Applications of the Linear Complementarity Problem

Yoni Nazarathy, May 2017.

The Linear Complementarity Problem (LCP)

Data: $q \in \mathbb{R}^n$, $M \in \mathbb{R}^{n \times n}$.

Find $w, z \in \mathbb{R}^n$ such that:

1. $w - Mz = q$
2. $w, z \geq 0$
3. $w'z = 0$

It is all about choosing a subset

$$w - Mz = q$$

Set $\alpha \subset \{1, \ldots, n\}$ and denote $B(\alpha)$ as matrix with collumns α taken from I and collumns $\bar{\alpha}$ taken from $-M$.

Seek α:

$$B(\alpha)x = q.$$

It is all about choosing a subset

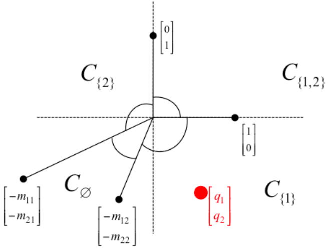
$$\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\begin{bmatrix}w_1\\w_2\end{bmatrix} - \begin{bmatrix}m_{11} & m_{12}\\m_{21} & m_{22}\end{bmatrix}\begin{bmatrix}z_1\\z_2\end{bmatrix} = \begin{bmatrix}q_1\\q_2\end{bmatrix}$$

$$\alpha = \{1,2\}: \quad \begin{bmatrix}1\\0\end{bmatrix} w_1 + \begin{bmatrix}0\\1\end{bmatrix} w_2 = \begin{bmatrix}q_1\\q_2\end{bmatrix}$$

$$\alpha = \{1\}: \quad \begin{bmatrix}1\\0\end{bmatrix} w_1 + \begin{bmatrix}-m_{12}\\-m_{22}\end{bmatrix} z_2 = \begin{bmatrix}q_1\\q_2\end{bmatrix}$$

$$\alpha = \{2\}: \quad \begin{bmatrix}-m_{11}\\-m_{21}\end{bmatrix} z_1 + \begin{bmatrix}0\\1\end{bmatrix} w_2 = \begin{bmatrix}q_1\\q_2\end{bmatrix}$$

$$\alpha = \varnothing: \quad \begin{bmatrix}-m_{11}\\-m_{21}\end{bmatrix} z_1 + \begin{bmatrix}-m_{12}\\-m_{22}\end{bmatrix} z_2 = \begin{bmatrix}q_1\\q_2\end{bmatrix}$$



Complementary cones:

$$C_\alpha = \{y : y = B(\alpha)u, u \geq 0\}$$

Immediate naïve algorithm with complexity $2^n n^3$ *or* $2^n + n^3$

Outline

1. Linear Programming (LP)
2. Quadratic Programming (QP)
3. Bimatrix Games
4. Min-linear equations in queueing networks
5. Some facts about LCP

LP

Linear Programming

Primal-LP	Dual-LP
$\min\ c'x$ s.t. $Ax \geq b$, $x \geq 0$	$\max\ b'y$ s.t. $A'y \leq c$, $y \geq 0$

$$\begin{aligned} \min \quad & c'x \\ \text{s.t.} \quad & Ax \geq b \\ & x \geq 0 \end{aligned} \qquad\qquad \begin{aligned} \max \quad & b'y \\ \text{s.t.} \quad & A'y \leq c \\ & y \geq 0 \end{aligned}$$

$$\begin{aligned} \min \quad & c'x \\ \text{s.t.} \quad & Ax - b = v \\ & x, v \geq 0 \end{aligned} \qquad\qquad \begin{aligned} \max \quad & b'y \\ \text{s.t.} \quad & u = c - A'y \\ & y, u \geq 0 \end{aligned}$$

Theorem (complementary slackness)

Assume x, v, y, u are feasible for primal and dual:

$$x_i u_i = 0, \quad y_i v_i = 0 \qquad \Longleftrightarrow \qquad x \text{ and } y \text{ are optimal}$$

LP as LCP

Set $w = \begin{bmatrix} u \\ v \end{bmatrix}$, $z = \begin{bmatrix} x \\ y \end{bmatrix}$.

