CHAPTER 7

Sensitivity and Parametric Analyses

In this chapter, we consider two related subjects. The first, called sensitivity analysis (or postoptimality analysis) addresses the following question: having found an optimal solution to a given linear programming problem, how much can we change the data and have the current partition into basic and nonbasic variables remain optimal? The second subject addresses situations in which one wishes to solve not just one linear program, but a whole family of problems parametrized by a single real variable.

We shall study parametric analysis in a very specific context in which we wish to find the optimal solution to a given linear programming problem by starting from a problem whose solution is trivially known and then deforming this problem back to the original problem, maintaining as we go optimality of the current solution. The result of this deformation approach to solving a linear programming problem is a new variant of the simplex method, which is called the parametric self-dual simplex method. We will see in later chapters that this variant of the simplex method resembles, in certain respects, the interior-point methods that we shall study.

1. Sensitivity Analysis

One often needs to solve not just one linear programming problem but several closely related problems. There are many reasons that this need might arise. For example, the data that define the problem may have been rather uncertain and one may wish to consider various possible data scenarios. Or perhaps the data are known accurately but change from day to day, and the problem must be resolved for each new day. Whatever the reason, this situation is quite common. So one is led to ask whether it is possible to exploit the knowledge of a previously obtained optimal solution to obtain more quickly the optimal solution to the problem at hand. Of course, the answer is often yes, and this is the subject of this section.

We shall treat a number of possible situations. All of them assume that a problem has been solved to optimality. This means that we have at our disposal the final, optimal dictionary:

$$\begin{aligned} \zeta &= \zeta^* - z_{\mathcal{N}}^{*T} x_{\mathcal{N}} \\ x_{\mathcal{B}} &= x_{\mathcal{B}}^* - B^{-1} N x_{\mathcal{N}}. \end{aligned}$$

Suppose we wish to change the objective coefficients from c to, say, $\tilde{c}$. It is natural to ask how the dictionary at hand could be adjusted to become a valid dictionary for the new problem. That is, we want to maintain the current classification of the variables

R.J. Vanderbei, *Linear Programming*, International Series in Operations Research & Management Science 196, DOI 10.1007/978-1-4614-7630-6_7,

into basic and nonbasic variables and simply adjust ζ^*, $z_{\mathcal{N}}^*$, and $x_{\mathcal{B}}^*$ appropriately. Recall from (6.7) to (6.9) that

$$\begin{aligned} x_{\mathcal{B}}^* &= B^{-1}b, \\ z_{\mathcal{N}}^* &= (B^{-1}N)^T c_{\mathcal{B}} - c_{\mathcal{N}}, \\ \zeta^* &= c_{\mathcal{B}}^T B^{-1} b. \end{aligned}$$

Hence, the change from c to $\tilde{c}$ requires us to recompute $z_{\mathcal{N}}^*$ and ζ^*, but $x_{\mathcal{B}}^*$ remains unchanged. Therefore, after recomputing $z_{\mathcal{N}}^*$ and ζ^*, the new dictionary is still primal feasible, and so there is no need for a Phase I procedure: we can jump straight into the primal simplex method, and if $\tilde{c}$ is not too different from c, we can expect to get to the new optimal solution in a relatively small number of steps.

Now suppose that instead of changing c, we wish to change only the right-hand side b. In this case, we see that we need to recompute $x_{\mathcal{B}}^*$ and ζ^*, but $z_{\mathcal{N}}^*$ remains unchanged. Hence, the new dictionary will be dual feasible, and so we can apply the dual simplex method to arrive at the new optimal solution fairly directly.

