, which is an interesting property to be proven later (see Remark 4.1), in $(CP_1(\pi))$ we chose $\hat{x}(t_0)$ as given. This is because if $x(0)$ is given (i.e., not a decision variable) in $(SCCP)$, we can simply set $\hat{x}(t_0) = x(0)$, and expect the constraints in (5) to hold automatically. On the other hand, if $x(0)$ is a decision variable in $(SCCP)$, then we can obtain $\hat{x}(t_0)$ from solving another conic program, the one in (20) below.*

There is a close connection between $(CP_1(\pi))$ and $(SCCP)$: Any feasible solution of $(CP_1(\pi))$ can be turned into a feasible solution to $(SCCP)$ by applying the piecewise constant and piecewise linear extensions as specified above (to $\hat{u}$ and $\hat{x}$, respectively). Conversely, any feasible solution to $(SCCP)$ that is piecewise constant in u and piecewise linear in x can be turned into a feasible solution to $(CP_1(\pi))$. Furthermore, in both cases, the objective values of the two problems are equal. The precise details are presented in the following lemma.

Lemma 2.3

(i) *Suppose* $(\hat{u}, \hat{x})$ *is a feasible solution to* $(CP_1(\pi))$, *with*

$$\hat{u} = (\hat{u}(t_0), \ldots, \hat{u}(t_{m-1})), \qquad \hat{x} = (\hat{x}(t_1), \ldots, \hat{x}(t_m)).$$

Let $u(t)$ *be the piecewise constant extension of* $(\frac{\hat{u}(t_0)}{t_1-t_0}, \ldots, \frac{\hat{u}(t_{m-1})}{t_m-t_{m-1}})$, *and* $x(t)$ *be the piecewise linear extension of* $(\hat{x}(t_0), \hat{x})$. *Then,* $(u(t), x(t))$ *is a feasible solution to* $(SCCP)$, *and*

$$\begin{aligned}
& \int_0^T [(\gamma + (T-t)c)'u(t) + d'x(t)]dt \\
= & \sum_{i=1}^m \left((\gamma + (T - \frac{t_i + t_{i-1}}{2})c)'\hat{u}(t_{i-1}) + d'[\hat{x}(t_i) + \hat{x}(t_{i-1})] \frac{t_i - t_{i-1}}{2} \right).
\end{aligned} \tag{6}$$

(ii) *Suppose* $(u(t), x(t))$ *is a feasible solution to* $(SCCP)$, *with* $u(t)$ *being piecewise constant and* $x(t)$ *piecewise linear. Then,* $\hat{u}(t_{i-1}) = (t_i - t_{i-1})u(t_{i-1})$ *and* $\hat{x}(t_i) = x(t_i)$, $i = 1, \ldots, m$, *form a feasible solution to* $(CP_1(\pi))$, *and the equality in (6) holds.*

Proof. We show part (i) of the lemma; part (ii) is similarly verified. We have,

$$u(t) = \begin{cases} \frac{\hat{u}(t_{i-1})}{t_i - t_{i-1}}, & t \in [t_{i-1}, t_i), i = 1, \ldots, m \\ \frac{\hat{u}(t_{m-1})}{t_m - t_{m-1}}, & t = T \end{cases}$$

and

$$x(t) = \frac{t_i - t}{t_i - t_{i-1}} \hat{x}(t_{i-1}) + \frac{t - t_{i-1}}{t_i - t_{i-1}} \hat{x}(t_i), \quad t \in [t_{i-1}, t_i].$$

Clearly, $u(t) \in \mathcal{K}_3$ and $x(t) \in \mathcal{K}_4$ since $\mathcal{K}_3$ and $\mathcal{K}_4$ are convex cones.

Next, for any $t \in [t_{i-1}, t_i)$, $i = 2, \ldots, m$, we have

$$\begin{aligned}
& \alpha + ta - \int_0^t Gu(s)ds - Fx(t) \\
= & \alpha + ta - \left[G\hat{u}(t_0) + \cdots + G\hat{u}(t_{i-2}) + G\frac{\hat{u}(t_{i-1})}{t_i - t_{i-1}}(t - t_{i-1}) \right] \\
& -F \left[\frac{t_i - t}{t_i - t_{i-1}} \hat{x}(t_{i-1}) + \frac{t - t_{i-1}}{t_i - t_{i-1}} \hat{x}(t_i) \right] \\
= & \frac{t - t_{i-1}}{t_i - t_{i-1}} \Big(\alpha + t_i a - [G\hat{u}(t_0) + \cdots + G\hat{u}(t_{i-1}) + F\hat{x}(t_i)] \Big) \\
& + \frac{t_i - t}{t_i - t_{i-1}} \Big(\alpha + t_{i-1}a - [G\hat{u}(t_0) + \cdots + G\hat{u}(t_{i-2}) + F\hat{x}(t_{i-1})] \Big) \\
\in & \mathcal{K}_1,
\end{aligned}$$

following the first constraint in $(CP_1(\pi))$ and the conic property of $\mathcal{K}_1$. The case for $i = 1$ (i.e., $t \in [0, T)$) and the case for $t = t_m = T$ are similarly verified.

Furthermore, for any $t \in [t_{i-1}, t_i),\ i = 1, \ldots, m$, we have

$$b - Hu(t) = b - H\frac{\hat{u}(t_{i-1})}{t_i - t_{i-1}} = \frac{1}{t_i - t_{i-1}}((t_i - t_{i-1})b - H\hat{u}(t_{i-1})) \in \mathcal{K}_2,$$

and the case for $t = t_m = T$ is similarly verified.

Finally, we can write the objective function of $(SCCP)$ as follows:

$$\begin{aligned} & \int_0^T [(\gamma + (T - t)c)' u(t) + d' x(t)] dt \\ = & \sum_{i=1}^m \int_{t_{i-1}}^{t_i} [(\gamma + (T - t)c)' u(t) + d' x(t)] dt. \end{aligned}$$

Since

$$\begin{aligned} \int_{t_{i-1}}^{t_i} (\gamma + (T - t)c)' u(t) dt &= \int_{t_{i-1}}^{t_i} (\gamma + (T - t)c)' \frac{\hat{u}(t_{i-1})}{t_i - t_{i-1}} dt \\ &= (\gamma + (T - \frac{t_i + t_{i-1}}{2})c)' \hat{u}(t_{i-1}), \end{aligned}$$

and

$$\begin{aligned} \int_{t_{i-1}}^{t_i} d' x(t) dt &= \int_{t_{i-1}}^{t_i} d' \left(\frac{t_i - t}{t_i - t_{i-1}} \hat{x}(t_{i-1}) + \frac{t - t_{i-1}}{t_i - t_{i-1}} \hat{x}(t_i) \right) dt \\ &= d'[\hat{x}(t_i) + \hat{x}(t_{i-1})] \frac{(t_i - t_{i-1})}{2}, \end{aligned}$$

it follows that (6) holds. □

The same discretization applies to $(SCCP^*)$, the dual problem. Specifically, corresponding to the partition $\pi = \{t_0, \ldots, t_m\}$, we have

$$\begin{array}{lll} (CP_2(\pi)) & \min & \sum_{i=1}^m \left((\alpha + (T - \frac{t_i + t_{i-1}}{2})a)' \hat{p}(t_{i-1}) + b'[\hat{q}(t_i) + \hat{q}(t_{i-1})] \frac{t_i - t_{i-1}}{2} \right) \\ & \text{s.t.} & G'\hat{p}(t_0) + G'\hat{p}(t_1) + \cdots + G'\hat{p}(t_{i-1}) + H'\hat{q}(t_i) - (\gamma + t_i c) \in \mathcal{K}_3^*,\ i = 1, \ldots, m; \\ & & F'\hat{p}(t_{i-1}) - (t_i - t_{i-1})d \in \mathcal{K}_4^*,\ i = 1, \ldots, m; \\ & & \hat{p}(t_{i-1}) \in \mathcal{K}_1^*,\ \hat{q}(t_i) \in \mathcal{K}_2^*, \quad i = 1, \ldots, m. \end{array} \tag{7}$$