Want to find non-negative, w, v such that $Ax - b = v$, $u = c - A'y$ and $w'v = 0$.

$$\begin{bmatrix} u \\ v \end{bmatrix} - \begin{bmatrix} 0 & -A' \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} c \\ -b \end{bmatrix}$$

LP facts (audience to fill in 8,9,10,. . .)

1. Weak duality: dual$^* = y'b \leq y'Ax = x'A'y \leq x'c =$ primal*
2. Strong duality
3. Simplex Algorithm works well on "most" cases
4. Simplex may have exponential inputs
5. Problem has polynomial time algorithm
6. Applications...
7. History...
8. ...
9. ...
10. ...
11. ...
12. ...
13. ...

QP

Quadratic Programming

$$\begin{aligned} \text{min} \quad & x'Dx + c'x \\ \text{s.t.} \quad & Ax \geq b \\ & x \geq 0 \end{aligned}$$

Example: Linear Quadratic MPC

Model:

$$x(k+1) = Ax(k) + Bu(k).$$

Cost:

$$J(u) = \sum_{k=0}^{N-1} x(k)'\,Q\,x(k) + u'(k)\,R\,u(k) + x'(N)\,Q_f\,x(N).$$

Constraints:

$$F \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \leq b.$$

Formulation as a Quadratic Program (QP)

At time k (taken to be 0 for simplicity), given a measured (or estimated) state $x(k)$ we need to solve,

$$min_{u(0),u(1),\dots,u(N-1)} \sum_{k=0}^{N-1} x(k)'\,Q\,x(k)+u'(k)\,R\,u(k)+x'(N)\,Q_f\,x(N)$$

s.t. $x(k+1) = Ax(k) + Bu(k)$ and,

$$F \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \leq b.$$

Formulation as a Quadratic Program (QP)

$$\begin{bmatrix} x(1) \\ x(2) \\ \vdots \\ x(N) \end{bmatrix} = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix} x(0) + \begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & & \vdots \\ \vdots & & \ddots & \\ A^{N-1}B & & \cdots & B \end{bmatrix} \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(N-1) \end{bmatrix}$$

This converts the optimization problem of the MPC controller to one that simply depends on the mN dimensional vector $u(0), \ldots, u(N-1)$.

The general form of a quadratic program (QP) is:

$$\begin{aligned} & min_z z' \tilde{Q} z + \tilde{P} z, \\ s.t. \qquad & \tilde{F} z \leq \tilde{b}. \end{aligned}$$

With a bit of (tedious rearranging) the MPC controller can then be presented as a convex QP in mN decision variables. QPs where $\tilde{Q} > 0$ have a unique solution and are quite efficiently solvable!!!

The Closed Loop System is Non-Linear

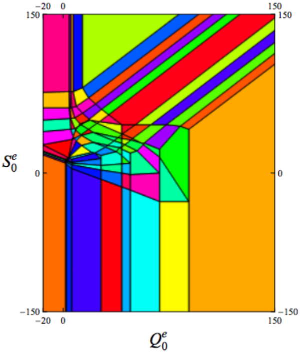
MPC generates a "feedback" control law $u(k) = g\big(x(k)\big)$, where the function $g(\cdot)$ is implicitly defined by the unique solution of the QP. The resulting controlled system,

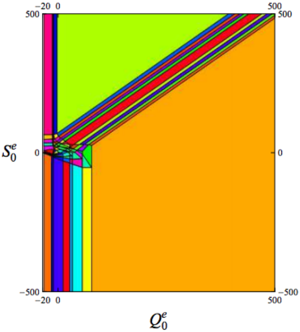
$$x(k+1) = Ax(k) + Bg\big(x(k)\big),$$

is in general non-linear (it is linear if there are no-constraints because then the problem is simply LQR).