Therefore, changing just the objective function or just the right-hand side results in a new dictionary having nice feasibility properties. What if we need/want to change some (or all) entries in both the objective function and the right-hand side and maybe even the constraint matrix too? In this case, everything changes: ζ^*, $z_{\mathcal{N}}^*$, $x_{\mathcal{B}}^*$. Even the entries in B and N change. Nonetheless, as long as the new basis matrix B is nonsingular, we can make a new dictionary that preserves the old classification into basic and nonbasic variables. The new dictionary will most likely be neither primal feasible nor dual feasible, but if the changes in the data are fairly small in magnitude, one would still expect that this starting dictionary will get us to an optimal solution in fewer iterations than simply starting from scratch. While there is no guarantee that any of these so-called warm-starts will end up in fewer iterations to optimality, extensive empirical evidence indicates that this procedure often makes a substantial improvement: sometimes the warm-started problems solve in as little as 1 % of the time it takes to solve the original problem.

1.1. Ranging. Often one does not wish to solve a modification of the original problem, but instead just wants to ask a hypothetical question:

> *If I were to change the objective function by increasing or decreasing one of the objective coefficients a small amount, how much could I increase/decrease it without changing the optimality of my current basis?*

To study this question, let us suppose that c gets changed to $c + t\Delta c$, where t is a real number and Δc is a given vector (which is often all zeros except for a one in a single entry, but we don't need to restrict the discussion to this case). It is easy to see that $z_{\mathcal{N}}^*$ gets incremented by

$$t\Delta z_{\mathcal{N}},$$

where

$$\Delta z_{\mathcal{N}} = (B^{-1}N)^T \Delta c_{\mathcal{B}} - \Delta c_{\mathcal{N}}. \tag{7.1}$$

Hence, the current basis will remain dual feasible as long as

$$z^*_{\mathcal{N}} + t\Delta z_{\mathcal{N}} \geq 0. \tag{7.2}$$

We've manipulated this type of inequality many times before, and so it should be clear that, for $t > 0$, this inequality will remain valid as long as

$$t \leq \left(\max_{j \in \mathcal{N}} -\frac{\Delta z_j}{z^*_j} \right)^{-1}.$$

Similar manipulations show that, for $t < 0$, the lower bound is

$$t \geq \left(\min_{j \in \mathcal{N}} -\frac{\Delta z_j}{z^*_j} \right)^{-1}.$$

Combining these two inequalities, we see that t must lie in the interval

$$\left(\min_{j \in \mathcal{N}} -\frac{\Delta z_j}{z^*_j} \right)^{-1} \leq t \leq \left(\max_{j \in \mathcal{N}} -\frac{\Delta z_j}{z^*_j} \right)^{-1}.$$

Let us illustrate these calculations with an example. Consider the following linear programming problem:

$$\begin{array}{lrcrcrcr}
\text{maximize} & 5x_1 + 4x_2 & + & 3x_3 & & \\
\text{subject to} & 2x_1 + 3x_2 & + & x_3 & \leq & 5 \\
 & 4x_1 + x_2 & + & 2x_3 & \leq & 11 \\
 & 3x_1 + 4x_2 & + & 2x_3 & \leq & 8 \\
 & x_1, \; x_2, \; x_3 & \geq & 0 & . &
\end{array}$$

The optimal dictionary for this problem is given by

$$\begin{array}{rcl}
\xi & = & 13 - 3x_2 - x_4 - x_6 \\
\hline
x_3 & = & 1 + x_2 + 3x_4 - 2x_6 \\
x_1 & = & 2 - 2x_2 - 2x_4 + x_6 \\
x_5 & = & 1 + 5x_2 + 2x_4 \; .
\end{array}$$

The optimal basis is $\mathcal{B} = \{3, \; 1, \; 5\}$. Suppose we want to know how much the coefficient of 5 on x_1 in the objective function can change without altering the optimality of this basis. From the statement of the problem, we see that

$$c = \begin{bmatrix} 5 & 4 & 3 & 0 & 0 & 0 \end{bmatrix}^T.$$

Since we are interested in changes in the first coefficient, we put

$$\Delta c = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T.$$

We partition c according to the final (optimal) basis. Hence, we have

$$\Delta c_{\mathcal{B}} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \qquad \text{and} \qquad \Delta c_{\mathcal{N}} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Next, we need to compute $\Delta z_{\mathcal{N}}$ using (7.1). We could compute $B^{-1}N$ from scratch, but it is easier to extract it from the constraint coefficients in the final dictionary. Indeed,

$$-B^{-1}N = \begin{bmatrix} 1 & 3 & -2 \\ -2 & -2 & 1 \\ 5 & 2 & 0 \end{bmatrix}.$$

