Mathematical Programming Applications of Linear Programs with Complementarity Constraints[1]

John E. Mitchell[1], Jing Hu[1] and Jong-Shi Pang[2]

[1]Department of Mathematical Sciences
RPI, Troy, NY 12180 USA

[2]IESE, UIUC

October 15, 2008

[1]Supported by AFOSR.

Linear Programs with Complementarity Constraints

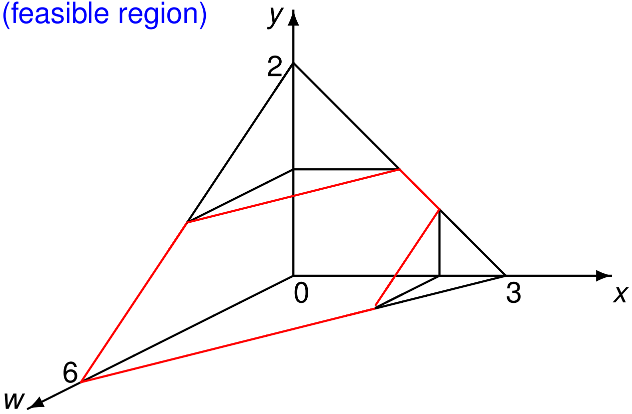
Find $(x, y) \in \mathbb{R}^n \times \mathbb{R}^m$ to globally solve the linear program with complementarity constraints (LPCC):

$$
\begin{array}{ll}
\underset{(x,y)}{\text{minimize}} & c^T x + d^T y \\
\text{subject to} & Ax + By \ \geq \ f \\
\text{and} & 0 \ \leq \ y \ \perp \ q + Nx + My \ \geq \ 0,
\end{array}
$$

An example

$$\begin{array}{ll} \min & 3x - 5y \\ \text{subject to} & 2x + 3y + w = 6, \quad y \le 1, \quad 0 \le x \le 2 \\ & 0 \le y \perp w \ge 0 \end{array}$$

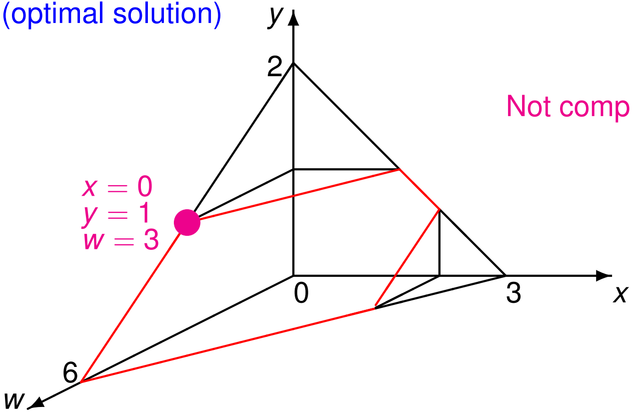
LP relaxation:
(feasible region)



An example

$$\begin{array}{ll} \min & 3x - 5y \\ \text{subject to} & 2x + 3y + w = 6, \quad y \leq 1, \quad 0 \leq x \leq 2 \\ & 0 \leq y \perp w \geq 0 \end{array}$$

LP relaxation:
(optimal solution)

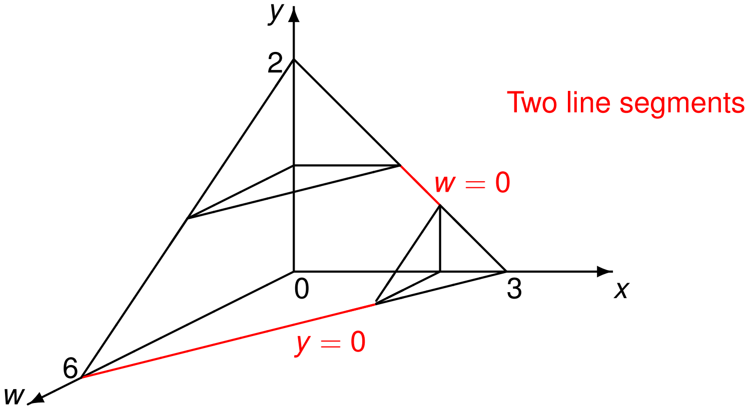


Not complementary!