In addition, *assume* (i.e., $\hat{q}(t_0)$ is given, not a variable):

$$H'\hat{q}(t_0) - \gamma \in \mathcal{K}_3^*, \qquad \hat{q}(t_0) \in \mathcal{K}_2^*. \tag{8}$$

Clearly $(CP_2(\pi))$ is also a conic program. Note that the comment in Remark 2.2 applies to $\hat{q}(t_0)$ as well. In particular, if $q(0)$ is not a decision variable in $(SCCP^*)$, we can obtain $\hat{q}(t_0)$ from solving the conic program in (21) below.

Similar to Lemma 2.3, we have the following relationship between $(CP_2(\pi))$ and $(SCCP^*)$:

Lemma 2.4

(i) *Suppose* $(\hat{p}, \hat{q})$ *is a feasible solution of* $(CP_2(\pi))$ *with*

$$\hat{p} = (\hat{p}(t_0), \ldots, \hat{p}(t_{m-1})), \qquad \hat{q} = (\hat{q}(t_1), \ldots, \hat{q}(t_m)).$$

Let $p(t)$ *be the piecewise constant extension of* $(\frac{\hat{p}(t_0)}{t_1 - t_0}, \ldots, \frac{\hat{p}(t_{m-1})}{t_m - t_{m-1}})$, *and* $q(t)$ *the piecewise linear extension of* $(\hat{q}(t_0), \hat{q})$. *Then,* $(p(t), q(t))$ *is a feasible solution to* $(SCCP^*)$, *and the following holds:*

$$\begin{aligned}
& \int_0^T \left[(\alpha + (T-t)a)' p(t) + b' q(t)\right] dt \\
= \; & \sum_{i=1}^m \left((\alpha + (T - \frac{t_i + t_{i-1}}{2})a)' \hat{p}(t_{i-1}) + b'[\hat{q}(t_i) + \hat{q}(t_{i-1})] \frac{t_i - t_{i-1}}{2} \right). \qquad (9)
\end{aligned}$$

(ii) *Suppose that* $(p(t), q(t))$ *is a feasible solution of* $(SCCP^*)$, *with* $p(t)$ *being piecewise constant and* $q(t)$ *piecewise linear. Then,* $\hat{p}(t_{i-1}) := (t_i - t_{i-1}) p(t_{i-1})$ *and* $\hat{q}(t_i) := q(t_i)$, $i = 1, \ldots, m$, *form a feasible solution to* $(CP_2(\pi))$, *and the equality in (9) holds.*

3 Duality Theory

3.1 Strong Feasibility

We say $(u(t), x(t))$ is a *strongly feasible* solution of $(SCCP)$ in (1), if for the closed and convex cones $\mathcal{K}_i$, $i = 1, 2, 3, 4$, with non-empty interiors, the following holds

$$\begin{aligned}
& \alpha + ta - \int_0^t Gu(s)ds - Fx(t) \in \text{int } \mathcal{K}_1, \; t \in (0, T], \\
& \alpha - Fx(0) \in \mathcal{K}_1, \\
& b - Hu(t) \in \text{int } \mathcal{K}_2, \; t \in [0, T], \\
& u(t) \in \text{int } \mathcal{K}_3, \; t \in [0, T], \\
& x(t) \in \text{int } \mathcal{K}_4, \; t \in (0, T], \; x(0) \in \mathcal{K}_4.
\end{aligned}$$

We say the problem $(SCCP)$ in (1) is *strongly feasible* if there exists a strongly feasible solution. The same notions apply to the dual problem $(SCCP^*)$.

In fact, the strong feasibility of $(SCCP)$ and $(SCCP^*)$ can be determined by the strong feasibility of the following two conic programs:

$$
(CP_1) \quad \max \quad c'u + d'x \quad \text{s.t.} \quad \begin{aligned} &\alpha + Ta - Gu - Fx \in \mathcal{K}_1, \\ &Tb - Hu \in \mathcal{K}_2, \\ &u \in \mathcal{K}_3,\ x \in \mathcal{K}_4; \end{aligned} \tag{10}
$$

and

$$
(CP_2) \quad \min \quad a'p + b'q \quad \text{s.t.} \quad \begin{aligned} &G'p + H'q - (\gamma + Tc) \in \mathcal{K}_3^*, \\ &F'p - Td \in \mathcal{K}_4^*, \\ &p \in \mathcal{K}_1^*,\ q \in \mathcal{K}_2^*. \end{aligned} \tag{11}
$$

Note that the constraints of (CP_1) and (CP_2) above are the same as the constraints of $(CP_1(\pi_1))$ and $(CP_2(\pi_1))$, respectively, where π_1 is the trivial partition (or, no partition at all): $\pi_1 = \{t_0 = 0, t_1 = T\}$ (i.e, $m = 1$). The objectives of (CP_1) and (CP_2), however, are different from those of $(CP_1(\pi_1))$ and $(CP_2(\pi_1))$. The choice of these objectives is to facilitate the explicit derivation of a bound on the duality gap; see the proof of Theorem 3.5 below.

Lemma 3.1 *If the two conic programs (CP_1) and (CP_2) are strongly feasible, then $(SCCP)$ and $(SCCP^*)$ are strongly feasible, and so are $(CP_1(\pi))$ and $(CP_2(\pi))$ for any partition π.*

Proof. As noted above the constraints of (CP_1) and (CP_2) are the same as the constraints of $(CP_1(\pi_1))$ and $(CP_2(\pi_1))$. Hence, both $(CP_1(\pi_1))$ and $(CP_2(\pi_1))$ are strongly feasible. Also note that Lemmas 2.3 and 2.4 hold for strongly feasible solutions too. It hence follows that both $(SCCP)$ and $(SCCP^*)$ are strongly feasible with $u(t), p(t)$ constant, $x(t), q(t)$ linear on $[0, T]$. This, in turn, implies that $(CP_1(\pi))$ and $(CP_2(\pi))$, for *any* partition π, are strongly feasible, based again on the strongly feasible version of Lemmas 2.3 and 2.4. $\square$