Hence, from (7.1) we see that

$$\Delta z_{\mathcal{N}} = \begin{bmatrix} 2 \\ 2 \\ -1 \end{bmatrix}.$$

Now, (7.2) gives the condition on t. Writing it out componentwise, we get

$$3 + 2t \geq 0, \qquad 1 + 2t \geq 0, \qquad \text{and} \qquad 1 - t \geq 0.$$

These three inequalities, which must all hold, can be summarized by saying that

$$-\frac{1}{2} \leq t \leq 1$$

Hence, in terms of the coefficient on x_1, we finally see that it can range from 4.5 to 6.

Now suppose we change b to $b + t\Delta b$ and ask how much t can change before the current basis becomes nonoptimal. In this case, $z^*_{\mathcal{N}}$ does not change, but $x^*_{\mathcal{B}}$ gets incremented by $t\Delta x_{\mathcal{B}}$, where

$$\Delta x_{\mathcal{B}} = B^{-1}\Delta b.$$

Hence, the current basis will remain optimal as long as t lies in the interval

$$\left(\min_{i \in \mathcal{B}} -\frac{\Delta x_i}{x^*_i} \right)^{-1} \leq t \leq \left(\max_{i \in \mathcal{B}} -\frac{\Delta x_i}{x^*_i} \right)^{-1}.$$

2. Parametric Analysis and the Homotopy Method

In this section, we illustrate the notion of *parametric analysis* by applying a technique called the *homotopy method* to get a new algorithm for solving linear programming problems. The homotopy method is a general technique in which one creates a continuous deformation that changes a given difficult problem into a related but trivially solved problem and then attempts to work backwards from the trivial problem to the difficult problem by solving (hopefully without too much effort) all the problems in between. Of course, there is a continuum of problems between the hard one and the trivial one, and so we shouldn't expect that this technique will be effective in every situation; but for linear programming and for many other problem domains, it turns out to yield efficient algorithms.

We start with an example. Suppose we wish to solve the following linear programming problem:

$$\begin{array}{lrcr} \text{maximize} & -2x_1 + 3x_2 & & \\ \text{subject to} & -x_1 + \ x_2 & \leq & -1 \\ & -x_1 - 2x_2 & \leq & -2 \\ & x_2 & \leq & 1 \\ & x_1,\ x_2 & \geq & 0\ . \end{array}$$

The starting dictionary is

$$\begin{array}{rl} \zeta = & \quad - 2x_1 - (-3)x_2 \\ \hline x_3 = & -1 + \ x_1 - \quad\ x_2 \\ x_4 = & -2 + \ x_1 + \quad 2x_2 \\ x_5 = & \ \ 1 \qquad\ - \quad\ x_2\ . \end{array}$$

This dictionary is neither primal nor dual feasible. Let's perturb it by adding a positive real number μ to each right-hand side and subtracting it from each objective function coefficient. We now arrive at a family of dictionaries, parametrized by μ:

$$\begin{array}{rl} \zeta = & \qquad\qquad - (2+\mu)x_1 - (-3+\mu)x_2 \\ \hline x_3 = & -1 + \mu + \qquad x_1 - \qquad\quad x_2 \\ x_4 = & -2 + \mu + \qquad x_1 + \qquad\ 2x_2 \\ x_5 = & \ \ 1 + \mu \qquad\qquad\ - \qquad\quad x_2\ . \end{array} \tag{7.3}$$

