IE (598ns)
September 17, 2008
Nedić & Shanbhag

Homework 2
Due September 29, 2008

Exercise 1. Let $K \subseteq \mathbb{R}^n$ be a nonempty closed convex set, and let $\Pi_K(x)$ denote the projection of a vector $x \in \mathbb{R}^n$ on the set K. Show the following properties of the projection:

(a) For any $x \in \mathbb{R}^n$, the projection $\Pi_K(x)$ exists and is unique.

(b) For each $x \in \mathbb{R}^n$, the projection $\Pi_K(x)$ is the unique vector $\bar{x} \in K$ satisfying the following inequality

$$(y - \bar{x})^T(\bar{x} - x) \geq 0 \qquad \text{for all } y \in K.$$

(c) For any two vectors $u, v \in \mathbb{R}^n$, we have

$$\left(\Pi_K(u) - \Pi_k(v)\right)^T (u - v) \geq \|\Pi_K(u) - \Pi_K(v)\|^2.$$

(d) The mapping $\Pi_K(x)$ is non-expansive, i.e.,

$$\|\Pi_K(u) - \Pi_K(v)\| \leq \|u - v\| \qquad \text{for all } u, v \in \mathbb{R}^n.$$

(e) The squared distance function

$$\rho(x) = \frac{1}{2}\, \|x - \Pi_K(x)\|^2 \qquad \text{for } x \in \mathbb{R}^n,$$

is continuously differentiable in x, and its gradient is given by

$$\nabla\rho(x) = x - \Pi_K(x) \qquad \text{for } x \in \mathbb{R}^n.$$

Exercise 2. Let $K \subseteq \mathbb{R}^n$ be a nonempty closed convex set, and let $F : K \to \mathbb{R}^n$ be an arbitrary mapping. The map $F_K^{\text{nor}}(x)$ defined by

$$F_K^{\text{nor}}(x) = F(\Pi_K(x)) + x - \Pi_K(x) \qquad \text{for } x \in \mathbb{R}^n$$

is referred to as the **normal map**. Show that a vector x^* belongs to $SOL(K, F)$ if and only if there is a vector z such that $x^* = \Pi_K(z)$ and $F_K^{\text{nor}}(z) = 0$.

Exercise 3. Let $K \subseteq \mathbb{R}^n$ be closed and convex and $F : K \to \mathbb{R}^n$ be a continuous mapping. If there exists a vector $x^{\text{ref}} \in K$ and a scalar $\zeta \geq 0$ such that

$$\liminf_{x \in K, \|x\| \to \infty} \frac{F(x)^T(x - x^{\text{ref}})}{\|x\|^{\zeta}} > 0$$

holds, then the VI(K,F) has a nonempty compact solution set.

The following exercises of Facchinei and Pang's book, vol. I:
1.8.10, 1.8.11, 1.8.13, 2.9.10 part (ii), 2.9.11, and 2.9.16.

Exercise 4. Computational exercise: Consider the Nash-Cournot game that was described earlier in the course. Formulate the game as a complementarity problem with appropriately chosen parameters and solve it using Tomlab/Matlab.

The following code is from `/usr/local/bin/tomlab/examples/mcptest1.m` with the relevant other files being housed in the same directory. You may want to modify this code with your own formulation.

Some questions of interest when solving the problem:

1. How does the algorithm scale with the number of users?
2. What is the local behavior of the method?
3. What is the global behavior of the method (change starting points)?

Please attach only the output but you need NOT attach any code.

```
% mcptest1
% Transportation model as a variational inequality in MATLAB with isoelstic
% demand.
%
% lcptest1 is similar, see problem formualtion in this file.
%
% 0 <= x_k _|_ F(x) >= 0  (_|_ means complements)
%
% s.t b_L <= A <= b_U

clear all

MPEC = [1 0 0 0 1 0; ...
```

```
        2 0 0 0 2 0;
        3 0 0 0 3 0;
        4 0 0 0 4 0;
        5 0 0 0 5 0;
        6 0 0 0 6 0;
        7 0 0 0 7 0;
        8 0 0 0 8 0;
        9 0 0 0 9 0;
        10 0 0 0 10 0;
        11 0 0 0 11 0];

d2c = [];
ConsPattern = [];

c_L = zeros(11,1);
c_U = inf*ones(11,1);

Prob = mcpAssign([], [], [], [], [], [], 'MCP 1', ones(11,1), ...
                 MPEC, [], ...
                 [], [], [], 'mcptest1_c', 'mcptest1_dc', d2c,...
     ConsPattern, c_L, c_U);

Prob.KNITRO.options.ALG = 1;
R = tomRun('knitro', Prob, 1);

% function Prob = mcpAssign(F, J, JacPattern, x_L, x_U, Name, x_0, ...
%                           A, b_L, b_U, x_min, x_max, f_opt, x_opt);

% Prob = mcpAssign('mcptest1_f', [], [], [], [], 'MCP 1', ones(11,1));
% Prob.PriLevOpt = 1;
%
% Result = tomRun('path',Prob,1);
% pause

% ---------------------------------------------------------------------
% Also give the Jacobian explicitly.
%
% Prob2 = mcpAssign('mcptest1_f', 'mcptest1_J', [], [], [], 'MCP 2', ones(11,1));
```

```
% Prob2.PriLevOpt = 1;
%
% Result2 = tomRun('path',Prob2,1);
% pause

% ---------------------------------------------------------------------
% Test first problem with MAD

% Prob.ADObj = 1;
% madinitglobals;
% Result3 = tomRun('path',Prob,1);

% Approximate times:
% 1. Only f given:        0.25 s
% 2. Both f anf J given:  0.03 s
% 3. Only f with MAD:     0.11 s

% Adding a linear constraint.

% A = [0 0 0 0 0 0 0 0 0 1 1];
% b_U = 2;
%
% Prob3 = mcpAssign('mcptest1_f', 'mcptest1_J', [], [], [], 'MCP 3',...
ones(11,1), A, [], b_U);
% Prob3.PriLevOpt = 1;
%
% Result3 = tomRun('path',Prob3,1);
```