Introduction to Mathematical Programming
IE406

Lecture 14

Dr. Ted Ralphs

Reading for This Lecture

- Bertsimas Chapter 5

AMPL: Displaying Auxiliary Values with Suffixes

- In AMPL, it's possible to display many of the auxiliary values we've been looking at using suffixes.
- For example, to display the reduced cost of a variable, type the variable name with the suffix *.rc*.
- Using the product mix example (`prod.mod` and `prod1.dat`),

```
ampl: display make;
make [*] :=
gadgets  15000
widgets  20000
;

ampl: display make.rc;
make.rc [*] :=
gadgets  0
widgets  0
;
```

AMPL: Other Auxiliary Information

- You can display the *status* of each variable

```
ampl: display make.sstatus;
make.sstatus [*] :=
gadgets  bas
widgets  bas
```

- You can also display such things as the slack in the constraints

```
ampl: display hours_limit.slack;
hours_limit.slack = 10000
```

- Or the status of a slack variable

```
ampl: display hours_limit.status;
hours_limit.status = bas
```

- A list of all the possible suffixes is on the AMPL Web site.

AMPL: Sensitivity

- AMPL does not have built-in sensitivity analysis commands.
- AMPL/CPLEX does provide such capability.
- To get sensitivity information, type the following

```
ampl: option cplex_options 'sensitivity';
```

- Solve the model from product mix model:

```
ampl: solve;
CPLEX 7.0.0: sensitivity

CPLEX 7.0.0: optimal solution; objective 105000
2 simplex iterations (0 in phase I)

suffix up OUT;
suffix down OUT;
suffix current OUT;
```

AMPL: Accessing Sensitivity Information

Access sensitivity information using the suffixes *.up* and *.down*.

```
ampl: display hours_limit.up;
hours_limit.up = 180000

ampl: let max_prd := 190000;
ampl: solve;
CPLEX 7.0.0: sensitivity

CPLEX 7.0.0: optimal solution; objective 135000
1 simplex iterations (1 in phase I)
ampl: display make;
make [*] :=
gadgets  45000
widgets      0
;
```

AMPL: Accessing Sensitivity Information (cont.)

```
ampl: display make.rc;
make.rc [*] :=
gadgets   0
widgets  -1.5
;

ampl: display make.sstatus;
make.sstatus [*] :=
gadgets  bas
widgets  low
;
```

A Case Study in Sensitivity Analysis

- This study is from Section 1.2 and 5.1 of the book.
- The following new computer models are being introduced by DEC.

System	Price	# disk drives	# 256K boards
GP-1	\$60,000	0.3	4
GP-2	\$40,000	1.7	2
GP-3	\$30,000	0	2
WS-1	\$30,000	1.4	2
WS-2	\$15,000	0	1

Scenario

- The following difficulties were anticipated:
  - The in-house CPU supply is limited to 7,000 units.
  - The supply of disk drives is uncertain (3,000-7,000).
  - The supply of memory boards is also limited (8,000-16,000).
- Some maximum and minimum demand information is known.
- Possible Approaches
  - To address the disk drive shortage, GP-1 systems could be produced with no disk drives.
  - In addition, the GP-1 systems could be built using two alternative memory boards instead of four standard boards.
  - 4,000 alternative boards are available.

Questions to Be Answered

- Should the GP-1s be manufactured with or without disk drives?
- Should the alternative memory boards be used?
- The small staff can only devote their efforts to either finding an alternative source of disk drives or memory boards. Which one?
- We can answer these questions using sensitivity analysis.

Analysis

Alt. Boards	Drives	Revenue	GP-1	GP-2	GP-3	WS-1	WS-2
no	no	145	0	2.5	0	0.5	2
yes	no	248	1.8	2	0	1	2
no	yes	133	0.272	1.304	0.3	0.5	2.7
yes	yes	213	1.8	1.035	0.3	0.5	2.7

Mode	No Disk Drives	Disk Drives
Revenue	248	213
Shadow price for boards	15	0
Range for boards	$[-1.5, 2]$	$[-1.62, \infty]$
Shadow price for drives	0	23.52
Range for drives	$[-0.2, 0.75]$	$[-0.91, 1.13]$

Global Dependence on the Right-hand Side Vector

- Consider a family of polyhedra parameterized by the vector b

$$\mathcal{P}(b) = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$$

- Note that

$$S = \{b : \mathcal{P}(b) \text{ is nonempty}\} = \{Ax : x \geq 0\}$$

is a convex set.

- We now consider the function $F(b) = \min_{x \in \mathcal{P}(b)} c^\top x$.
- In what follows, we will assume feasibility of the dual and hence that $F(b)$ is finite for all $b \in S$.
- We will try to characterize the function $F(b)$.