Clearly, for μ sufficiently large, specifically $\mu \geq 3$, this dictionary is both primal and dual feasible. Hence, the associated solution $x = [0, 0, -1+\mu, -2+\mu, 1+\mu]$ is optimal. Starting with μ large, we reduce it as much as we can while keeping dictionary (7.3) optimal. This dictionary will become nonoptimal as soon as $\mu < 3$, since the associated dual variable $y_2^* = -3 + \mu$ will become negative. In other words, the coefficient of x_2, which is $3 - \mu$, will become positive. This change of sign on the coefficient of x_2 suggests that we make a primal pivot in which x_2 enters the basis. The usual ratio test (using the specific value of $\mu = 3$) indicates that x_3 must be the leaving variable. Making the pivot, we get

$$\begin{array}{rl} \zeta = & -3 + 4\mu - \mu^2 - (-1+2\mu)x_1 - (3-\mu)x_3 \\ \hline x_2 = & -1 + \ \mu \qquad + \qquad\quad x_1 - \qquad\quad x_3 \\ x_4 = & -4 + 3\mu \qquad + \qquad 3x_1 - \qquad 2x_3 \\ x_5 = & \ \ 2 \qquad\qquad - \qquad\quad x_1 + \qquad\quad x_3. \end{array}$$

This dictionary is optimal as long as

$$\begin{array}{rr} -1 + 2\mu \geq 0, & 3 - \mu \geq 0, \\ -1 + \mu \geq 0, & -4 + 3\mu \geq 0. \end{array}$$

These inequalities reduce to

$$\frac{4}{3} \leq \mu \leq 3.$$

So now we can reduce μ from its current value of 3 down to $4/3$. If we reduce it below $4/3$, the primal feasibility inequality, $-4 + 3\mu \geq 0$, becomes violated. This violation suggests that we perform a dual pivot with x_4 serving as the leaving variable. The usual (dual) ratio test (with $\mu = 4/3$) then tells us that x_1 must be the entering variable. Doing the pivot, we get

$$\begin{array}{rlcll}
\zeta = & -\frac{5}{3} + \frac{1}{3}\mu + \mu^2 & - & (-\frac{1}{3} + \frac{2}{3}\mu)x_4 & - (\frac{7}{3} + \frac{1}{3}\mu)x_3 \\
\hline
x_2 = & \frac{1}{3} & + & \frac{1}{3}x_4 & - \frac{1}{3}x_3 \\
x_1 = & \frac{4}{3} - \mu & + & \frac{1}{3}x_4 & + \frac{2}{3}x_3 \\
x_5 = & \frac{2}{3} + \mu & - & \frac{1}{3}x_4 & + \frac{1}{3}x_3 .
\end{array}$$

Now the conditions for optimality are

$$\begin{aligned} -\frac{1}{3} + \frac{2}{3}\mu &\geq 0, & \frac{7}{3} + \frac{1}{3}\mu &\geq 0, \\ \frac{4}{3} - \mu &\geq 0, & \frac{2}{3} + \mu &\geq 0, \end{aligned}$$

which reduce to

$$\frac{1}{2} \leq \mu \leq \frac{4}{3}.$$

For the next iteration, we reduce μ to $1/2$ and see that the inequality that becomes binding is the dual feasibility inequality

$$-\frac{1}{3} + \frac{2}{3}\mu \geq 0.$$

Hence, we do a primal pivot with x_4 entering the basis. The leaving variable is x_5, and the new dictionary is

$$\begin{array}{rlcll}
\zeta = & -1 - \mu^2 & - & (1 - 2\mu)x_5 & - (2 + \mu)x_3 \\
\hline
x_2 = & 1 + \mu & - & x_5 & \\
x_1 = & 2 & - & x_5 & + x_3 \\
x_4 = & 2 + 3\mu & - & 3x_5 & + x_3 .
\end{array}$$

