Separated Continuous Conic Programming: Theory and Method

SHUZHONG ZHANG
(joint work with Xiaoqing Wang and David Yao)

Department of Systems Engineering and Engineering Management
The Chinese University of Hong Kong

Workshop on Semidefinite Programming
NUS, Singapore
January 12, 2006

Outline

- Separated Continuous Linear Programming
- Separated Continuous Conic Programming
- Motivations
- Duality
- Solution Methods
- Conclusions

Separated Continuous Linear Programming

A Continuous-Time LP Model:

$$
\begin{array}{lll}
(SCLP) & \max & \int_0^T ((\gamma + (T-t)c)' u(t) + d' x(t)) dt \\
& \text{s.t.} & \alpha + ta - \int_0^t G u(s) ds - F x(t) \ge 0, \\
& & b - H u(t) \ge 0, \\
& & u(t) \ge 0, \ x(t) \ge 0, \ t \in [0, T].
\end{array}
$$

History

- Bellman (1953)
- Andersen (1978)
- Andersen and Nash (1987)
- Anstreicher, Pullan, Bertimas, Luo, Fleischer, Sethuramen, ...
- Shapiro (2001)
- Weiss (2000)

A Fact to Note

If (SCLP) has an optimal solution, then there is a particular optimal solution with the following property: there is a finite partition of $[0, T]$ and $u(\cdot)$ is piecewise constant over each sub-interval.

Problem (SCLP) is thus essentially a finite problem!

Separated Continuous Conic Programming

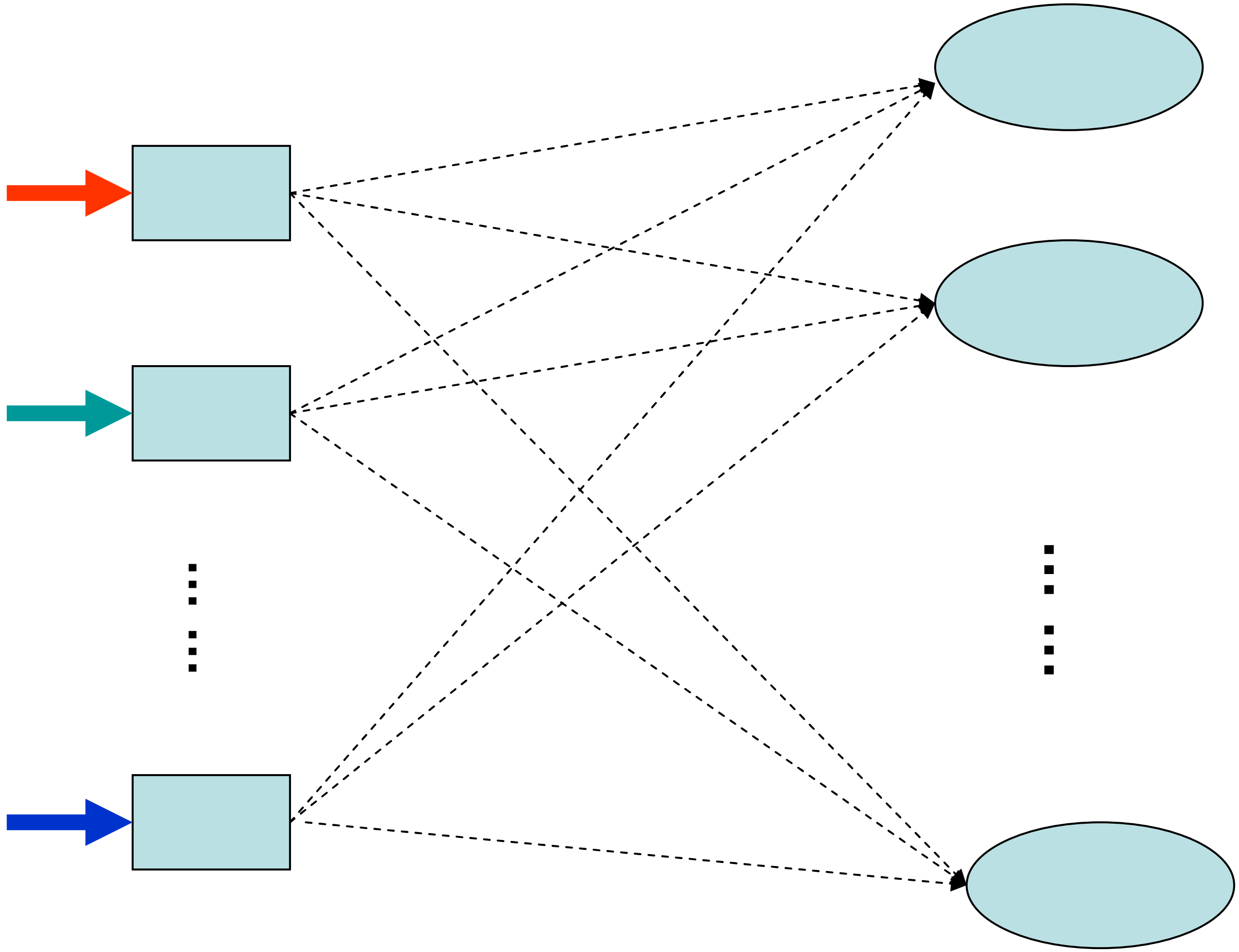
The Model:

