A SIMPLEX ALGORITHM FOR PIECEWISE-LINEAR PROGRAMMING II: FINITENESS, FEASIBILITY AND DEGENERACY

Robert FOURER

Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL 60208, USA

Received March 1985
Revised manuscript received 11 May 1987

The simplex method for linear programming can be extended to permit the minimization of any convex separable piecewise-linear objective, subject to linear constraints. Part I of this paper has developed a general and direct simplex algorithm for piecewise-linear programming, under convenient assumptions that guarantee a finite number of basic solutions, existence of basic feasible solutions, and nondegeneracy of all such solutions. Part II now shows how these assumptions can be weakened so that they pose no obstacle to effective use of the piecewise-linear simplex algorithm. The theory of piecewise-linear programming is thereby extended, and numerous features of linear programming are generalized or are seen in a new light. An analysis of the algorithm's computational requirements and a survey of applications will be presented in Part III.

Key words: Linear programming, simplex methods, piecewise-linear programming, non-differentiable optimization.

1. Introduction

A piecewise-linear program minimizes a convex separable piecewise-linear objective function, subject to linear constraints. Part I of this paper [8] has developed a version of the simplex algorithm that solves piecewise-linear programs directly. Although this piecewise-linear simplex algorithm differs in certain ways from familiar linear simplex methods, it has substantially the same steps and computational requirements as the algorithm that is commonly applied to solve linear programs in bounded variables.

The derivations in Part I rely on three restrictive assumptions about the nature of piecewise-linear programs. Part II now shows how these assumptions can be weakened so that they pose no obstacle to effective use of the piecewise-linear simplex algorithm. The theory of piecewise-linear programming is thereby extended, and numerous features of linear programming are generalized or are seen in a new light. The stage is thus set for a detailed analysis of the algorithm's computational requirements, in Part III [9] of this paper.

This research has been supported in part by the National Science Foundation under grant DMS-8217261.
Outline

The remainder of this introduction briefly reviews the terminology and properties of piecewise-linear programming that will be required in the sequel, and restates the piecewise-linear simplex algorithm. Subsequent sections consider relaxing or removing Part I's three assumptions, in the reverse of the order in which they were imposed.

According to the third assumption, nondegeneracy, every basic variable in every basic solution must lie strictly within one of the objective function's intervals of linearity. At bases that violate this restriction, the piecewise-linear simplex algorithm may be forced to perform degenerate iterations that fail to reduce the objective. Section 2 shows how the algorithm may be extended so that it is well-defined in the presence of degeneracy, and so that it discourages degenerate iterations or executes them in an efficient way. Section 3 then describes how the algorithm can be further modified, by extension of well-known perturbational and combinatorial approaches, to prevent infinite cycles of degenerate iterations.

The second assumption says that every piecewise-linear program has a basic feasible solution, if it has any feasible solutions at all. Section 4 demonstrates that a basic (but perhaps infeasible) solution can be constructed in all but certain exceptional situations that are easily circumvented. Once any basic solution is known, a basic feasible solution can be found by minimizing a piecewise-linear penalty function or by adding weighted penalty terms to the true objective, much as in bounded-variable linear programming. If penalties are added then the required size of their weights has a simple relation to the optimal dual values.

Under the first assumption, the objective function of a piecewise-linear program must be defined by only finitely many intervals of linearity. Section 5 shows that the piecewise-linear simplex algorithm remains well-defined under much weaker conditions that permit an infinite number of intervals. These conditions must again be somewhat strengthened, however, if the algorithm is to retain its property of finite termination. The principal difficulty in the analysis is that, although the relaxation of finiteness need not affect the local nature of piecewise-linear functions, it does significantly expand their range of asymptotic behavior.

Piecewise-linear programs

A closed proper convex piecewise-linear (P-L) function \([c/\gamma]_k\) of \(x_k\) is characterized (to within an additive constant) by an increasing sequence of breakpoints \(\gamma_k^{(h)}\),

\[
\ldots < \gamma_k^{(-2)} < \gamma_k^{(-1)} < \gamma_k^{(0)} < \gamma_k^{(1)} < \gamma_k^{(2)} < \cdots
\]

and by an increasing sequence of slopes \(c_k^{(h)}\),

\[
\ldots < c_k^{(-2)} < c_k^{(-1)} < c_k^{(0)} < c_k^{(1)} < c_k^{(2)} < \cdots.
\]