For this dictionary, the range of optimality is given by

$$\begin{aligned} 1 - 2\mu &\geq 0, & 2 + \mu &\geq 0, \\ 1 + \mu &\geq 0, & 2 + 3\mu &\geq 0, \end{aligned}$$

which reduces to

$$-\frac{2}{3} \leq \mu \leq \frac{1}{2}.$$

This range covers $\mu = 0$, and so now we can set μ to 0 and get an optimal dictionary for our original problem:

$$\begin{array}{rl}
\zeta = & -1 - x_5 - 2x_3 \\
\hline
x_2 = & 1 - x_5 \\
x_1 = & 2 - x_5 + x_3 \\
x_4 = & 2 - 3x_5 + x_3 .
\end{array}$$

The algorithm we have just illustrated is called the *parametric self-dual simplex method*.[1] We shall often refer to it more simply as the self-dual simplex method. It has some attractive features. First, in contrast to the methods presented earlier, this

[1] In the first edition, this method was called the primal–dual simplex method.

algorithm does not require a separate Phase I procedure. It starts with any problem, be it primal infeasible, dual infeasible, or both, and it systematically performs pivots (whether primal or dual) until it finds an optimal solution.

A second feature is that a trivial modification of the algorithm can avoid entirely ever encountering a degenerate dictionary. Indeed, suppose that, instead of adding/subtracting μ from each of the right-hand sides and objective coefficients, we add/subtract a positive constant times μ. Suppose further that the positive constant is different in each addition/subtraction. In fact, suppose that they are chosen independently from, say, a uniform distribution on $[1/2, 3/2]$. Then with probability one, the algorithm will produce no primal degenerate or dual degenerate dictionary in any iteration. In Chapter 3, we discussed perturbing the right-hand side of a linear programming problem to avoid degeneracy in the primal simplex method, but back then the perturbation changed the problem. The present perturbation does not in any way affect the problem that is solved.

With the above randomization trick to resolve the degeneracy issue, the analysis of the convergence of the algorithm is straightforward. Indeed, let us consider a problem that is feasible and bounded (the questions regarding feasibility and boundedness are addressed in Exercise 7.10). For each nondegenerate pivot, the next value of μ will be strictly less than the current value. Since each of these μ values is determined by a partition of the variables into basics and nonbasics and there are only a finite number of such partitions, it follows that the method must reach a partition with a negative μ value in a finite number of steps.

3. The Parametric Self-Dual Simplex Method

In the previous section, we illustrated on an example a new algorithm for solving linear programming problems, called the parametric self-dual simplex method. In this section, we shall lay out the algorithm in matrix notation.

Our starting point is an initial dictionary as written in (6.10) and transcribed here for convenience:

$$\begin{aligned} \zeta &= \zeta^* - {z^*_{\mathcal{N}}}^T x_{\mathcal{N}} \\ x_{\mathcal{B}} &= x^*_{\mathcal{B}} - B^{-1} N x_{\mathcal{N}}, \end{aligned}$$

where

$$\begin{aligned} x^*_{\mathcal{B}} &= B^{-1} b \\ z^*_{\mathcal{N}} &= (B^{-1}N)^T c_{\mathcal{B}} - c_{\mathcal{N}} \\ \zeta^* &= c^T_{\mathcal{B}} x^*_{\mathcal{B}} = c^T_{\mathcal{B}} B^{-1} b. \end{aligned}$$

Generally speaking, we don't expect this dictionary to be either primal or dual feasible. So we perturb it by adding essentially arbitrary perturbations $\bar{x}_{\mathcal{B}}$ and $\bar{z}_{\mathcal{N}}$ to $x^*_{\mathcal{B}}$ and $z^*_{\mathcal{N}}$, respectively:

$$\begin{aligned} \zeta &= \zeta^* - (z^*_{\mathcal{N}} + \mu \bar{z}_{\mathcal{N}})^T x_{\mathcal{N}} \\ x_{\mathcal{B}} &= (x^*_{\mathcal{B}} + \mu \bar{x}_{\mathcal{B}}) - B^{-1} N x_{\mathcal{N}}. \end{aligned}$$

We assume that the perturbations are all strictly positive,

$$\bar{x}_{\mathcal{B}} > 0 \qquad \text{and} \qquad \bar{z}_{\mathcal{N}} > 0,$$

so that by taking μ sufficiently large the perturbed dictionary will be optimal. (Actually, to guarantee optimality for large μ, we only need to perturb those primal and dual variables that are negative in the initial dictionary.)