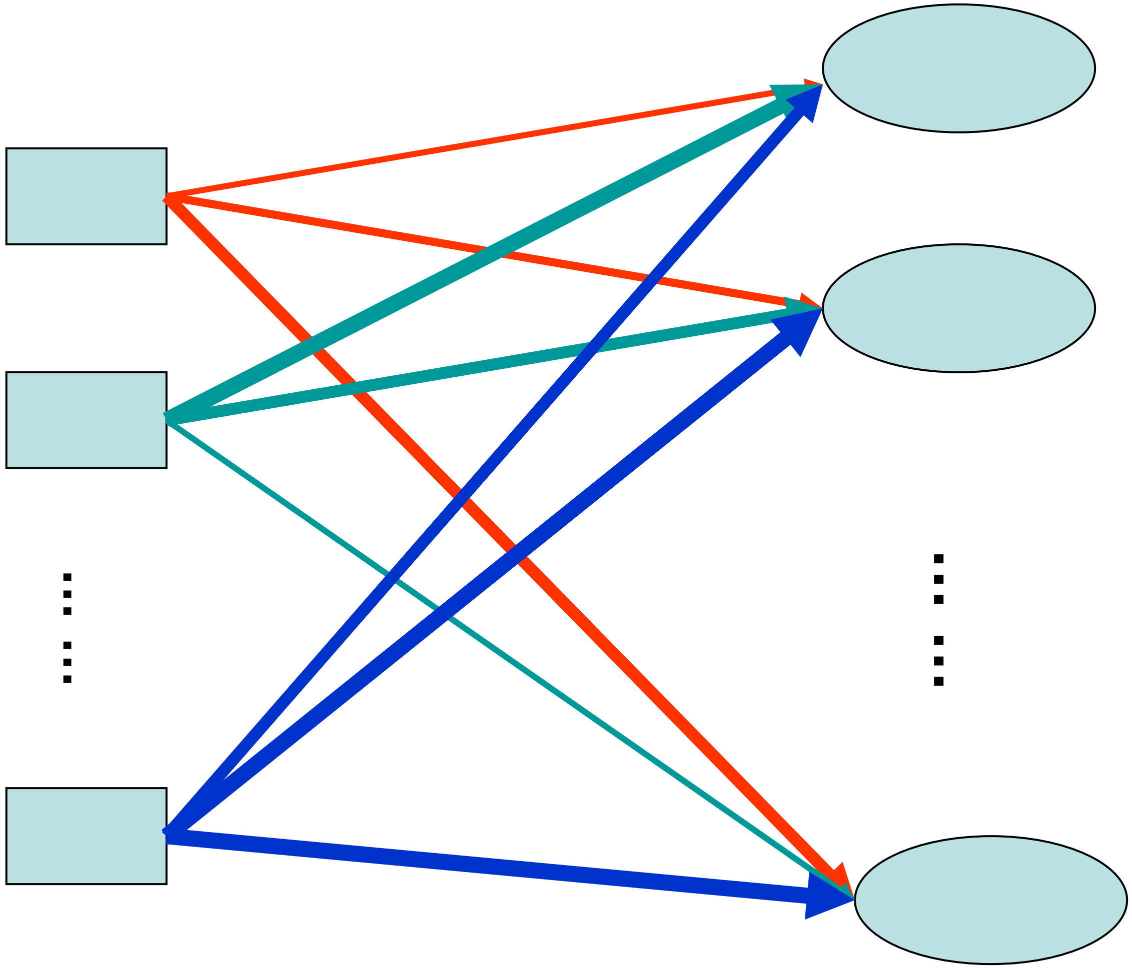
$$
\begin{array}{lll}
(SCCP) & \max & \int_0^T ((\gamma + (T-t)c)'u(t) + d'x(t))dt \\
& \text{s.t.} & \alpha + ta - \int_0^t Gu(s)ds - Fx(t) \in \mathcal{K}_1, \\
& & b - Hu(t) \in \mathcal{K}_2, \\
& & u(t) \in \mathcal{K}_3,\ x(t) \in \mathcal{K}_4,\ t \in [0,T],
\end{array}
$$