Below we shall focus of an *even* partition, denoted π_ϵ, which divides the interval $[0, T]$ into m equal segments, each of length 2ϵ; i.e., $\epsilon = \frac{T}{2m}$. For $i = 1, 2, 3, 4$, denote $\mathcal{K}_{i,m} := \underbrace{\mathcal{K}_i \times \cdots \times \mathcal{K}_i}_{m}$; and similarly denote $\mathcal{K}_{i,m}^*$. We can then express $(CP_1(\pi_\epsilon))$ and $(CP_2(\pi_\epsilon))$ as follows:

$$
(CP_1(\pi_\epsilon)) \quad \max \quad \hat{h}_1'\hat{u} + \hat{d}_1'\hat{x} + \epsilon d'\hat{x}(t_0) \quad \text{s.t.} \quad \begin{aligned} &\hat{g}_1 - \hat{G}'\hat{u} - \hat{F}'\hat{x} \in \mathcal{K}_{1,m}, \\ &\hat{f}_1 - \hat{H}'\hat{u} \in \mathcal{K}_{2,m}, \\ &\hat{u} \in \mathcal{K}_{3,m},\ \hat{x} \in \mathcal{K}_{4,m}; \end{aligned} \tag{12}
$$

and

$$
(CP_2(\pi_\epsilon)) \quad \begin{array}{ll} \min & \hat{g}_2'\hat{p} + \hat{f}_2'\hat{q} + \epsilon b'\hat{q}(t_0) \\ \text{s.t.} & \hat{G}\hat{p} + \hat{H}\hat{q} - \hat{h}_2 \in \mathcal{K}_{3,m}^*, \\ & \hat{F}\hat{p} - \hat{d}_2 \in \mathcal{K}_{4,m}^*, \\ & \hat{p} \in \mathcal{K}_{1,m}^*,\ \hat{q} \in \mathcal{K}_{2,m}^*; \end{array} \tag{13}
$$

where

$$
\hat{F} = \begin{pmatrix} F' & & & \\ & F' & & \\ & & \ddots & \\ & & & F' \end{pmatrix}, \quad \hat{G} = \begin{pmatrix} G' & & & \\ G' & G' & & \\ \cdots & & & \\ G' & G' & \cdots & G' \end{pmatrix}, \quad \hat{H} = \begin{pmatrix} H' & & & \\ & H' & & \\ & & \ddots & \\ & & & H' \end{pmatrix};
$$

$$
\hat{f}_1 = \begin{pmatrix} 2\epsilon b \\ 2\epsilon b \\ \vdots \\ 2\epsilon b \end{pmatrix}, \hat{f}_2 = \begin{pmatrix} 2\epsilon b \\ \vdots \\ 2\epsilon b \\ \epsilon b \end{pmatrix}, \hat{g}_1 = \begin{pmatrix} \alpha + Ta \\ \alpha + (T-2\epsilon)a \\ \vdots \\ \alpha + 2\epsilon a \end{pmatrix}, \hat{g}_2 = \begin{pmatrix} \alpha + (T-\epsilon)a \\ \alpha + (T-3\epsilon)a \\ \vdots \\ \alpha + \epsilon a \end{pmatrix};
$$

and

$$
\hat{h}_1 = \begin{pmatrix} \gamma + \epsilon c \\ \gamma + 3\epsilon c \\ \vdots \\ \gamma + (T-\epsilon)c \end{pmatrix}, \hat{h}_2 = \begin{pmatrix} \gamma + 2\epsilon c \\ \gamma + 4\epsilon c \\ \vdots \\ \gamma + Tc \end{pmatrix}, \hat{d}_1 = \begin{pmatrix} \epsilon d \\ 2\epsilon d \\ \vdots \\ 2\epsilon d \end{pmatrix}, \hat{d}_2 = \begin{pmatrix} 2\epsilon d \\ 2\epsilon d \\ \vdots \\ 2\epsilon d \end{pmatrix}.
$$

In the analysis to follow, we need to compare the dual of $(CP_1(\pi_\epsilon))$, $CP_1^*(\pi_\epsilon)$ below, with $(CP_2(\pi_\epsilon))$. For this technical purpose, we reversed the order of $\hat{u}, \hat{x}$ in the formulation of $(CP_1(\pi_\epsilon))$; that is, in $(CP_1(\pi_\epsilon))$,

$$
\hat{u} = (\hat{u}(t_{m-1}), \hat{u}(t_{m-2}), \ldots, \hat{u}(t_0)); \hat{x} = (\hat{x}(t_m), \hat{x}(t_{m-1}), \ldots, \hat{x}(t_1)).
$$

Note that the constant terms $\epsilon d'\hat{x}(0)$ and $\epsilon b'\hat{q}(0)$ can be dropped from the objectives of $(CP_1(\pi_\epsilon))$ and $(CP_2(\pi_\epsilon))$, respectively. We can then write out the duals of $(CP_1(\pi_\epsilon))$ and $(CP_2(\pi_\epsilon))$ as follows:

$$
(CP_1^*(\pi_\epsilon)) \quad \begin{array}{ll} \min & \hat{g}_1'\hat{p} + \hat{f}_1'\hat{q} \\ \text{s.t.} & \hat{G}\hat{p} + \hat{H}\hat{q} - \hat{h}_1 \in \mathcal{K}_{3,m}^*, \\ & \hat{F}\hat{p} - \hat{d}_1 \in \mathcal{K}_{4,m}^*, \\ & \hat{p} \in \mathcal{K}_{1,m}^*,\ \hat{q} \in \mathcal{K}_{2,m}^* \end{array}
$$

and

$$
(CP_2^*(\pi_\epsilon)) \quad \begin{array}{ll} \max & \hat{h}_2'\hat{u} + \hat{d}_2'\hat{x} \\ \text{s.t.} & \hat{g}_2 - \hat{G}'\hat{u} - \hat{F}'\hat{x} \in \mathcal{K}_{1,m}, \\ & \hat{f}_2 - \hat{H}'\hat{u} \in \mathcal{K}_{2,m}, \\ & \hat{u} \in \mathcal{K}_{3,m},\ \hat{x} \in \mathcal{K}_{4,m}. \end{array}
$$

Lemma 3.2 *Suppose (CP_1) and (CP_2) are strongly feasible. Then, $(CP_1(\pi_\epsilon))$, $(CP_2(\pi_\epsilon))$ and their dual problems are all strongly feasible.*

Proof. The strong feasibility of $(CP_1(\pi_\epsilon))$ and $(CP_2(\pi_\epsilon))$ has already been established in Lemma 3.1 (which holds for any partition π). Here we show that $(CP_1^*(\pi_\epsilon))$ is strongly feasible; the case for $(CP_2^*(\pi_\epsilon))$ is completely analogous. Consider the partition $\pi = \{t_0, t_1, t_2, \dots, t_m, t_{m+1}\}$, with $t_0 = 0, t_{m+1} = T$, and $t_1 - t_0 = t_{m+1} - t_m = \epsilon$, while the length of all other segments is 2ϵ. Then, the constraints of $(CP_2(\pi))$ are:

$$\begin{aligned}
&\begin{pmatrix} & \hat{G} & \\ G' & \dots\ G' & G' \end{pmatrix}\hat{p} + \begin{pmatrix} \hat{H} & \\ & H' \end{pmatrix}\hat{q} - \begin{pmatrix} \hat{h}_1 \\ \gamma + Tc \end{pmatrix} \in \mathcal{K}^*_{3,m+1},\\
&\begin{pmatrix} \hat{F} & \\ & F' \end{pmatrix}\hat{p} - \begin{pmatrix} \hat{d}_1 \\ \epsilon d \end{pmatrix} \in \mathcal{K}^*_{4,m+1},\\
&\hat{p} \in \mathcal{K}^*_{1,m+1},\ \hat{q} \in \mathcal{K}^*_{2,m+1}.
\end{aligned}$$