The parametric self-dual simplex method generates a sequence of dictionaries having the same form as the initial one—except, of course, the basis $\mathcal{B}$ will change, and hence all the data vectors ($z^*_\mathcal{N}$, $\bar{z}_\mathcal{N}$, $x^*_\mathcal{B}$, and $\bar{x}_\mathcal{B}$) will change too. Additionally, the current value of the objective function ζ^* will, with the exception of the first dictionary, depend on μ.

One step of the self-dual simplex method can be described as follows. First, we compute the smallest value of μ for which the current dictionary is optimal. Letting μ^* denote this value, we see that

$$\mu^* = \min\{\mu : z^*_\mathcal{N} + \mu\bar{z}_\mathcal{N} \geq 0 \text{ and } x^*_\mathcal{B} + \mu\bar{x}_\mathcal{B} \geq 0\}.$$

There is either a $j \in \mathcal{N}$ for which $z^*_j + \mu^*\bar{z}_j = 0$ or an $i \in \mathcal{B}$ for which $x^*_i + \mu^*\bar{x}_i = 0$ (if there are multiple choices, an arbitrary selection is made). If the blocking constraint corresponds to a nonbasic index $j \in \mathcal{N}$, then we do one step of the primal simplex method. If, on the other hand, it corresponds to a basic index $i \in \mathcal{B}$, then we do one step of the dual simplex method.

Suppose, for definiteness, that the blocking constraint corresponds to an index $j \in \mathcal{N}$. Then, to do a primal pivot, we declare x_j to be the entering variable, and we compute the step direction for the primal basic variables as the jth column of the dictionary. That is,

$$\Delta x_\mathcal{B} = B^{-1}Ne_j.$$

Using this step direction, we find an index $i \in \mathcal{B}$ that achieves the maximal value of $\Delta x_i/(x^*_i + \mu^*\bar{x}_i)$. Variable x_i is the leaving variable. After figuring out the leaving variable, the step direction vector for the dual nonbasic variables is just the negative of the ith row of the dictionary

$$\Delta z_\mathcal{N} = -(B^{-1}N)^T e_i.$$

After computing the primal and dual step directions, it is easy to see that the step length adjustments are given by

$$t = \frac{x^*_i}{\Delta x_i}, \qquad \bar{t} = \frac{\bar{x}_i}{\Delta x_i},$$
$$s = \frac{z^*_j}{\Delta z_j}, \qquad \bar{s} = \frac{\bar{z}_j}{\Delta z_j}.$$

And from these, it is easy to write down the new solution vectors:

$$x^*_j \leftarrow t, \quad \bar{x}_j \leftarrow \bar{t}, \quad z^*_i \leftarrow s, \quad \bar{z}_i \leftarrow \bar{s},$$

$$\begin{aligned} x^*_\mathcal{B} &\leftarrow x^*_\mathcal{B} - t\Delta x_\mathcal{B}, & \bar{x}_\mathcal{B} &\leftarrow \bar{x}_\mathcal{B} - \bar{t}\Delta x_\mathcal{B}, \\ z^*_\mathcal{N} &\leftarrow z^*_\mathcal{N} - s\Delta z_\mathcal{N}, & \bar{z}_\mathcal{N} &\leftarrow \bar{z}_\mathcal{N} - \bar{s}\Delta z_\mathcal{N}. \end{aligned}$$

Finally, the basis is updated by adding the entering variable and removing the leaving variable

$$\mathcal{B} \leftarrow \mathcal{B} \setminus \{i\} \cup \{j\}.$$

Compute $\mu^* = \max\left(\max_{\substack{j\in\mathcal{N}\\ \bar{z}_j>0}} -\frac{z_j^*}{\bar{z}_j}, \max_{\substack{i\in\mathcal{B}\\ \bar{x}_i>0}} -\frac{x_i^*}{\bar{x}_i} \right)$

