ACM 113 Introduction to Optimization - Problem Set 5

Ling Li, ling@cs.caltech.edu

May 21, 2001

5.1 From [N&S, p.136], both basis $\{x_1, x_2, x_7\}$ and $\{x_1, x_6, x_7\}$ correspond to $x_i = 0$ $(i \neq 7)$ and $x_7 = 1$. Thus this problem is degenerate. Using lexicographic perturbation, we have

basic	$\overset{\Downarrow}{x_1}$	x_2	x_3	x_4	x_5	x_6	x_7	rhs
$-z$	$-\frac{3}{4}$	150	$-\frac{1}{50}$	6	0	0	0	0
x_5	$\frac{1}{4}$	-60	$-\frac{1}{25}$	9	1	0	0	ϵ_0
x_6	$\boxed{\frac{1}{2}}$	-90	$-\frac{1}{50}$	3	0	1	0	ϵ_0^2
x_7	0	0	1	0	0	0	1	$1+\epsilon_0^3$

basic	x_1	x_2	$\overset{\Downarrow}{x_3}$	x_4	x_5	x_6	x_7	rhs
$-z$	0	15	$-\frac{1}{20}$	$\frac{21}{2}$	0	$\frac{3}{2}$	0	$\frac{3}{2}\epsilon_0^2$
x_5	0	-15	$-\frac{3}{100}$	$\frac{15}{2}$	1	$-\frac{1}{2}$	0	$\epsilon_0 - \frac{1}{2}\epsilon_0^2$
x_1	1	-180	$-\frac{1}{25}$	6	0	2	0	$2\epsilon_0^2$
x_7	0	0	$\boxed{1}$	0	0	0	1	$1+\epsilon_0^3$

basic	x_1	x_2	x_3	x_4	x_5	x_6	x_7	rhs
$-z$	0	15	0	$\frac{21}{2}$	0	$\frac{3}{2}$	$\frac{1}{20}$	$\frac{1}{20} + \frac{3}{2}\epsilon_0^2 + \frac{1}{20}\epsilon_0^3$
x_5	0	-15	0	$\frac{15}{2}$	1	$-\frac{1}{2}$	$\frac{3}{100}$	$\frac{3}{100} + \epsilon_0 - \frac{1}{2}\epsilon_0^2 + \frac{3}{100}\epsilon_0^3$
x_1	1	-180	0	6	0	2	$\frac{1}{25}$	$\frac{1}{25} + 2\epsilon_0^2 + \frac{1}{25}\epsilon_0^3$
x_3	0	0	1	0	0	0	1	$1+\epsilon_0^3$

The cycling doesn't occur, and the optimal feasible point we get is $(\frac{1}{25}, 0, 1, 0, \frac{3}{100}, 0)^T$, with objective $-\frac{1}{20}$.

5.2 [N&S, p.166-167] gives the optimal solution $x_B = (x_2, x_1, x_3)^T = (5, 3, 3)^T$.

$$N = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix},\ B^{-1} = \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 1 \\ 1 & -\frac{1}{2} & \frac{3}{2} \end{pmatrix},\ c_B = \begin{pmatrix} -2 \\ -1 \\ 0 \end{pmatrix}.$$

(a) Denote the change of b by $\Delta b = (\delta, 0, 0)^T$. This change does not affect the optimality conditions. As long as the feasibility conditions

$$B^{-1}(b + \Delta b) = x_B + B^{-1}\Delta b \geq 0 \tag{1}$$

remain satisfied, the current basis is still optimal. (1) is

$$B^{-1}\Delta b = \begin{pmatrix} 0 \\ 0 \\ \delta \end{pmatrix} \geq \begin{pmatrix} -5 \\ -5 \\ -3 \end{pmatrix},$$

i.e., $\delta \geq -3$. So b_1 can be decreased by at most 3 and can be increased by any (positive) value.

(b) Since

$$\bar{x}_B = B^{-1}\left(b + (0,0,5)^T\right) = x_B + \left(\frac{5}{2}, 5, \frac{15}{2}\right)^T = \left(\frac{15}{2}, 8, \frac{21}{2}\right)^T > 0,$$

increasing b_3 by 5 doesn't change the optimal basis. And the new solution is $(x_1, x_2, x_3, x_4)^T = \left(8, \frac{15}{2}, \frac{21}{2}, 0\right)^T$, with objective $c_B^T \bar{x}_B = -23$.

(c) Denote the change of c_B by $\Delta c_B = (0, \delta, 0)^T$. The change of c_B only affects the optimal conditions.

$$c_N^T - (c_B + \Delta c_B)^T B^{-1} N = \hat{c}_N^T - \Delta c_B^T B^{-1} N = (1,2) - (0,\delta) \geq 0.$$

Thus $\delta \leq 2$. So if c_1 is increased or decreased by 2, the solution doesn't change. However, the optimal objective changes by $\Delta c_B^T x_B = 3\delta$.

5.3 Since $s^T x = \sum_{i=1}^n x_i s_i = n\tau$, we have

$$c^T x = (A^T y + s)^T x = y^T (Ax) + s^T x = y^T b + n\tau,$$

i.e., $c^T x - b^T y = n\tau$.

5.4 During the iterates of the primal-dual Newton step, $A^T \Delta y + \Delta s = 0$ and $A\Delta x = 0$. Thus

$$\Delta x^T \Delta s = \Delta x^T (-A^T \Delta y) = -(A\Delta x)^T \Delta y = 0.$$

5.5 I programmed the primal-dual predictor-corrector method in Matlab.

(a) After 10 iterations, we get

$$\begin{aligned} x &= \left(0.65264, 1.1737, 2.1316, 5.3053, 2.257 \times 10^{-17}\right)^T, \\ s &= \left(4.3299 \times 10^{-17}, 1.0776 \times 10^{-16}, 1.3619 \times 10^{-17}, 3.49 \times 10^{-17}, 1\right)^T, \\ y &= \left(3.4001 \times 10^{-18}, -1.7881 \times 10^{-17}, -1\right)^T. \end{aligned}$$

The optimized objective is $c^T x = b^T y = -3$.

(b) The command used to verify my answer on larger LPs is

```
x = linprog(c, -eye(n), zeros(n,1), A, b);
```

- For $m = 100$, $n = 150$, my program typically took about 1.1 seconds ($16 \sim 17$ iterations) to achieve $n\mu_k < 10^{-15}$. And the `linprog` usually took about 4.5 seconds to get the same result.
- For $m = 500$, $n = 650$, my program took about 40 seconds (around 20 iterations) to achieve $n\mu_k < 10^{-15}$. The `linprog` usually can not get the result.