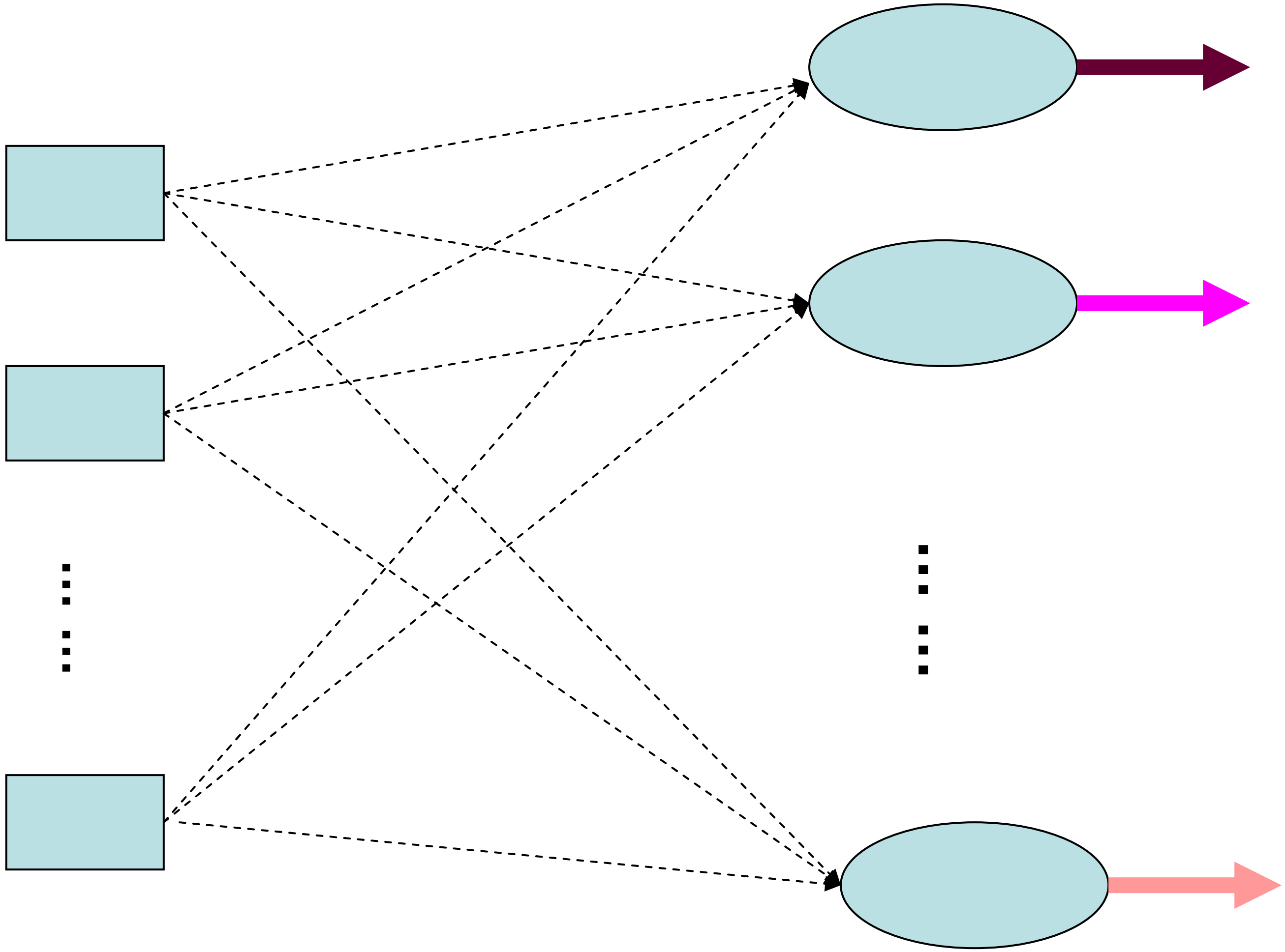
where $\mathcal{K}_i,\ i = 1,2,3,4$, are given closed convex cones.