The resulting system is piece-wise linear, (Bemporad, Morari, Dua and Pistikopoulus, 2002, "The explicit linear quadratic regulator for constrained systems").

Discrete Queueing Network MPC





Towards QP as LCP

$$\text{QP:} \quad \begin{array}{rl} \min & Q(x) = x'Dx + c'x \\ \text{s.t.} & Ax \geq b \\ & x \geq 0 \end{array}$$

Claim: An optimiser $\bar{x}$ of the QP also optimises the LP:

$$\begin{array}{rl} \min & (c + D\bar{x})'x \\ \text{s.t.} & Ax \geq b \\ & x \geq 0 \end{array}$$

Proof:Let $\hat{x}$ be feasible (of QP and LP) and $\eta \in (0,1)$. Set $x_\eta = \bar{x} + \eta(\hat{x} - \bar{x})$. By convexity it is also feasible.

$Q(x_\eta) - Q(\bar{x}) \geq 0$

$\eta(c' + \bar{x}'D)(\hat{x} - \bar{x}) + \eta^2(\hat{x} - \bar{x})'D(\hat{x} - \bar{x}) \geq 0$

$(c' + \bar{x}'D)(\hat{x} - \bar{x}) \geq -\eta(\hat{x} - \bar{x})'D(\hat{x} - \bar{x})$ for all $\eta \in (0,1)$

$(c' + \bar{x}'D)(\hat{x} - \bar{x}) \geq 0$

$(c' + \bar{x}'D)\hat{x} \geq (c' + \bar{x}'D)\bar{x}$

Now since $\hat{x}$ is arbitrary, $\bar{x}$ must optimise LP. $\square$

QP as LCP

We use the LP of the QP:

$$\begin{aligned} \min \quad & (c + D\bar{x})'x \\ \text{s.t.} \quad & Ax \geq b \\ & x \geq 0 \end{aligned}$$

As with LCP of LP:

$$\begin{bmatrix} u \\ v \end{bmatrix} - \begin{bmatrix} 0 & -A' \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} c + Dx \\ -b \end{bmatrix}$$

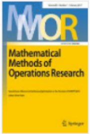
That is, setup and LCP with,

$$M = \begin{bmatrix} D & -A' \\ A & 0 \end{bmatrix}, \qquad q = \begin{bmatrix} c \\ -b \end{bmatrix}.$$

QP facts

1. An example of convex-programming.
2. If D is Positive Semi Definite (PSD), that is $Q(\cdot)$ convex and solved efficiently.
3. If D is not PSD the problem is NP-hard.
4. Applications: ...
5. Algorithms: ...
6. ...
7. ...
8. ...

Another Encounter with QPs



Mathematical Methods of Operations Research
pp 1–32

Scheduling for a processor sharing system with linear slowdown

Authors | Authors and affiliations

Liron Ravner, Yoni Nazarathy

Original Article

First Online: 20 March 2017

DOI: 10.1007/s00186-017-0583-3

Cite this article as:
Ravner, L. & Nazarathy, Y. Math Meth Oper Res (2017). doi:10.1007/s00186-017-0583-3

Bimatrix Games

Bimatrix Game

	Player I	Player II
Choice:	i	j
Loses:	$A_{ij} \geq 0$	$B_{ij} \geq 0$
Strategy (distribution):	x	y
Expected losses:	$x' A\, y$	$x' B\, y$

The strategy pair, $(\bar{x}, \bar{y})$ is an **equilibrium pair** if no player benefits by changing her own strategy while the other player keeps her strategy fixed:

$$\bar{x}' A \bar{y} \leq x' A \bar{y}, \qquad \forall x$$
$$\bar{x}' B \bar{y} \leq \bar{x}' B y, \qquad \forall y$$

or

$$\bar{x}' A \bar{y}\ \mathbf{1}_{m_1} \leq A \bar{y}$$
$$\bar{x}' B \bar{y}\ \mathbf{1}_{m_2} \leq \bar{x}' B$$

$$\bar{x}'A\bar{y}\ \mathbf{1}_{m_1} \leq A\bar{y}$$
$$\bar{x}'B\bar{y}\ \mathbf{1}_{m_2} \leq \bar{x}'B$$