Comparing the above with the constraints of $(CP_1^*(\pi_\epsilon))$, we observe that if

$$(\hat{p}(t_0), \dots, \hat{p}(t_m)), \qquad (\hat{q}(t_1), \dots, \hat{q}(t_{m+1}))$$

is a strongly feasible solution of $(CP_2(\pi))$, then

$$((\hat{p}(t_0), \dots, \hat{p}(t_{m-1})), \qquad (\hat{q}(t_1), \dots, \hat{q}(t_m))$$

is a strongly feasible solution to $(CP_1^*(\pi_\epsilon))$. Since $(CP_2(\pi))$ is strongly feasible following Lemma 3.1, we know $(CP_1^*(\pi_\epsilon))$ is also strongly feasible. □

Proposition 3.3 *Suppose (CP_1) and (CP_2) are strongly feasible. Then, we have*

$$v(CP_1(\pi_\epsilon)) \le v(SCCP) \le v(SCCP^*) \le v(CP_2(\pi_\epsilon)). \tag{14}$$

Proof. From Lemma 3.2, we know that $(CP_1(\pi_\epsilon))$ and its dual are both strongly feasible; hence, $(CP_1(\pi_\epsilon))$ has an optimal solution, with a finite $v(CP_1(\pi_\epsilon))$. By the same reasoning, $(CP_2(\pi_\epsilon))$ also has an optimal solution, with a finite $v(CP_2(\pi_\epsilon))$.

From Lemma 2.3, we know the optimal solution of $(CP_1(\pi_\epsilon))$ can be extended to a feasible solution of $(SCCP)$; hence, the first inequality (since $(SCCP)$ is a maximization problem). A similar argument justifies the third inequality. The second inequality follows from the weak duality in Proposition 2.1. □

In the special case of SCLP (refer to (2)), the strong feasibility condition in the above proposition reduces to plain feasibility.

Corollary 3.4 *In the special case of SCLP, the inequalities in (14) hold, if the linear programming counterparts of (CP_1) and (CP_2) are feasible.*

3.2 Strong Duality

Theorem 3.5 *Suppose (CP_1) and (CP_2) are strongly feasible, with finite optimal values. Then, there exists a constant $\Gamma > 0$, such that*

$$v(CP_2(\pi_\epsilon)) - v(CP_1(\pi_\epsilon)) \leq \Gamma\epsilon. \tag{15}$$

Consequently, we must have $v(SCCP) = v(SCCP^*)$, i.e., strong duality holds.

Proof. First note that strong duality follows immediately from the inequality in (15) by letting $m \to \infty$ and hence $\epsilon = \frac{T}{2m} \to 0$, taking into account the inequalities in Proposition 3.3.

To establish the error bound in (15), consider the following primal-dual pair of conic programs:

$$\begin{array}{ll} \max & \hat{h}_2'\hat{u} + \hat{d}_2'\hat{x} \\ \text{s.t.} & \hat{g}_1 - \hat{G}'\hat{u} - \hat{F}'\hat{x} \in \mathcal{K}_{1,m}, \\ & \hat{f}_1 - \hat{H}'\hat{u} \in \mathcal{K}_{2,m}, \\ & \hat{u} \in \mathcal{K}_{3,m}, \ \hat{x} \in \mathcal{K}_{4,m}, \end{array} \tag{16}$$

and

$$\begin{array}{ll} \min & \hat{g}_1'\hat{p} + \hat{f}_1'\hat{q} \\ \text{s.t.} & \hat{G}\hat{p} + \hat{H}\hat{q} - \hat{h}_2 \in \mathcal{K}_{3,m}^*, \\ & \hat{F}\hat{p} - \hat{d}_2 \in \mathcal{K}_{4,m}^*, \\ & \hat{p} \in \mathcal{K}_{1,m}^*, \ \hat{q} \in \mathcal{K}_{2,m}^*. \end{array} \tag{17}$$

Note that the primal problem in (16) has the same constraints as $(CP_1(\pi_\epsilon))$ but the objective function of $(CP_2^*(\pi_\epsilon))$; whereas the dual problem in (17) has the constraints of $(CP_2(\pi_\epsilon))$ but the objective function of $(CP_1^*(\pi_\epsilon))$. Hence, both primal and dual are strongly feasible, since $(CP_1(\pi_\epsilon))$ and $(CP_2(\pi_\epsilon))$ are. Consequently they both have optimal solutions, which we denote as $(\hat{u}^*, \hat{x}^*)$ and $(\hat{p}^*, \hat{q}^*)$. Note that these are feasible solutions to $(CP_1(\pi_\epsilon))$ and $(CP_2(\pi_\epsilon))$, respectively. Hence, from (12) and (13), we have

$$v(CP_1(\pi_\epsilon)) \geq \hat{h}_1'\hat{u}^* + \hat{d}_1'\hat{x}^* + \epsilon d'\hat{x}(t_0), \quad \text{and} \quad v(CP_2(\pi_\epsilon)) \leq \hat{g}_2'\hat{p}^* + \hat{f}_2'\hat{q}^* + \epsilon b'\hat{q}(t_0).$$

Hence,

$$\begin{aligned} & v(CP_2(\pi_\epsilon)) - v(CP_1(\pi_\epsilon)) \\ \leq\ & \hat{g}_2'\hat{p}^* + \hat{f}_2'\hat{q}^* - \hat{h}_1'\hat{u}^* - \hat{d}_1'\hat{x}^* + \epsilon[b'\hat{q}(t_0) - d'\hat{x}(t_0)] \\ =\ & \left[\hat{g}_1 - \begin{pmatrix} a\epsilon \\ a\epsilon \\ \vdots \\ a\epsilon \end{pmatrix}\right]'\hat{p}^* + \left[\hat{f}_1 - \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \epsilon b \end{pmatrix}\right]'\hat{q}^* - \left[\hat{h}_2 - \begin{pmatrix} \epsilon c \\ \epsilon c \\ \vdots \\ \epsilon c \end{pmatrix}\right]'\hat{u}^* - \left[\hat{d}_2 - \begin{pmatrix} \epsilon d \\ 0 \\ \vdots \\ 0 \end{pmatrix}\right]'\hat{x}^* \end{aligned}$$

$$
\begin{aligned}
& +\epsilon[b'\hat{q}(t_0) - d'\hat{x}(t_0)] \\
= \quad & \hat{g}_1'\hat{p}^* - \epsilon a'(\hat{p}_1^* + \cdots + \hat{p}_m^*) + \hat{f}_1'\hat{q}^* - \epsilon b'\hat{q}_m^* - \hat{h}_2'\hat{u}^* + \epsilon c'(\hat{u}_1^* + \cdots + \hat{u}_m^*) - \hat{d}_2'\hat{x}^* + \epsilon d'\hat{x}_1^* \\
& +\epsilon[b'\hat{q}(t_0) - d'\hat{x}(t_0)] \\
= \quad & -\epsilon\Big(a'\sum_{i=1}^m \hat{p}_i^* + b'\hat{q}_m^*\Big) + \epsilon\Big(c'\sum_{i=1}^m \hat{u}_i^* + d'\hat{x}_1^*\Big) + \epsilon[b'\hat{q}(t_0) - d'\hat{x}(t_0)].
\end{aligned}
$$