Applications

- Queueing networks
- Production planning and scheduling
- Economic systems
- Traffic control
- Water resources control
- ... ...







Duality

Like ordinary conic programming, there is a symmetric duality structure for SCCP:

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

Weak duality holds in a straightforward manner.

How about the strong duality relation?

Consider the following two auxiliary problems

$$\begin{array}{lll} (CP_0) & \max & c'u + d'x \\ & \text{s.t.} & \alpha + Ta - Gu - Fx \in \mathcal{K}_1, \\ & & Tb - Hu \in \mathcal{K}_2, \\ & & u \in \mathcal{K}_3, \; x \in \mathcal{K}_4 \end{array}$$

and

$$\begin{array}{lll} (CP_0^*) & \min & a'p + b'q \\ & \text{s.t.} & G'p + H'q - (\gamma + Tc) \in \mathcal{K}_3^*, \\ & & F'p - Td \in \mathcal{K}_4^*, \\ & & p \in \mathcal{K}_1^*, \; q \in \mathcal{K}_2^*. \end{array}$$

Theorem (Wang, Yao, and Z., 2005). If (CP_0) and (CP_0^*) satisfy the Slater condition with finite optimum values, then the strong duality holds between $(SCCP)$ and $(SCCP^*)$.

Sketch of the proof:

Let π_m be a partition of the interval $[0, T]$ into m subintervals.

Let t_i be the ith grid of the partition.

Consider the discretized version of $(SCCP)$, which we shall call $(CP(\pi_m))$:

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

Similarly, we introduced the discretized version of $(SCCP^*)$, to be called $(CP^*(\pi_m))$:

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

Remarks:

- $(CP(\pi_m))$ is a discretized version of $(SCCP)$, and $(CP^*(\pi_m))$ is a discretized version of $(SCCP^*)$. But they are not a dual pair of conic programs!
- However, the dual of $(CP(\pi_m))$ is closely related to $(CP^*(\pi_m))$.
- The feasible set of (CP_0) is identical to the feasible set of $(CP(\pi_1))$, and the feasible set of (CP_0^*) is identical to the feasible set of $(CP^*(\pi_1))$, with slightly altered objective functions.

Further Facts:

- If there is feasible solution of $(CP(\pi_m))$, then the solution can be extended to a feasible solution of $(SCCP)$ with precisely the same objective value.
- The reverse is also true. That is, if there is a piecewise constant function $u(\cdot)$ (according to π_m) that is a feasible control for $(SCCP)$, then this function can be truncated as a feasible solution of $(CP(\pi_m))$ with the same objective value as $u(\cdot)$ is for $(SCCP)$.
- (CP_0) satisfies the Slater condition implies that so is true for all $(CP(\pi_m))$ with any positive integer value m.

By the weak duality relationship and the nature of discretization, we have

$$v(CP(\pi_m)) \le v(SCCP) \le v(SCCP^*) \le v(CP^*(\pi_m)).$$

The question is how to bound the gap $v(CP^*(\pi_m)) - v(CP(\pi_m))$.

Now we turn to a particular case where π_m is a trivial partition.

As we remarked before,

- Although $(CP(\pi_m))$ and $(CP^*(\pi_m))$ are not exactly a dual pair,
- they almost are.