An example

$$\begin{array}{ll} \min & 3x - 5y \\ \text{subject to} & 2x + 3y + w = 6, \quad y \leq 1, \quad 0 \leq x \leq 2 \\ & 0 \leq y \perp w \geq 0 \end{array}$$

The LPCC:



Fundamental importance

The LPCC plays the same important role in disjunctive nonlinear programs as a linear program does in convex programs.

Additionally, it has many applications of its own:

Novel paradigms in mathematical programming

- hierarchical optimization
- inverse optimization
- parameter identification/model validation in optimization
- optimizing Value-at-Risk and order statistics

Key formulations for

- B-stationary conditions of MPECs
  - verification and computation without MPEC-constraint qualification
- global resolution of nonconvex quadratic programs

More on bilevel programs

Bilevel program:

$$\begin{aligned} \min_{x,y} \quad & f(x,y) \\ \text{s.t.} \quad & g(x,y) \le 0 \\ & y \in \operatorname{argmin}_y \{h(y) : r(y) \le v(x)\} \end{aligned}$$

MPCC:

$$\begin{aligned} \min \quad & f(x,y) \\ \text{s.t.} \quad & g(x,y) \le 0 \\ & \nabla h(y) + Jr(y)\lambda = 0 \\ & 0 \le v(x) - r(y) \perp \lambda \ge 0 \end{aligned}$$

Equivalent if $h(y)$ and $r(y)$ are convex. ($Jr(y)$ is Jacobian of $r(y)$.)

Get an LPCC if all of the constraints are linear.

Preliminary observations

An LPCC is equivalent to 2^m linear programs, each called a piece and derived from a subset $\alpha \subseteq \{1, \cdots, m\}$ with complement $\bar{\alpha}$:

$$\text{LP}(\alpha): \quad \begin{array}{|lll|} \hline \underset{(x,y)}{\text{minimize}} & c^T x + d^T y & \\ \text{subject to} & Ax + By \geq f & \\ & (q + Nx + My)_{\alpha} \geq 0 = y_{\alpha} & \\ \text{and} & (q + Nx + My)_{\bar{\alpha}} = 0 \leq y_{\bar{\alpha}} & \\ \hline \end{array}$$

Thus, there are 3 states of an LPCC in general:

- infeasibility–all pieces are infeasible
- unboundedness–one piece is feasible and unbounded below
- global solvability–one piece is feasible and all feasible pieces are bounded below.

Equivalent Integer Program

Given a sufficiently large parameter θ and denoting the vector of ones by **1**, get an equivalent mixed integer problem:

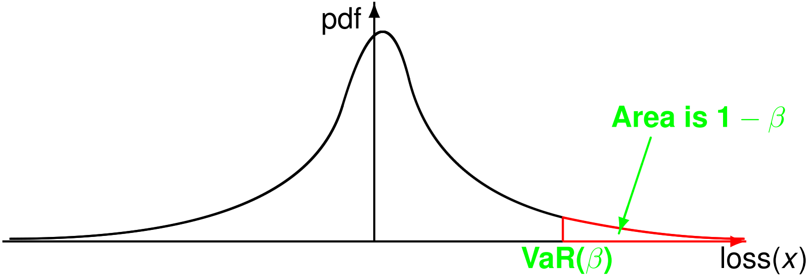
$$
\begin{array}{lrcl}
\underset{(x,y,z)}{\text{minimize}} & c^T x + d^T y & & \\
\text{subject to} & Ax + By & \geq & f \\
& \theta z \geq w := q + Nx + My & \geq & 0 \\
& \theta(\mathbf{1} - z) \geq y & \geq & 0 \\
\text{and} & z & \in & \{0,1\}^m
\end{array}
$$

If good bounds can be found on all components of w and y then this can be solved as an MIP.

Of interest: (some of) the variables are unbounded: see Jing's talk.

Optimizing Value-at-Risk (VaR)

Choose *x* to minimize VaR(β) for some β.



Rockafellar and Uryasev: VaR is argmin of a CVaR problem.
(See also Pang and Leyffer; Luedtke, Ahmed and Shapiro,...)
Get LPEC formulation when have finite number of scenarios.

LPCC formulation of VaR

$$\begin{array}{llcll} \min_{\alpha,x,\tau,s,\lambda} & \alpha & & & \\ \text{subject to} & 0 & \leq & \tau_i \perp \frac{p_i}{1-\beta} - \lambda_i \geq 0 & i = 1,\ldots,k \\ & 0 & \leq & \lambda_i \perp s_i := \alpha + \tau_i - x^T r^i \geq 0 & i = 1,\ldots,k \\ & 1 & = & \sum_{i=1}^k \lambda_i & \\ & & & x \in X & \end{array}$$

α is the Value-at-Risk.

k is number of scenarios.

r^i is the loss in scenario i.