From the primal feasibility of $(\hat{u}^*, \hat{x}^*)$, we have

$$
\begin{aligned}
& \alpha + Ta - G\sum_{i=1}^m \hat{u}_i^* - F\hat{x}_1^* \in \mathcal{K}_1, \\
& Tb - H\sum_{i=1}^m \hat{u}_i^* \in \mathcal{K}_2, \\
& \sum_{i=1}^m \hat{u}_i^* \in \mathcal{K}_3, \;\; \hat{x}_1^* \in \mathcal{K}_4.
\end{aligned}
$$

So we know that $\begin{pmatrix} \sum_{i=1}^m \hat{u}_i^* \\ \hat{x}_1^* \end{pmatrix}$ is a feasible solution to (CP_1). Thus,

$$
c'\sum_{i=1}^m \hat{u}_i^* + d'\hat{x}_1^* \le v(CP_1).
$$

Similarly, from the dual feasibility of $(\hat{p}^*, \hat{q}^*)$, we have

$$
\begin{aligned}
& G'\sum_{i=1}^m \hat{p}_i^* + H'\hat{q}_m^* - (\gamma + Tc) \in \mathcal{K}_3^*, \\
& F'\sum_{i=1}^m \hat{p}_i^* - Td \in \mathcal{K}_4^*, \\
& \sum_{i=1}^m \hat{p}_i^* \in \mathcal{K}_1^*, \;\; \hat{q}_m^* \in \mathcal{K}_2^*.
\end{aligned}
$$

Hence, $\begin{pmatrix} \sum_{i=1}^m \hat{p}_i^* \\ \hat{q}_m^* \end{pmatrix}$ is a feasible solution to (CP_2), and

$$
a'\sum_{i=1}^m \hat{p}_i^* + b'\hat{q}_m^* \ge v(CP_2).
$$

Putting the above together, we have

$$
\begin{aligned}
& v(CP_2(\pi_\epsilon)) - v(CP_1(\pi_\epsilon)) \\
\le \quad & \epsilon\left(-a'\sum_{i=1}^m \hat{p}_i^* - b'\hat{q}_m^* + c'\sum_{i=1}^m \hat{u}_i^* + d'\hat{x}_1^*\right) + \epsilon[b'\hat{q}(t_0) - d'\hat{x}(t_0)] \\
\le \quad & \epsilon[v(CP_1) - v(CP_2) + b'\hat{q}(t_0) - d'\hat{x}(t_0)].
\end{aligned}
$$

Hence, we can let

$$\Gamma := v(CP_1) - v(CP_2) + b'\hat{q}(t_0) - d'\hat{x}(t_0);$$

and $\Gamma < \infty$, since $v(CP_1) < \infty$ and $v(CP_2) < \infty$ as assumed. □

Combining the above theorem with Corollary 3.4, we have

Corollary 3.6 *In the special case of SCLP, Theorem 3.5 holds — in particular, strong duality holds for SCLP, if the two corresponding linear programs (counterparts of* (CP_1) *and* (CP_2)*) are feasible, with finite optimal values.*

Note, the strong duality of SCLP has only been recently established in [30], based on a simplex-like algorithm that solves the SCLP.

4 Approximation Algorithm

Theorem 3.5 suggests that we can solve $(SCCP)$ (and/or its dual) through solving its discretized version, the ordinary conic program $(CP_1(\pi_\epsilon))$ (and/or $(CP_2(\pi_\epsilon))$). The latter is readily solvable by standard algorithms, e.g., SeDuMi [28]; and the (discrete) solution can then be extended into the piecewise-constant control and piecewise-linear state variables as a feasible solution to the SCCP problem. Furthermore, the explicit error bound in (15) means that we can achieve any required accuracy by a partition (π_ϵ) of $[0, T]$ into a sufficiently large number (m) of equal segments to construct the discretized conic program. Specifically, if δ is the required accuracy, then we can choose

$$m = \left\lceil \frac{T\Gamma}{2\delta} \right\rceil, \tag{18}$$

where (refer to the end of the proof of Theorem 3.5)

$$\Gamma = v(CP_1) - v(CP_2) + b'\hat{q}(t_0) - d'\hat{x}(t_0). \tag{19}$$

Then, from (14) and (15), we have

$$\delta \geq \frac{T\Gamma}{2m} = \Gamma\epsilon \geq v(CP_2(\pi_\epsilon)) - v(CP_1(\pi_\epsilon)) \geq v(SCCP^*) - v(SCCP).$$

That is, the duality gap is guaranteed to be no greater than δ.

To select m following (18), we need to first derive Γ. This involves solving the two conic programs (CP_1) and (CP_2). In addition, we also need to determine $\hat{x}(t_0)$ and $\hat{q}(t_0)$. This can be accomplished by solving the following two conic programs:

$$\begin{array}{ll} \max & d'x(t_0) \\ \text{s.t.} & \alpha - Fx(t_0) \in \mathcal{K}_1, \\ & x(t_0) \in \mathcal{K}_4; \end{array} \tag{20}$$

and

$$\begin{array}{ll} \min & b'q(t_0) \\ \text{s.t.} & H'q(t_0) - \gamma \in \mathcal{K}_3^*, \\ & q(t_0) \in \mathcal{K}_2^*. \end{array} \tag{21}$$

Note that the constraints of the above two problems originate from the constraints of $(SCCP)$ and $(SCCP^*)$ in (1) and (3). Clearly, maximizing $d'x(t_0)$ and minimizing $b'q(t_0)$ improves the best we can the estimation of the error bound. In general, we can choose $\hat{x}(t_0)$ and $\hat{q}(t_0)$ as any feasible solution to the two conic programs in (20) and (21).

Remark 4.1 Observe that the choice of $\hat{x}(t_0)$ has no bearing on $v(SCCP)$ as it will be weighted by $t_1 - t_0 = \epsilon$ in (4). (The same applies to $\hat{q}(t_0)$.) This fact, along with the strong duality result in Theorem 3.5, means that the initial value $x(0)$ has no impact on the optimal objective value $v(SCCP)$. Hence, whether $x(0)$ is given or is a decision variable is immaterial. This appears to be a special feature of the SCCP, which makes it distinct from other control problems.

In summary, our algorithm amounts to solving two conic programming problems: $(CP_1(\pi_\epsilon))$ and $(CP_2(\pi_\epsilon))$, and in determining ϵ (or, equivalently, m), we also need to solve four more conic programs: the two in (20) and (21), and (CP_1) and (CP_2). Conic programs are known to be polynomially solvable. Hence, ours is a polynomial-time algorithm.

The above choice of m via (18) typically results in a rather conservative overestimate, since (15) is an *upper* bound on the duality gap. Specifically,

$$\begin{aligned} \Gamma\epsilon &\geq v(CP_2(\pi_\epsilon)) - v(CP_1(\pi_\epsilon)) \\ &= [v(CP_2(\pi_\epsilon)) - v(SCCP^*)] + [v(SCCP^*) - v(SCCP)] + [v(SCCP) - v(CP_1(\pi_\epsilon))], \end{aligned}$$

with all three differences on the right hand side being non-negative (refer to (14)), whereas what is really needed is to bound either the first or the third difference.

We can avoid specifying the value of m at the beginning by simply solving a sequence of conic programs, $(CP_1(\pi_\epsilon))$ and $(CP_2(\pi_\epsilon))$, starting from some initial choice of m (or, equivalently, setting $\epsilon = 1/m$), and every time double the value of m (or, equivalently, half the value of ϵ), until the duality gap falls below the required δ. The reason to double m in each iteration is to ensure that $v(CP_1(\pi_\epsilon))$ is non-decreasing and $v(CP_2(\pi_\epsilon))$ non-increasing (since the optimal solution in the current iteration can be considered as a feasible solution in the next iteration by copying the previous solution values to the new finer sub-intervals, hence feasible for the new problem); therefore, the error bound, $v(CP_2(\pi_\epsilon)) - v(CP_1(\pi_\epsilon))$, is non-increasing.