... ...

The key is to establish the following relationship:

There is $\Gamma > 0$ such that

$$v(CP^*(\pi_m)) - v(CP(\pi_m)) \leq \frac{\Gamma}{m}.$$

This is done by investigating the relationships between a sequence of bridging conic programs:

$$
\begin{array}{ll}
\min & \hat{g}_u'\hat{p} + \hat{f}_u'\hat{q} \\
\text{s.t.} & \hat{A}_1\hat{p} + \hat{A}_2\hat{q} - \hat{h}_u \in \underbrace{\mathcal{K}_3^* \times \cdots \times \mathcal{K}_3^*}_{m}, \\
& \hat{A}_3\hat{p} - \hat{d}_u \in \underbrace{\mathcal{K}_4^* \times \cdots \times \mathcal{K}_4^*}_{m}, \\
& \hat{p} \in \underbrace{\mathcal{K}_1^* \times \cdots \times \mathcal{K}_1^*}_{m},\ \hat{q} \in \underbrace{\mathcal{K}_2^* \times \cdots \times \mathcal{K}_2^*}_{m}
\end{array}
$$

and its dual

$$
\begin{array}{ll}
\max & \hat{h}_u'\hat{u} + \hat{d}_u'\hat{x} \\
\text{s.t.} & \hat{g}_u - \hat{A}_1'\hat{u} - \hat{A}_3'\hat{x} \in \underbrace{\mathcal{K}_1 \times \cdots \times \mathcal{K}_1}_{m}, \\
& \hat{f}_u - \hat{A}_2'\hat{u} \in \underbrace{\mathcal{K}_2 \times \cdots \times \mathcal{K}_2}_{m}, \\
& \hat{u} \in \underbrace{\mathcal{K}_3 \times \cdots \times \mathcal{K}_3}_{m}, \;\; \hat{x} \in \underbrace{\mathcal{K}_4 \times \cdots \times \mathcal{K}_4}_{m}.
\end{array}
$$

and another conic program

$$
\begin{aligned}
\min \quad & \hat{g}_l' \hat{p} + \hat{f}_l' \hat{q} \\
\text{s.t.} \quad & \hat{A}_1 \hat{p} + \hat{A}_2 \hat{q} - \hat{h}_l \in \underbrace{\mathcal{K}_3^* \times \cdots \times \mathcal{K}_3^*}_{m}, \\
& \hat{A}_3 \hat{p} - \hat{d}_l \in \underbrace{\mathcal{K}_4^* \times \cdots \times \mathcal{K}_4^*}_{m}, \\
& \hat{p} \in \underbrace{\mathcal{K}_1^* \times \cdots \times \mathcal{K}_1^*}_{m}, \ \hat{q} \in \underbrace{\mathcal{K}_2^* \times \cdots \times \mathcal{K}_2^*}_{m}
\end{aligned}
$$

and its dual

$$
\begin{aligned}
\max \quad & \hat{h}_l' \hat{u} + \hat{d}_l' \hat{x} \\
\text{s.t.} \quad & \hat{g}_l - \hat{A}_1' \hat{u} - \hat{A}_3' \hat{x} \in \underbrace{\mathcal{K}_1 \times \cdots \times \mathcal{K}_1}_{m}, \\
& \hat{f}_l - \hat{A}_2' \hat{u} \in \underbrace{\mathcal{K}_2 \times \cdots \times \mathcal{K}_2}_{m}, \\
& \hat{u} \in \underbrace{\mathcal{K}_3 \times \cdots \times \mathcal{K}_3}_{m}, \ \hat{x} \in \underbrace{\mathcal{K}_4 \times \cdots \times \mathcal{K}_4}_{m},
\end{aligned}
$$

where

$$\hat{A}_1 = \begin{pmatrix} G' & & & \\ G' & G' & & \\ \cdots & & & \\ G' & G' & \cdots & G' \end{pmatrix}, \quad \hat{A}_2 = \begin{pmatrix} H' & & & \\ & H' & & \\ & & \ddots & \\ & & & H' \end{pmatrix}, \quad \hat{A}_3 = \begin{pmatrix} F' & & & \\ & F' & & \\ & & \ddots & \\ & & & F' \end{pmatrix}$$

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

and

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

In a precise form, what we have established is the following:

Theorem (Wang, Yao, Z., 2005).

$$v(CP^*(\pi_m)) - v(CP(\pi_m)) \le \frac{\Gamma}{m}$$

with

$$\Gamma = v(CP_0^*) - v(CP_0) + b'\hat{q}(t_0) - d'\hat{x}(t_0).$$

Solution Methods

The above proof procedure actually suggests solution methods.