Computational Results for VaR

$\beta = 0.95$

n	k	time
5	999	247
20	401	701
20	501	1998
50	301	2935
50	401	14449
100	201	1762

$\beta = 0.99$

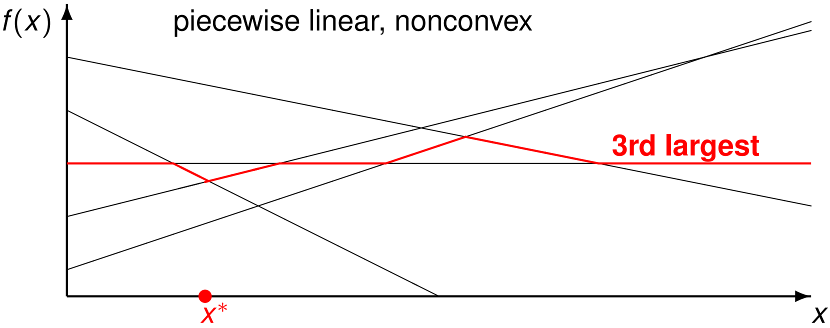
n	k	time
10	801	50
10	1001	98
20	801	60
20	1001	177
50	801	1010
50	1001	1162
100	801	4737

n: dimension of x.

k: number of scenarios, randomly generated from a random return vector and covariance matrix.

time: in seconds on a Mac Pro.

Minimizing the kth largest



$$
\begin{array}{lll}
\min_{\alpha,\beta,x,s} & \alpha & \\
\text{subject to} & \alpha + \beta_i \geq f_i(x) & i = 1,\ldots,m \\
& 0 \leq \beta \perp s \geq 0 & \\
& \mathbf{1}^T s = m - k + 1 & \\
& 0 \leq s \leq 1, \quad x \in P &
\end{array}
$$

Computational Results for Order Statistics

Linear functions on $0 \leq x \leq \mathbf{1}$, so can get upper bound on β.
Minimize the median so $k = m/2$. Randomly generated functions.
Formulate as an MIP, solve with CPLEX 11.0.

n	m	time
20	20	0
40	20	1
60	20	1
80	20	2
100	20	3
20	40	3
40	40	20
60	40	63
80	40	413
100	40	624

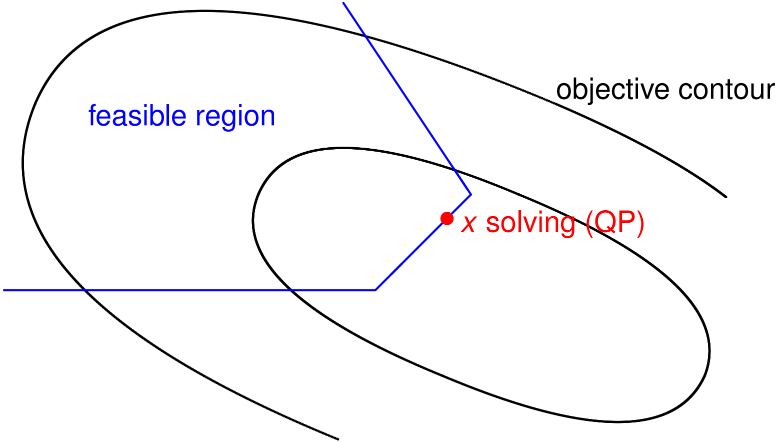
n	m	time
20	60	61
40	60	525
60	60	8307
20	80	343

n: dimension of x

m: number of functions

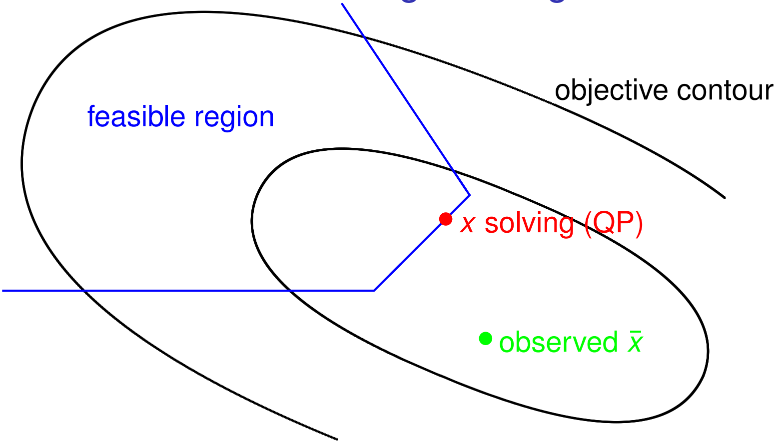
time: seconds on a Mac Pro (average of three runs)