While ($\mu^* > 0$) {

If max is achieved by

$j \in \mathcal{N}$:	$i \in \mathcal{B}$:
$\Delta x_{\mathcal{B}} = B^{-1}Ne_j$	$\Delta z_{\mathcal{N}} = -(B^{-1}N)^T e_i$
pick $i \in \text{argmax}_{i\in\mathcal{B}} \frac{\Delta x_i}{x_i^* + \mu^* \bar{x}_i}$	pick $j \in \text{argmax}_{j\in\mathcal{N}} \frac{\Delta z_j}{z_j^* + \mu^* \bar{z}_j}$
$\Delta z_{\mathcal{N}} = -(B^{-1}N)^T e_i$	$\Delta x_{\mathcal{B}} = B^{-1}Ne_j$

$$t = \frac{x_i^*}{\Delta x_i} \qquad \bar{t} = \frac{\bar{x}_i}{\Delta x_i}$$
$$s = \frac{z_j^*}{\Delta z_j} \qquad \bar{s} = \frac{\bar{z}_j}{\Delta z_j}$$

$$x_j^* \leftarrow t \qquad \bar{x}_j \leftarrow \bar{t}$$
$$z_i^* \leftarrow s \qquad \bar{z}_i \leftarrow \bar{s}$$

$$x_{\mathcal{B}}^* \leftarrow x_{\mathcal{B}}^* - t\Delta x_{\mathcal{B}} \qquad \bar{x}_{\mathcal{B}} \leftarrow \bar{x}_{\mathcal{B}} - \bar{t}\Delta x_{\mathcal{B}}$$
$$z_{\mathcal{N}}^* \leftarrow z_{\mathcal{N}}^* - s\Delta z_{\mathcal{N}} \qquad \bar{z}_{\mathcal{N}} \leftarrow \bar{z}_{\mathcal{N}} - \bar{s}\Delta z_{\mathcal{N}}$$

$\mathcal{B} \leftarrow \mathcal{B} \setminus \{i\} \cup \{j\}$

Recompute μ^* as above

}

FIGURE 7.1. The parametric self-dual simplex method.

The algorithm is summarized in Figure 7.1.

Exercises

In solving the following problems, the advanced pivot tool can be used to check your arithmetic:

www.princeton.edu/∼rvdb/JAVA/pivot/advanced.html

7.1 The final dictionary for

$$\begin{array}{ll} \text{maximize} & x_1 + 2x_2 + x_3 + x_4 \\ \text{subject to} & 2x_1 + x_2 + 5x_3 + x_4 \le 8 \\ & 2x_1 + 2x_2 + 4x_4 \le 12 \\ & 3x_1 + x_2 + 2x_3 \le 18 \\ & x_1, x_2, x_3, x_4 \ge 0 \end{array}$$

is

$$
\begin{array}{rcrcrcrcrcr}
\zeta & = & 12.4 & - & 1.2x_1 & - & 0.2x_5 & - & 0.9x_6 & - & 2.8x_4 \\
\hline
x_2 & = & 6 & - & x_1 & & & - & 0.5x_6 & - & 2x_4 \\
x_3 & = & 0.4 & - & 0.2x_1 & - & 0.2x_5 & + & 0.1x_6 & + & 0.2x_4 \\
x_7 & = & 11.2 & - & 1.6x_1 & + & 0.4x_5 & + & 0.3x_6 & + & 1.6x_4 \; .
\end{array}
$$

(the last three variables are the slack variables).

(a) What will be an optimal solution to the problem if the objective function is changed to

$$3x_1 + 2x_2 + x_3 + x_4?$$

(b) What will be an optimal solution to the problem if the objective function is changed to

$$x_1 + 2x_2 + 0.5x_3 + x_4?$$

(c) What will be an optimal solution to the problem if the second constraint's right-hand side is changed to 26?