Another point of interest is that we do not have to be concerned with whether $(SCCP)$ and $(SCCP^*)$ are solvable or not. (Note that even if $v(SCCP)$ and $v(SCCP^*)$ are finite, they may still not be attainable by any feasible solutions.) As long as we can solve $(CP_1(\pi_\epsilon))$ and $(CP_2(\pi_\epsilon))$, we get feasible solutions to $(SCCP)$ and $(SCCP^*)$ with a known duality gap.

5 Applications

Here we illustrate the power and versatility of the SCCP through two concrete applications, the first one being developed in the next subsection, and the second in the two following ones. In both cases the problems would be intractable without the solution approach developed here.

5.1 Robust SCLP

Consider the SCLP problem in (2), reproduced here as follows:

$$\begin{array}{ll} \max & \int_0^T [(\gamma + (T-t)c)' u(t) + d' x(t)] dt \\ \text{s.t.} & \alpha + ta - \int_0^t G u(s) ds - F x(t) \in \Re^n_+, \\ & b - H u(t) \in \Re^\ell_+, \\ & u(t) \in \Re^m_+, \ x(t) \in \Re^k_+, \ t \in [0, T], \end{array} \tag{22}$$

where G, F, H are matrices of dimensions $n \times m, n \times k, \ell \times m$, and α, a, and b are vectors.

Now, consider the robust counterpart of the above SCLP. Suppose there is data uncertainty regarding the matrix F. Specifically, suppose

$$F \in \{F^0 + \sum_{j=1}^h y_j F^j \mid y' y \leq 1\}, \tag{23}$$

where F^j, $j = 0, 1, \ldots, h$ are given matrices of dimension $n \times k$. Then, the first constraint in the SCLP becomes

$$\alpha + ta - \int_0^t G u(s) ds - (F^0 + \sum_{j=1}^h y_j F^j) x(t) \in \Re^n_+, \ \forall y : \ y' y \leq 1,$$

or, equivalently,

$$\min_{y'y \leq 1} \left\{ \alpha_i + t a_i - \int_0^t G_i u(s) ds - (F_i^0 + \sum_{j=1}^h y_j F_i^j) x(t) \right\} \geq 0, \quad i = 1, 2, \ldots, n, \tag{24}$$

where F_i^j is the ith row in F^j, $j = 0, 1, \ldots, h$; and G_i is the ith row of G. The above can be written as

$$\alpha_i + ta_i - \int_0^t G_i u(s) ds - (F_i^0 x(t) + \|F_i x(t)\|) \geq 0,$$

where

$$F_i = \begin{pmatrix} F_i^1 \\ \vdots \\ F_i^h \end{pmatrix}.$$

That is,

$$\begin{pmatrix} \alpha_i + ta_i - \int_0^t G_i u(s) ds - F_i^0 x(t) \\ F_i x(t) \end{pmatrix} \in \mathrm{SOC}(1+h),$$

where $\mathrm{SOC}(n)$ denotes the n-dimensional second-order cone. (Recall from §1.2 (point 4), for $s \in \Re$ and $v \in \Re^n$, $\begin{pmatrix} s \\ v \end{pmatrix} \in \mathrm{SOC}(n+1)$ means $s \geq ||v||$.)

In other words, we may write (24) as

$$\begin{pmatrix} \alpha_i \\ 0 \end{pmatrix} + t \begin{pmatrix} a_i \\ 0 \end{pmatrix} - \int_0^t \begin{pmatrix} G_i \\ 0 \end{pmatrix} u(s) ds - \begin{pmatrix} F_i^0 \\ -F_i \end{pmatrix} x(t) \in \mathrm{SOC}(1+h), \; i = 1, 2, \ldots, n. \tag{25}$$

So the robust counterpart for the SCLP is precisely an SCCP, with a second-order cone constraint.

Similarly, it is readily verified that if either G or H, or a combination of F, G and H, involves uncertainty, then the robust counterpart of the SCLP is again an SCCP, as long as the uncertainty sets are ellipsoidal as in (23) and mutual independent. (Note that when a or b is subject to ellipsoidal uncertainty, the robust counterpart of the SCLP remains an SCLP.) In much the same spirit (see, e.g. [9]), one can show that the robust version of SCSOCP (*Separated Continuous Second Order Cone Programming*), once the uncertainty is captured by ellipsoids and the uncertainty in the 'norm' part and in the 'scalar' part with regard to the SOC cones can be separated, is equivalent to a SCSDP (*Separated Continuous Semidefinite Programming*) problem, where the cones in question are nothing but the cones of positive semidefinite matrices.

5.2 Sign-Constrained Linear Quadratic Control

Consider the following control problem:

$$\begin{array}{ll} \min & \int_0^T (x(t)'Qx(t) + c'x(t) + u(t)'Ru(t) + d'u(t))dt \\ \text{s.t.} & \dot{x}(t) = Bu(t) + b, \\ & x(0) = \alpha, \alpha \geq 0, \\ & a - Hu(t) \in \Re_+^\ell, \\ & u(t) \in \Re_+^m, \; x(t) \in \Re_+^n, \end{array} \tag{26}$$

where B, H, Q and R are matrices of dimensions $n \times m$, $\ell \times m$, $n \times n$, and $m \times m$, respectively; with Q and R being positive semidefinite; and b, α, a, c and d are vectors.

This is a linear quadratic (LQ) control problem, with two distinct features. First, the control and the state variables are all subject to, in a point-wise manner, some polyhedral constraints. Traditional approaches (see e.g. [2]) designed for *unconstrained* LQ control problems are useless here. In general, sign-constrained LQ problems are known to be extremely difficult to solve. Regarding the sign-constrained LQ problems we refer to Chen and Zhou [12], and Hu and Zhou [17]. In these papers the authors studied the LQ control problem where the control variable $u(t)$ is constrained to a given conic region while the state variable $x(t)$ is unconstrained. As a matter of fact, the authors of these papers assumed the state variable is one-dimensional; in our notation, this amounts to $n = 1$. Compared to their models, our sign-constrained LQ model is much more general, in that the state variable $x(t)$ in our case can be constrained to a conic region as well. But of course our solution procedure is a numerical one. The second feature of the problem in (26) is that the state equation $\dot{x}(t) = Bu(t) + b$ does not involve $x(t)$ on the right hand side. This enables us to reformulate the problem as an SCCP, with *second-order cone* constraints, which can be efficiently solved by our approach developed in the last two sections.