Algorithm SCCP (primal)

Let δ be the pre-defined precision.

Step 0 Choose

$$\hat{x}(t_0) \in \{x \mid \alpha - Fx \in \mathcal{K}_1\} \cap \mathcal{K}_4 \text{ and } \hat{q}(t_0) \in \{q \mid H'q - \gamma \in \mathcal{K}_3^*\} \cap \mathcal{K}_2^*.$$

Step 1 Let

$$m := \lceil \frac{T}{2\delta} (v(CP_2) - v(CP_1) + b'\hat{q}(t_0) - d'\hat{x}(t_0)) \rceil$$

and solve $(CP(\pi_m))$.

Step 2 Use the extension of the optimal solution obtained for $(CP(\pi_m))$ to construct a feasible solution for $(SCCP)$. Stop.

Algorithm SCCP (primal-dual)

Let δ be the pre-defined precision.

Step 0 Choose

$$\hat{x}(t_0) \in \{x \mid \alpha - Fx \in \mathcal{K}_1\} \cap \mathcal{K}_4 \text{ and } \hat{q}(t_0) \in \{q \mid H'q - \gamma \in \mathcal{K}_3^*\} \cap \mathcal{K}_2^*.$$

Let $m = 1$, and go to Step 1.

Step 1 Solve $(CP(\pi_m))$ and $(CP^*(\pi_m))$.

Step 2 If $v(CP^*(\pi_m)) - v(CP(\pi_m)) \leq \delta$ stop; otherwise, let $m := 2m$ and go to Step 1.

Numerical Example

An input/output model.

There are 8 assets, $k = 1, \ldots, 8$, 12 activities, $j = 1, \ldots, 12$, and 5 resources. Initially, there are some inventories for these 8 assets. We also can use these 12 activities to produce more inventories for some of these 8 assets, at the price of consuming the other assets within these 8 assets.

The data matrices.

c	2	7	3	5	2	7	2	4	6	3	4	3	α	a
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-2.6	0.0	0.0	36.0	1.2
	0.0	0.0	0.0	-2.8	0.0	0.0	0.0	3.0	-3.7	-1.1	-3.4	8.1	28.0	1.1
	2.9	3.1	7.4	8.9	0.0	0.0	-3.5	-2.9	-3.7	0.0	0.0	0.0	31.0	1.2
G	-1.9	0.0	0.0	0.0	0.0	5.4	8.4	0.0	4.5	3.6	3.3	-1.6	29.0	1.3
	0.0	0.0	0.0	-1.5	0.0	-3.4	-2.2	-1.2	0.0	0.0	-3.5	-3.2	26.0	1.0
	-2.2	0.0	0.0	0.0	-2.5	0.0	0.0	-2.8	-2.7	0.0	0.0	0.0	30.0	1.9
	0.0	-1.9	0.0	0.0	0.0	0.0	-3.7	0.0	0.0	0.0	0.0	0.0	26.0	1.4
	0.0	0.0	0.0	-3.5	5.2	-2.7	0.0	0.0	-3.7	-1.9	0.0	0.0	34.0	1.3
														b
	6.5	8.0	6.0	6.4	5.4	7.8	6.5	5.6	7.4	3.6	7.3	6.9		106.0
	0.0	3.9	5.8	4.8	0.0	0.0	0.0	7.4	0.0	7.3	0.0	3.8		66.0
H	0.0	0.0	3.1	0.0	5.9	0.0	5.8	6.4	0.0	7.1	5.5	0.0	$\leq$	115.0
	4.9	0.0	7.5	5.2	4.6	7.4	0.0	6.9	0.0	0.0	0.0	6.4		86.0
	7.0	4.3	4.9	0.0	0.0	0.0	3.6	0.0	7.5	0.0	6.2	0.0		112.0

Numerical results of our method.