7.2 For each of the objective coefficients in the problem in Exercise 7.1, find the range of values for which the final dictionary will remain optimal.

7.3 Consider the following dictionary which arises in solving a problem using the self-dual simplex method:

$$
\begin{array}{rcrcrcrcr}
\zeta & = & -3 & & & - & (-1+2\mu)x_1 & - & (3-\mu)x_3 \\
\hline
x_2 & = & -1 & + & \mu & + & x_1 & - & x_3 \\
x_4 & = & -4 & + & 3\mu & + & 3x_1 & - & 2x_3 \\
x_5 & = & 2 & & & + & x_1 & + & x_3 .
\end{array}
$$

(a) For which values of μ is the current dictionary optimal?

(b) For the next pivot in the self-dual simplex method, identify the entering and the leaving variable.

7.4 Solve the linear program given in Exercise 2.3 using the self-dual simplex method. *Hint: It is easier to use dictionary notation than matrix notation.*

7.5 Solve the linear program given in Exercise 2.4 using the self-dual simplex method. *Hint: It is easier to use dictionary notation than matrix notation.*

7.6 Solve the linear program given in Exercise 2.6 using the self-dual simplex method. *Hint: It is easier to use dictionary notation than matrix notation.*

7.7 Using today's date (MMYY) for the seed value, solve ten problems using the self-dual simplex method:

www.princeton.edu/∼rvdb/JAVA/pivot/pd1phase.html

7.8 Use the self-dual simplex method to solve the following problem:

$$\begin{array}{ll} \text{maximize} & 3x_1 - x_2 \\ \text{subject to} & \begin{aligned} x_1 - x_2 &\leq 1 \\ -x_1 + x_2 &\leq -4 \\ x_1,\ x_2 &\geq 0\,. \end{aligned} \end{array}$$

7.9 Let P_μ denote the perturbed primal problem (with perturbation μ). Show that if P_μ is infeasible, then $P_{\mu'}$ is infeasible for every $\mu' \leq \mu$. State and prove an analogous result for the perturbed dual problem.

7.10 Using the notation of Figure 7.1 state precise conditions for detecting infeasibility and/or unboundedness in the self-dual simplex method.

7.11 Consider the following one parameter family of linear programming problems (parametrized by μ):

$$\begin{array}{lrcl} \max & (4-4\mu)x_0 - 2x_1 - 2x_2 - 2x_3 - 2x_4 & & \\ \text{s.t.} & x_0 - x_1 \qquad\qquad\qquad\qquad & \leq & 1 \\ & x_0 \qquad - x_2 \qquad\qquad\quad & \leq & 2 \\ & x_0 \qquad\qquad - x_3 \qquad & \leq & 4 \\ & x_0 \qquad\qquad\qquad - x_4 & \leq & 8 \\ & x_0,\ x_1,\ x_2,\ x_3,\ x_4 & \geq & 0. \end{array}$$

Starting from $\mu = \infty$, use the parametric simplex method to decrease μ as far as possible. Don't stop at $\mu = 0$. If you cannot get to $\mu = -\infty$, explain why. *Hint: the pivots are straight forward and, after the first couple, a clear pattern should emerge which will make the subsequent pivots easy.* Clearly indicate the range of μ values for which each dictionary is optimal.

Notes

Parametric analysis has its roots in Gass and Saaty (1955). G.B. Dantzig's classic book (Dantzig 1963) describes the self-dual simplex method under the name of the *self-dual parametric simplex method*. It is a special case of "Lemke's algorithm" for the linear complementarity problem (Lemke 1965) (see Exercise 18.7). Smale (1983) and Borgwardt (1982) were first to realize that the parametric self-dual simplex method is amenable to probabilistic analysis. For a more recent discussion of homotopy methods and the parametric self-dual simplex method, see Nazareth (1986, 1987).