First, we need to replace the quadratic objective with a linear objective. To this end, we introduce four additional variables, $x_0(t)$, $y_0(t)$, $u_0(t)$ and $z_0(t)$, such that

$$\begin{aligned} x_0^2(t) &\geq x(t)'Qx(t), \qquad & y_0(t) &\geq x_0^2(t); \\ u_0^2(t) &\geq u(t)'Ru(t), \qquad & z_0(t) &\geq u_0^2(t). \end{aligned}$$

We can rewrite the above constraints as follows, with SOC(k) denoting the second-order cone (or Lorentz cone) in $\Re^k$:

$$\begin{pmatrix} x_0(t) \\ Q^{\frac{1}{2}} x(t) \end{pmatrix} \in \mathrm{SOC}(n+1), \qquad \begin{pmatrix} 1 + y_0(t) \\ 1 - y_0(t) \\ 2x_0(t) \end{pmatrix} \in \mathrm{SOC}(3), \tag{27}$$

$$\begin{pmatrix} u_0(t) \\ R^{\frac{1}{2}} u(t) \end{pmatrix} \in \mathrm{SOC}(m+1), \qquad \begin{pmatrix} 1 + z_0(t) \\ 1 - z_0(t) \\ 2u_0(t) \end{pmatrix} \in \mathrm{SOC}(3). \tag{28}$$

This way, the LQ problem in (26) can be equivalently reformulated as the following SCCP problem:

$$\begin{aligned} \min \quad & \int_0^T [y_0(t) + c'x(t) + z_0(t) + d'u(t)]dt \\ \text{s.t.} \quad & \alpha + bt + \int_0^t Bu(s)ds - x(t) = 0, \end{aligned}$$

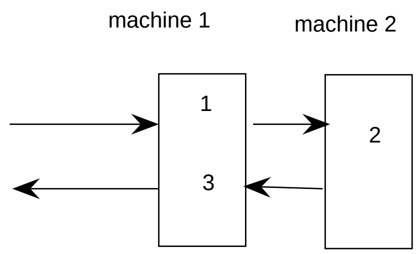

Figure 1: The fluid network model.

$$
\begin{aligned}
& a - Hu(t) \in \Re^{\ell}_{+}, \\
& u(t) \in \Re^{m}_{+},\ x(t) \in \Re^{n}_{+}, \\
& \text{SOC constraints in (27,28).}
\end{aligned}
$$

5.3 A Fluid Network

Here we provide a numerical example of the LQ control problem in the last section. We consider the same example as in [30], but change its linear objective to a quadratic objective.

A network processes a continuous flow of jobs at two machines. The jobs visit machines 1 and 2 in the order $1 \to 2 \to 1$, i.e., a total of three processing steps; see Figure 1. Corresponding to each processing step, there is a buffer holding the fluid. At $t = 0$, the initial levels of fluid at the three steps are 50, 20 and 120 units. The input rates of fluid from outside to the three buffers are 0.01, 0.01, 0.01. To process each unit of job ("fluid"), the time requirements at the three steps are 0.4, 0.8, 0.2 time units.

The problem is to find the processing rates at the three steps, $u_i(t),\ i = 1, 2, 3$, which determine the fluid levels in the three buffers, $x_i(t),\ i = 1, 2, 3$, during a given time interval $[0, T]$ such that the fluid levels in the three buffers are maintained as close as possible to a prespecified constant level $d = (30\ 10\ 80)'$.

The problem can be formulated as follows:

$$
\begin{aligned}
\min \quad & \int_0^T [(x(t) - d)'(x(t) - d)]dt \\
\text{s.t.} \quad & \int_0^t Gu(s)ds + x(t) = \alpha + ta,
\end{aligned}
$$

	Number of Segments (m)							
	1		4		8		16	
T	value	e.b.	value	e.b.	value	e.b.	value	e.b.
3	17354.55	168.08	17459.59	10.50	17464.85	2.63	17466.16	0.66
7	42632.05	2091.45	43933.57	123.03	43994.47	31.71	44010.30	7.76
9	55388.58	3907.64	57763.82	210.32	57867.90	55.86	57895.76	13.30

Table 1: Objective values (without the $T - Td'd$ term) and error bounds (e.b.) for the SCCP in (29).

$$\begin{aligned} & b - Hu(t) \geq 0, \\ & u(t) \geq 0, \; x(t) \geq 0, \; t \in [0, T], \end{aligned}$$

where

$$G = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}, \qquad H = \begin{pmatrix} 0.4 & 0 & 0.2 \\ 0 & 0.8 & 0 \end{pmatrix},$$

$$\alpha = \begin{pmatrix} 50 \\ 20 \\ 120 \end{pmatrix}, \quad a = \begin{pmatrix} 0.01 \\ 0.01 \\ 0.01 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Following the manipulation in §5.2, we can turn the above problem into the following equivalent form:

$$\begin{aligned} \min \quad & \int_0^T [y_0(t) - 2d'x(t)]dt + Td'd \\ \text{s.t.} \quad & \begin{pmatrix} \alpha \\ 0 \end{pmatrix} + t \begin{pmatrix} a \\ 0 \end{pmatrix} - \int_0^t \begin{pmatrix} G \\ 0 \end{pmatrix} u(s)ds - \begin{pmatrix} 0 & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & -I \end{pmatrix} \tilde{x}(t) \in \{0\}^3 \times \Re^3_+, \\ & b - Hu(t) \in \Re^2_+, \\ & u(t) \in \Re^3_+, \\ & \tilde{x}(t) \in \mathrm{SOC}(3) \times \mathrm{SOC}(4), \; t \in [0, T]. \end{aligned}$$

Here,

$$\tilde{x}(t) := \begin{pmatrix} 1 + y_0(t) \\ 1 - y_0(t) \\ 2x_0(t) \\ x_0(t) \\ x(t) \end{pmatrix};$$

and the last constraint in the above problem simply represents $y_0(t) - x_0^2(t) \geq 0$ and $x_0^2(t) - x'(t)x(t) \geq 0$.

Denote

$$\tilde{d} := \begin{pmatrix} -1 \\ 0 \\ 0 \\ 0 \\ 2d \end{pmatrix}, \qquad \tilde{\alpha} := \begin{pmatrix} \alpha \\ 2 \\ 0 \\ 0 \end{pmatrix}, \qquad \tilde{a} := \begin{pmatrix} a \\ 0 \\ 0 \\ 0 \end{pmatrix};$$

and

$$\tilde{F} := \begin{pmatrix} 0 & 0 & 0 & 0 & I \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -2 & 0 \\ 0 & 0 & 0 & 0 & -I \end{pmatrix} \qquad \tilde{G} := \begin{pmatrix} G \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

We can further express the above problem in the canonical form of SCCP, with u and $\tilde{x}$ as variables, as follows:

$$\begin{aligned} \max \quad & \int_0^T \tilde{d}'\tilde{x}(t)dt + T - Td'd \\ \text{s.t.} \quad & \tilde{\alpha} + t\tilde{a} - \int_0^t \tilde{G}u(s)ds - \tilde{F}\tilde{x}(t) \in \{0\}^5 \times \Re^3_+, \\ & b - Hu(t) \in \Re^2_+, \\ & u(t) \in \Re^3_+, \\ & \tilde{x}(t) \in \text{SOC}(3) \times \text{SOC}(4);\ t \in [0, T]. \end{aligned} \tag{29}$$

(The constant term, $T - Td'd$, can be dropped from the objective function.) The dual of the above is:

$$\begin{aligned} \min \quad & \int_0^T [(\tilde{\alpha} + (T-t)\tilde{a})'p(t) + b'q(t)]dt \\ \text{s.t.} \quad & \int_0^t \tilde{G}'p(s)ds + H'q(t) \in \Re^3_+, \\ & \tilde{F}'p(t) - \tilde{d} \in \text{SOC}(3) \times \text{SOC}(4), \\ & p(t) \in \Re^5 \times \Re^3_+,\ q(t) \in \Re^2_+,\ t \in [0, T]. \end{aligned}$$

We first derive $\hat{\tilde{x}}(0)$ and $\hat{q}(0)$ via solving the two conic programs in (20) and (21). We then solve the discretized versions of the SCCP in (29) and its dual, $(CP_1(\pi_\epsilon))$ and $(CP_2(\pi_\epsilon))$. The results are summarized in Table 1, for $T = 3, 7$ and 9, where the entries under "value" are the objective values of $(CP_1(\pi_\epsilon))$, without the $T - Td'd$ term; and the error bounds report the difference $v(CP_2(\pi_\epsilon)) - v(CP_1(\pi_\epsilon))$. We observe that the accuracy of the approximation algorithm is very good even with a small number (m) of segments. For instance, in the case of $T = 9$, with $m = 4$, the error bound indicates the objective value obtained (57,763.82) is already well below 0.4% in terms of relative error from the optimal value.