	Number of Intervals							
	1		3		7		11	
T	value	gap	value	gap	value	gap	value	gap
0.5	11.8340	0.0140	11.8433	0.0047	11.8460	0.0020	11.8467	0.0008
1.3	78.6555	0.9756	79.2561	0.1547	79.3402	0.0757	79.3609	0.0206
1.4	90.7811	1.4709	91.7172	0.3131	91.8267	0.0799	91.8360	0.0490
2.2	218.7587	7.8659	221.7020	1.5392	222.1393	0.3631	222.2853	0.0944
3.7	606.2048	18.1425	612.0795	3.3616	613.3126	0.6850	613.5420	0.1761
4.4	853.1338	21.9009	858.6794	6.1350	861.1191	0.7014	861.4035	0.2505
4.8	1006.9197	30.1000	1017.3889	8.8249	1021.1853	0.6306	1021.1081	0.6635
5.2	1155.5733	57.1707	1186.9810	12.4132	1192.1926	2.5073	1193.3943	0.8260

More Applications

Sign-constrained linear quadratic control

$$
\begin{aligned}
\min \quad & \int_0^T (x(t)'Qx(t) + u(t)'Ru(t))dt \\
\text{s.t.} \quad & \dot{x}(t) = Bu(t) + b, \\
& x(0) = \alpha, \alpha \geq 0, \\
& a - Hu(t) \in \Re_+^{n_1}, \\
& u(t) \in \Re_+^m, x(t) \in \Re_+^n.
\end{aligned}
$$

The problem can be reformulated as

$$
\begin{array}{ll}
\min & \int_0^T (y_0(t) + z_0(t))dt \\
\text{s.t.} & \alpha + bt + \int_0^t Bu(s)ds - x(t) = 0, \\
& a - Hu(t) \in \Re^{n_1}_+, \\
& u(t) \in \Re^m_+,\ x(t) \in \Re^n_+, \\
& \begin{pmatrix} x_0(t) \\ Q^{\frac{1}{2}} x(t) \end{pmatrix} \in \mathrm{SOC}(n+1), \\
& \begin{pmatrix} 1 + y_0(t) \\ 1 - y_0(t) \\ 2x_0(t) \end{pmatrix} \in \mathrm{SOC}(3), \\
& \begin{pmatrix} u_0(t) \\ R^{\frac{1}{2}} u(t) \end{pmatrix} \in \mathrm{SOC}(m+1), \\
& \begin{pmatrix} 1 + z_0(t) \\ 1 - z_0(t) \\ 2u_0(t) \end{pmatrix} \in \mathrm{SOC}(3).
\end{array}
$$