Inverse Convex Quadratic Programming Problems



$$\begin{array}{ll} \min_x & \bar{c}^T x + \frac{1}{2} x^T Q x \\ \text{s.t.} & Ax \leq \bar{b} \qquad\qquad (QP) \end{array}$$

Inverse Convex Quadratic Programming Problems



$$\begin{array}{lll} \min_x & \bar{c}^T x + \frac{1}{2} x^T Q x & \\ \text{s.t.} & Ax \leq \bar{b} & (QP) \end{array}$$

Modify (c, b, x) so x solves (QP), (c, b, x) close to desired values.

Computational Results for Inverse QPs

Use constraints $0 \leq x \leq u$, with upper bound u variable.

$$
\begin{array}{ll}
\min_{x,c,u,y,w} & ||(x,c,u) - (\bar{x},\bar{c},\bar{u})||_1 \\
\text{subject to} & 0 \leq x \perp w := c + Qx + y \geq 0 \\
& 0 \leq u - x \perp y \geq 0 \\
& (c,u) \in P
\end{array}
$$

Choose P to be a polytope. Can then get bounds on y and w.

So, solve as an MIP using CPLEX 11.0.

$Q = MM^T$ generated with varying levels of sparsity.

Computational Results for Inverse QPs

n	NZ	time
20	5	0
20	7	0
20	14	0
20	27	0
20	83	0
20	92	0
40	48	1
40	53	1
40	172	1
40	206	1
40	370	10
40	499	17

n	NZ	time
60	85	0
60	381	20
60	417	23
60	473	5
60	813	56
80	86	1
80	283	6
80	302	4
80	704	3671
100	139	1
100	489	26
100	813	1832
100	1073	70674

n: dimension of x, c, and u.

NZ: number of nonzeroes in Q.

time: in seconds on a Mac Pro.

General LPCCs: recall the formulation

Find $(x, y) \in \mathbb{R}^n \times \mathbb{R}^m$ to globally solve the linear program with complementarity constraints (LPCC):

$$\begin{array}{ll} \underset{(x,y)}{\text{minimize}} & c^T x + d^T y \\ \text{subject to} & Ax + By \geq f \\ \text{and} & 0 \leq y \perp q + Nx + My \geq 0, \end{array}$$

General LPCCs with logical Benders decomposition

LPCCs with $B = 0$, $A \in \Re^{200\times300}$, and 300 complementarities.

# IPs	Time	$LPCC_{min}$	# LPs	rx-cnt	rx-dual	rx-master
2	18.61	2478.2256	140	122	1	17
3	67.02	3270.1844	513	413	16	84
2	30.84	3660.5412	237	205	13	19
3	54.27	3176.4108	506	427	10	69
2	5.06	2959.9495	23	20	2	1
4	38.25	2672.5709	383	334	3	46
0	0.23	2617.2638	0	0	0	0
2	16.53	2771.2372	134	131	1	2
2	25.28	2847.6926	197	188	1	8
3	46.94	3230.9896	436	361	2	73

Solved on a Dell Core Duo CPU 2.33 GHz 1.95 GB of RAM.
rx-cnt: sparsification LPs. rx-dual: LPs to find ray cuts. rx-master: LPs to find point cuts.

Conclusions

The LPCC has broad applicability. We have two approaches for finding global solutions:

> *When good bounds are available, formulate as a fixed charge mixed integer program and solve directly.*
> *When it is not easy to find good bounds, use a logical Benders decomposition approach.*

The complementarities can be further exploited by **lift-and-project** or **RLT**. This is very helpful in the logical Benders approach. In the direct MIP approach, it reduces the size of the tree, but (to date) we have not found it to reduce the runtime.

Further work: Extend the theory and computational work to more general classes of mathematical programs with equilibrium constraints.