6 Concluding Remarks

We have developed in this paper a duality theory for the SCCP. Specifically, we have shown that the strong duality between the SCCP and its dual is implied by two related ordinary conic programs being strongly feasible, with finite optimal solutions. As a by-product, we have also developed a polynomial-time approximation algorithm that solves the SCCP to any desired accuracy, and with an easily computable error bound.

All the results in this paper can be readily generalized to allow the input data — the vectors a, b, c, d in $(SCCP)$ — to be *piecewise* constants over $[0, T]$. Refer to Wang [29] for this and other extensions.

From Theorem 3.5, we know as the granularity of the partition π_ϵ increases (i.e., when $\epsilon \to 0$), the duality gap can be closed, and the optimal objective value of the discretized conic program $v(CP_1(\pi_\epsilon))$ approaches the optimal objective value $v(SCCP)$ of the original SCCP. It remains an open problem, however, whether the optimal *solution* to $(CP_1(\pi_\epsilon))$ — the control and state variables — will also approach the optimal *solution* to $(SCCP)$. While we have enough numerical evidence as well as intuition to believe this is the case, we do not have a proof yet. If such a proof exists, then perhaps it should stretch beyond the scope of the duality theory, since the latter is solely concerned with the value of the solution rather than the solution itself. Moreover, it is likely to involve nondegeneracy of some type.

The application examples in Section 5 all involve the second order cone in the SCCP. Our solution method, however, works for any convex cones, including the popular positive semidefinite matrix cone. It remains open to find interesting applications for such SCCP.

References

[1] E.J. Anderson. *A Continuous Model for Job-Shop Scheduling*, PhD. Thesis, University of Cambridge, Cambridge, U.K., 1978.

[2] B.D.O. Anderson and J.B. Moore. *Optimal Control - Linear Quadratic Methods*, Prentice-Hall, New Jersey, 1989.

[3] E.J. Anderson, P. Nash, and A.F. Perold. Some properties of a class of continuous linear programs, *SIAM J. Control & Opt.*, 21 (1983) pp. 258-265.

[4] E.J. Anderson, P. Nash. *Linear Programming in Infinite-Dimensional Spaces: Theory and Applications*, John Wiley & Sons, 1987.

[5] E.J. Anderson, A.B. Philpott. A continuous-time network simplex algorithm, *Networks*, 19 (1989), pp. 395-425.

[6] E.J. Anderson, A.B. Philpott. On the solutions of a class of continuous linear programs, *SIAM J. Control and Optimization*, 32 (1994) pp. 1289-1296.

[7] R.E. Bellman. Bottleneck problem and dynamic programming, *Proc. Nat. Acad. Sci.*, 39 (1953) pp. 947-951.

[8] R.E. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, NJ, 1957.

[9] A. Ben-Tal, and A. Nemirovski. Robust convex optimization, *Mathematics of Operations Research*, 23 (1998), pp. 769–805.

[10] H. Chen and D.D. Yao. Dynamic scheduling control of a multi-class fluid network, *Operations Research*, 41 (1993), pp. 1104-1115.

[11] H. Chen and D.D. Yao. *Fundamentals of Queueing Networks: Performance, Asymptotics and Optimization*, Springer-Verlag, New York, Applications of Mathematics 46, 2001.

[12] X. Chen and X.Y. Zhou. Stochastic linear-quadratic control with conic control constraints on an infinite time horizon. *SIAM J. Control Optim.*, 43 (2004) pp. 1120-1150.

[13] V. Chvátal. *Linear Programming.* W.H. Freeman and Company, New York, 1980.

[14] J.G. Dai. On positive harris recurrence of multi-class queueing networks: a unified approach via fluid limit models, *Annals of Applied Probability*, 5 (1995), pp. 49-77.

[15] J.G. Dai and G. Weiss. Stability and Instability of Fluid Models for Certain Reentrant Lines, *Mathematics of Operations Research*, 21 (1996), pp. 115-134.

[16] L. Fleischer and J. Sethuraman. Efficient algorithms for separated continuous linear programs: the multi-commodity flow problem with holding costs and extensions, *Mathematics of Operations Research*, forthcoming.

[17] Y. Hu and X.Y. Zhou. Constrained stochastic LQ control with random coefficients, and application to portfolio selection, *SIAM J. Control Optim.*, 44 (2005) pp. 444-466.

[18] X. Luo and D. Bertsimas. A new algorithm for state-constrained separated continuous linear programs, *SIAM J. Control Optim.*, 37 (1998) pp. 177-210.

[19] Yu. Nesterov and A. Nemirovski. *Interior Point Polynomial Methods in Convex Programming*, SIAM, Philadelphia, 1994.

[20] M.C. Pullan. An algorithm for a class of continuous linear programs, *SIAM J. Control Optim.*, 31 (1993), pp. 1558-1577.

[21] M.C. Pullan. Forms of optimal solutions for separated continuous linear programs, *SIAM J. Control Optim.*, 33 (1995), pp. 1952-1977.

[22] M.C. Pullan. A duality theory for separated continuous linear programs, *SIAM J. Control Optim.*, 34 (1996), pp. 931-965.

[23] M.C. Pullan. Linear optimal control problems with piecewise analytic solutions, *Journal of Mathematical Analysis and Applications*, 197 (1996), pp. 207-226.

[24] M.C. Pullan. A study of general dynamic network programs with arc time-delays, *SIAM J. Optim.*, 7 (4) (1997), pp. 889-912.

[25] M.C. Pullan. Convergence of a general class of algorithms for separated continuous linear programs, *SIAM J. Optim.*, 10 (3), (2000) pp. 722-731.

[26] M.C. Pullan. An extended algorithm for separated continuous linear programs, *Mathematical Programming*, series A, 93 (2002) pp. 415-451.

[27] A. Shapiro. On duality theory of conic linear problems, Chapter 7 in *Semi-Infinite Programming*, eds. M.A. Goberna and M.A. Lopez, Kluwer Academic Publishers, Netherlands, pp. 135-165, 2001.

[28] J.F. Sturm. Using SeDuMi 1.02, a matlab toolbox for optimization over symmetric cones, *Optimization Methods and Software*, 11-12 (1999), pp. 625-653.

[29] X. Wang. *Theory and Algorithms for Separated Continuous Linear Programming and Its Extensions*, PhD. Thesis, The Chinese University of Hong Kong, Hong Kong, China, 2005.

[30] G. Weiss. *A simplex based algorithm to solve separated continuous linear programs.* Working Paper. Department of Statistics, Haifa University, 2000.