Set,

$$\bar{\xi} = \frac{\bar{x}}{\bar{x}'B\bar{y}}, \qquad \bar{\eta} = \frac{\bar{y}}{\bar{x}'A\bar{y}}$$

Hence,

$$\mathbf{1}_{m_1} \leq A\bar{\eta}$$
$$\mathbf{1}'_{m_2} \leq \bar{\xi}'B$$

And with slack variables, $\bar{u}, \bar{v} \geq 0$ we get:

$$0 = A\bar{\eta} - \mathbf{1}_{m_1} - \bar{u}$$
$$0 = \bar{\xi}'B - \mathbf{1}'_{m_2} - \bar{v}'$$

Noting that, $\bar{u}'\bar{\xi} = 0$ and $\bar{v}'\bar{\eta} = 0$, we get the LCP:

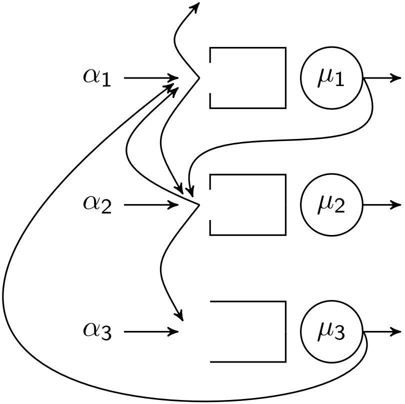
$$\begin{bmatrix} \bar{u} \\ \bar{v} \end{bmatrix} - \begin{bmatrix} 0 & A \\ B' & 0 \end{bmatrix} \begin{bmatrix} \bar{\xi} \\ \bar{\eta} \end{bmatrix} = \begin{bmatrix} -\mathbf{1}_{m_1} \\ -\mathbf{1}_{m_2} \end{bmatrix}$$

Bimatrix Game Facts

1. Basic example: "The Prisoners' Dilemma".
2. If $A + B = 0$, it is a zero-sum game.
3. ...
4. ...
5. ...
6. ...
7. ...
8. ...
9. ...
10. ...

Min-Linear Equations

Example: Queueing Networks



An illustration of an overflow fluid network with nodes $i = 1, 2, 3$: Fluid arrives exogenously at rates α_i and served at rates μ_i. Nodes 1 and 2 have finite buffer capacity, while node 3 has unbounded buffer capacity. Fluid drained out of node i is routed to node j with proportion $p_{i,j}$. Fluid reaching a node with a full buffer i overflows onto other nodes with proportion $q_{i,j}$. The remaining proportions of routing (p) or overflow (q) leave the system.

Example: Queueing Networks (cont.)

$$\lambda = \alpha + \lambda P, \quad (1)$$
$$\lambda = \alpha + \min(\lambda, \mu) P, \quad (2)$$
$$\lambda = \alpha + \min(\lambda, \mu) P + \max(\lambda - \mu, 0) Q. \quad (3)$$

$$\lambda_i = \alpha_i + \sum_{j=1}^{n} \min(\lambda_j, \mu_j) p_{j,i} + \sum_{j=1}^{n} \max(\lambda_j - \mu_j, 0) q_{j,i}, \quad i = 1, \dots, n.$$

Min Linear Equations

Data: A, b, c with non-negative elements.

Solve:

$$x = A(x \wedge b) + c, \qquad x \geq 0.$$

Or in scalar form, for $i = 1, \ldots, n$:

$$x_i = c_i + \sum_{j=1}^{n} A_{ij} \min(x_i, b_i), \qquad x_i \geq 0.$$

Let's make it an LCP.