Characterizing F(b)

- For a particular vector $\hat{b}$, suppose there is a nondenegenerate optimal basic feasible solution given by basis B.
- As before, nondegeneracy implies that we can perturb $\hat{b}$ without changing the optimal basis.
- Therefore, we have

$$F(b) = c_B^\top B^{-1} b = p^\top b, \quad \text{for } b \text{ ``close to'' } \hat{b}.$$

- This means that in the vicinity of $\hat{b}$, $F(b)$ is a linear function of b.
- Consider the extreme points $p^1, \ldots, p^N$ of the dual polyhedron.
- There must be an extremal optimum to the dual and so we can rewrite $F(b)$ as

$$F(b) = \max_{i=1,\ldots,N} (p^i)^\top b$$

- Hence, $F(b)$ is a piecewise linear convex function.

Another Parameterization

- Now consider the function $f(\theta) = F(\hat{b} + \theta d)$ for a particular vector $\hat{b}$ and direction d.
- Using the same approach, we obtain

$$f(\theta) = \max_{i=1,\ldots,N} (p^i)^\top (\hat{b} + \theta d), \quad \hat{b} + \theta d \in S.$$

- Again, this is a piecewise linear convex function.

The Set of all Dual Optimal Solutions

Consider once more the function $F(b)$.

Definition 1. *A vector $p \in \mathbb{R}^m$ is a* subgradient *of F at $\hat{b}$ if $F(\hat{b}) + p^\top (b - \hat{b}) \leq F(b)$.*

Theorem 1. *Suppose that the linear program $\min\{c^\top x : Ax = \hat{b}, x \geq 0\}$ is feasible and that the optimal cost is finite. Then p is an optimal solution to the dual if and only if it is a subgradient of F at $\hat{b}$.*

Global Dependence on the Cost Vector

- We have similar results for dependence on the cost vector.
- As before, assume primal feasibility and define the dual feasible set by

$$Q(c) = \{p : p^\top A \leq c\}$$

and set $T = \{c : Q(c) \text{ is nonempty}\}$.

- It is easy to show that T is again a convex set.
- Defining a function $G(c)$ to represent the optimal primal cost corresponding to vector c, we see as before that

$$G(c) = \min_{i=1,\ldots,N} c^\top x^i$$

where $x^1, \ldots, x^N$ are the extreme points of the primal polyhedron.

Characterizing G(c)

- As before, $G(c)$ is a piecewise linear concave function.
- Furthermore, if for some cost vector $\hat{c}$, there is unique optimal solution x^*, then $G(c)$ is linear in the vicinity of $\hat{c}$.
- The set of all optimal solutions is again given by the subgradients of $G(c)$.

Parametric Programming

- For a fixed matrix A, vectors b and c, and direction d and consider the problem

$$\begin{aligned} min\ & (c+\theta d)^\top x \\ s.t.\ & Ax = b \\ & x \geq 0 \end{aligned}$$

- If $g(\theta)$ is the optimal cost for a given θ, then as before

$$g(\theta) = \min_{i=1,\ldots,N} (c+\theta d)^\top x^i$$

where $x^1, \ldots, x^N$ are the extreme points of the feasible set.

- Hence, $g(\theta)$ is piecewise linear and concave.
- We now derive a systematic method for obtaining $g(\theta)$ for all θ.

The Basic Idea

- For a fixed basis B and a fixed θ, the reduced costs are linear functions of θ.
- Hence, we can easily determine the range of values of θ for which the current basis is optimal.
- The ends of this interval determine the nearest breakpoints of $g(\theta)$.
- By changing θ to the next breakpoint and performing a change of basis, we obtain a new interval and a new breakpoint.
- This process can be continued to obtain all breakpoints.

The Parametric Simplex Method

- Determine an initial feasible basis.
- Determine the interval $[\theta_1, \theta_2]$ for which this basis is optimal.
- Determine a variable j whose reduced cost is nonpositive for $\theta \geq \theta_2$.
- If the corresponding column has no positive entries, then the problem is unbounded for $\theta > \theta_2$.
- Otherwise, rotate column j into the basis.
- Determine a new interval $[\theta_2, \theta_3]$ in which the current basis is optimal.
- Iterate to find all breakpoint $\geq \theta_1$.
- Repeat the process to find breakpoints $\leq \theta_1$.

Comments on Parametric Simplex Method

- Note that as long as each interval has a positive length, this process will terminate finitely.
- If some interval has zero length, cycling is possible.
- We can prevent cycling with anticycling rules as in the regular simplex algorithm.
- We can perform the same analysis for a parametrically defined right hand side using dual simplex.