A fluid network example

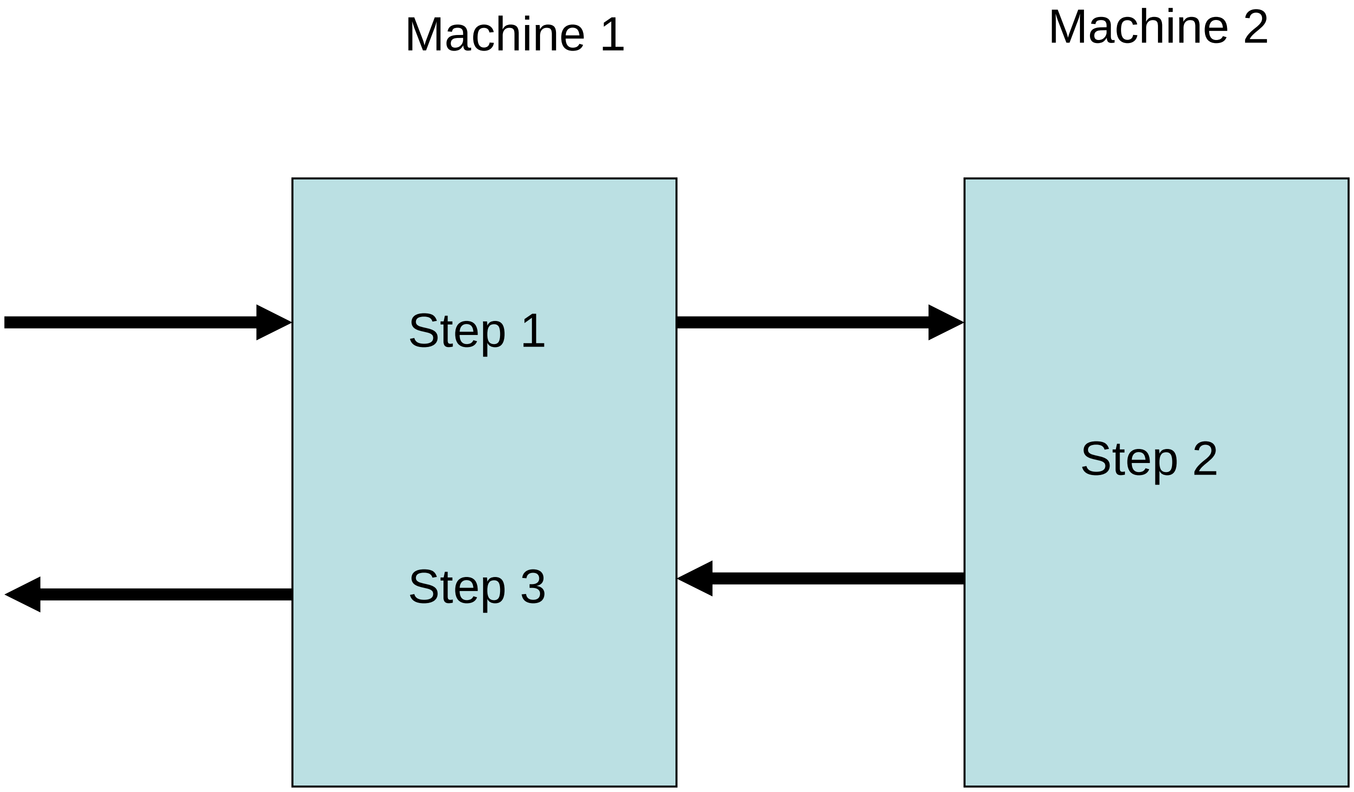
A network processes a continuous flow of jobs at two machines.

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 $d = (30\ 10\ 80)'$.


Machine 1
Machine 2
Step 1
Step 2
Step 3

The problem can be formulated as:

$$\begin{aligned} \min \quad & \int_0^T [(x(t)-d)'(x(t)-d)]dt \\ \text{s.t.} \quad & \int_0^t Gu(s)ds + x(t) = \alpha + ta, \\ & b - Hu(t) \geq 0, \\ & u(t) \geq 0,\ x(t) \geq 0,\ t \in [0,T], \end{aligned}$$

where

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

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

	Number of Intervals							
	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

Objective values and error bounds (e.b.) for the SCCP.

Robust separated continuous linear programming (SCLP)

$$
\begin{array}{lll}
(SCLP) & \max & \int_0^T ((\gamma + (T-t)c)' u(t) + d' x(t)) dt \\
& \text{s.t.} & \alpha + ta - \int_0^t Gu(s)ds - Fx(t) \in \Re^n_+, \\
& & b - Hu(t) \in \Re^l_+, \\
& & u(t) \in \Re^m_+, \; x(t) \in \Re^k_+, \; t \in [0, T].
\end{array}
$$

Suppose that F is subject to the uncertainty set
$F \in Y = \{F^0 + \sum_{j=1}^{k_3} y_j F^j \mid y' y \le 1\}$.

The robust version of the constraint then becomes

$$
\begin{pmatrix} \alpha_i \\ 0 \end{pmatrix} + t \begin{pmatrix} a_i \\ 0 \end{pmatrix} - \int_0^t \begin{pmatrix} G_i \\ 0 \end{pmatrix} u(s) ds - \begin{pmatrix} F_i^0 \\ -F_i \end{pmatrix} x(t) \in \text{SOC}(1+k_3),
$$

for $i = 1, 2, \ldots, n$.

Conclusions

- A novel model for continuous time optimization
- Practical solvability
- Computable and verifiable error bounds
- Ample opportunities for applications
- Beautiful theoretical structures
- Only a beginning ... ...

Thank you!

and

Q & A?