Min Linear Equations (cont.)

$$x = A(x \wedge b) + c, \qquad x \geq 0.$$

Set $\delta = x \wedge b$. So $\delta \in [0, x]$ and $\delta \in [0, b]$. Hence,

$$(x - \delta)'(b - \delta) = 0.$$

Set $w = x - \delta$ and $z = b - \delta$. Note that $w, z \geq 0$ and $w'z = 0$.

Now use $x = A\delta + c$, to get,

$$\begin{aligned} w + \delta &= A\delta + c \\ w + (I - A)\delta &= c \\ w + (I - A)(b - z) &= c \\ w - (I - A)z &= c - (I - A)b \end{aligned}$$

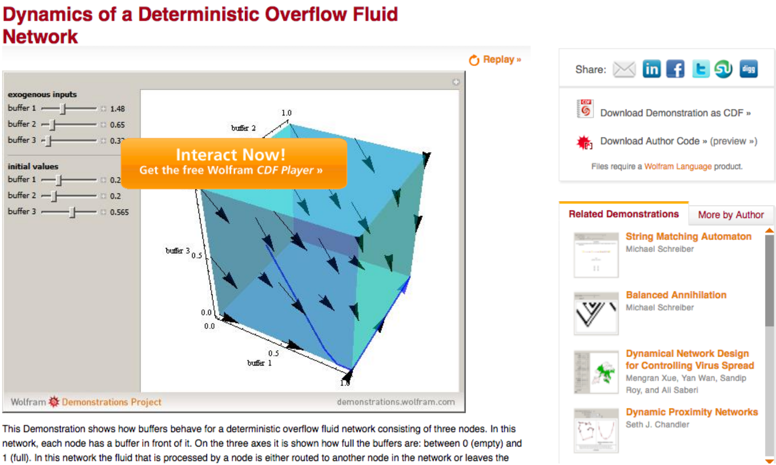
This is an LCP with $M = I - A$ and $q = c - (I - A)b$.

So what do we know about LCP?

LCP facts

1. (At least) two books dedicated to LCP:
   - *The Linear Complementarity Problem, Second Edition*, Richard W. Cottle, Jong-Shi Pang, Richard E. Stone. 1991, 2009.
   - *Linear Complementarity, Linear and Nonlinear Programming*, Katta G. Murty, 1988. Internet edition.
2. Appears in dynamic problems with boundaries - See last week's SMOR talk by David Stewart.
3. Stochastic Networks had contribution to dynamic problems: Unpublished paper (1989), Avi Mandelbaum, , *The Dynamic Complementarity Problem*
4. NP-Complete with general M matrix.
5. Has unique solution for every q if the M matrix is a "P-Matrix". If symmetric this means it is PSD.
6. ...
7. ...
8. ...
9. ...
10. ...

The End



Dynamics of a Deterministic Overflow Fluid Network
Replay »
exogenous inputs
buffer 1 1.48
buffer 2 0.65
buffer 3
initial values
buffer 1 0.2
buffer 2 0.2
buffer 3 0.565
Interact Now!
Get the free Wolfram CDF Player »
buffer 2
1.0
buffer 3
0.5
0.0
0.0
0.5
buffer 1
Wolfram Demonstrations Project
demonstrations.wolfram.com
This Demonstration shows how buffers behave for a deterministic overflow fluid network consisting of three nodes. In this network, each node has a buffer in front of it. On the three axes it is shown how full the buffers are: between 0 (empty) and 1 (full). In this network the fluid that is processed by a node is either routed to another node in the network or leaves the
Share:
Download Demonstration as CDF »
Download Author Code » (preview »)
Files require a Wolfram Language product.
Related Demonstrations
More by Author
String Matching Automaton
Michael Schreiber
Balanced Annihilation
Michael Schreiber
Dynamical Network Design for Controlling Virus Spread
Mengran Xue, Yan Wan, Sandip Roy, and Ali Saberi
Dynamic Proximity Networks
Seth J. Chandler