At least one slope or one breakpoint must be finite. If \(c_k^{(h)}\) is finite, then \([c/\gamma]_k x_k\) is linear with slope \(c_k^{(h)}\) on the interval between \(\gamma_k^{(h)}\) and \(\gamma_k^{(h+1)}\). Infinite slopes are
interpreted as follows:

\[ c_k^{(s)} = -\infty \Rightarrow [c/\gamma]_k x_k = \infty \quad \text{for all } x_k < \gamma_k^{(s+1)}; \]
\[ c_k^{(t)} = +\infty \Rightarrow [c/\gamma]_k x_k = \infty \quad \text{for all } x_k > \gamma_k^{(t)}. \]

Thus the slope-breakpoint sequences can effectively begin with either some \( c_k^{(s)} = -\infty \) and \( \gamma_k^{(s+1)} \) finite, or some \( \gamma_k^{(s)} = -\infty \) and \( c_k^{(s)} \) finite; they can effectively end with either some \( \gamma_k^{(t)} \) finite and \( c_k^{(t)} = +\infty \), or some \( c_k^{(t)} \) finite and \( \gamma_k^{(t+1)} = +\infty \).

A closed proper convex separable piecewise-linear function \([c/\gamma]\) of \( x = (x_1, \ldots, x_n) \) is defined by a sum of closed proper convex piecewise-linear functions:

\[ [c/\gamma]x = \sum_{k=1}^{n} [c/\gamma]_k x_k. \]

A piecewise-linear program (P-LP) minimizes such a function over all \( x \) that satisfy certain linear equations:

\[
\begin{align*}
\text{Minimize} & \quad [c/\gamma]x \\
\text{subject to} & \quad Ax = b.
\end{align*}
\]

The matrix \( A = [a_1, \ldots, a_n] \) is \( m \times n \) and may be taken to have linearly independent rows. A solution \( \bar{x} \) to \( Ax = b \) is feasible if \([c/\gamma]\bar{x}\) is finite; thus \( \bar{x} \) is feasible if each \( \bar{x}_k \) lies in an interval of finite slope of \([c/\gamma]_k\).

**Basic solutions for piecewise-linear programs**

A basis matrix \( B \) is an \( m \times m \) nonsingular submatrix of \( A \). The vector \( x_B \) denotes the basic variables corresponding to the columns of \( B \), and \( x_N \) denotes the remaining, nonbasic variables; an individual variable \( x_k \) is described as \( x_{BK} \) when it is in the basic set, and as \( x_{NK} \) when it is nonbasic. An analogous notation is employed to distinguish other vectors and values that correspond to basic and nonbasic variables.

Given a basis matrix \( B \) and a vector \( \gamma_N \) of finite breakpoints, a unique basic solution is defined by

\[ \bar{x}_N = \gamma_N, \]
\[ B\bar{x}_B = b - \sum_j a_{Nj} \gamma_{Nj}. \]

Under the assumption of nondegeneracy, the values of the basic variables determine a unique vector of slopes \( c_B \), where

\[ c_{Bi} = c_{Bi}^{(h)} \quad \text{if} \quad \gamma_{Bi}^{(h)} < \bar{x}_{Bi} < \gamma_{Bi}^{(h+1)}. \]

The basic solution is necessarily feasible if \( -\infty < c_{Bi} < +\infty \) for all \( i \).
Relative to the breakpoint $\gamma_{Nj}$ at which $x_{Nj}$ is fixed, the slopes and breakpoints of $[c/\gamma]_{Nj}$ are denoted as follows:

<table>
<thead>
<tr>
<th>Interval</th>
<th>Slope of $x_{Nj}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$[\gamma_{Nj}^{-(h)}, \gamma_{Nj}^{-(h-1)}]$</td>
<td>$c_{Nj}^{-(h)}$</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>$[\gamma_{Nj}^{-(2)}, \gamma_{Nj}^{-(1)}]$</td>
<td>$c_{Nj}^{-(2)}$</td>
</tr>
<tr>
<td>$[\gamma_{Nj}^{-(1)}, \gamma_{Nj}^{-(0)}]$</td>
<td>$c_{Nj}^{-(1)}$</td>
</tr>
<tr>
<td>$[\gamma_{Nj}, \gamma_{Nj}^{+(1)}]$</td>
<td>$c_{Nj}^{+(1)}$</td>
</tr>
<tr>
<td>$[\gamma_{Nj}^{+(1)}, \gamma_{Nj}^{+(2)}]$</td>
<td>$c_{Nj}^{+(2)}$</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>$[\gamma_{Nj}^{+(h-1)}, \gamma_{Nj}^{+(h)}]$</td>
<td>$c_{Nj}^{+(h)}$</td>
</tr>
</tbody>
</table>

The slopes and breakpoints for each $[c/\gamma]_{Bi}$ are described relative to $c_{Bi}$ by a complementary notation:

<table>
<thead>
<tr>
<th>Interval</th>
<th>Slope of $x_{Bi}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$[\gamma_{Bi}^{-(h)}, \gamma_{Bi}^{-(h-1)}]$</td>
<td>$c_{Bi}^{-(h)}$</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>$[\gamma_{Bi}^{-(3)}, \gamma_{Bi}^{-(2)}]$</td>
<td>$c_{Bi}^{-(2)}$</td>
</tr>
<tr>
<td>$[\gamma_{Bi}^{-(2)}, \gamma_{Bi}^{-(1)}]$</td>
<td>$c_{Bi}^{-(1)}$</td>
</tr>
<tr>
<td>$[\gamma_{Bi}^{-(1)}, \gamma_{Bi}^{-(0)}]$</td>
<td>$c_{Bi}^{-(0)}$</td>
</tr>
<tr>
<td>$[\gamma_{Bi}^{+(1)}, \gamma_{Bi}^{+(2)}]$</td>
<td>$c_{Bi}^{+(1)}$</td>
</tr>
<tr>
<td>$[\gamma_{Bi}^{+(2)}, \gamma_{Bi}^{+(3)}]$</td>
<td>$c_{Bi}^{+(2)}$</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>$[\gamma_{Bi}^{+(h)}, \gamma_{Bi}^{+(h+1)}]$</td>
<td>$c_{Bi}^{+(h)}$</td>
</tr>
</tbody>
</table>

By convention, $c_{Nj}^{-(0)} = c_{Bi}^{-(1)}$, $c_{Nj}^{+(0)} = c_{Bi}^{-(1)}$, and $\gamma_{Nj}^{-(0)} = \gamma_{Nj}^{+(0)} = \gamma_{Nj}$. Similarly, $\gamma_{Bi}^{-(0)} = \gamma_{Bi}^{+(1)}$, $\gamma_{Bi}^{+(0)} = \gamma_{Bi}^{-(1)}$, and $c_{Bi}^{-(0)} = c_{Bi}^{+(0)} = c_{Bi}$.

Starting from a basic solution $\bar{x}$, a line of solutions may be generated by pushing some nonbasic variable away from its current breakpoint $\gamma_{Nj}$ while adjusting the values of the basic variables to maintain $Ax = b$:

$$x_{Nj} = \gamma_{Nj} + \theta, \quad x_B = \bar{x}_B - \theta y_B, \quad \text{where } By_B = a_{Nj}, \quad \text{or}$$

$$x_{Nj} = \gamma_{Nj} - \theta, \quad x_B = \bar{x}_B - \theta y_B, \quad \text{where } By_B = -a_{Nj}.$$  \hspace{1cm} (1.1)
The rate of increase in the objective as $x_{Nj}$ increases from $\gamma_{Nj}$ is given by a reduced cost $d^+_{Nj}$, and the rate of decrease in the objective as $x_{Nj}$ decreases from $\gamma_{Nj}$ is given by $d^-_{Nj}$:

$$d^+_{Nj} = c^+_{Nj} - c_B y_B = c^+_{Nj} - \pi a_{Nj}, \quad \text{or}$$

$$d^-_{Nj} = c^-_{Nj} + c_B y_B = c^-_{Nj} - \pi a_{Nj},$$

where $\pi B = c_B$. Suppose that $x_{Np}$, in particular, is increased or decreased by an amount $\theta$. Then it must lie between some two breakpoints:

$$\gamma_{Np}^{(r-1)} < \gamma_{Np} + \theta \leq \gamma_{Np}^{(r)}, \quad \text{or}$$

and the basic variables $x_{Bi}$ must also lie between certain breakpoints:

$$\gamma_{Bi}^{(r-1)} < \bar{x}_{Bi} - \theta y_{Bi} \leq \gamma_{Bi}^{(r)} \quad \text{for each } y_{Bi} < 0, \quad \text{and}$$

$$\gamma_{Bi}^{(r-1)} > \bar{x}_{Bi} - \theta y_{Bi} \geq \gamma_{Bi}^{(r)} \quad \text{for each } y_{Bi} > 0.$$

The rate of increase in the objective as $x_{Nj} \rightarrow \gamma_{Nj} + \theta$, or the rate of decrease in the objective as $x_{Nj} \rightarrow \gamma_{Nj} - \theta$, is given by an effective reduced cost $d^+_{Np}(\theta)$ or $d^-_{Np}(\theta)$:

$$d^+_{Np}(\theta) = c^+_{Np} - \sum_{y_{Bi} < 0} c^+_{Bi} y_{Bi} - \sum_{y_{Bi} > 0} c^-_{Bi} y_{Bi}, \quad \text{or}$$

$$d^-_{Np}(\theta) = c^-_{Np} - \sum_{y_{Bi} < 0} c^-_{Bi} y_{Bi} + \sum_{y_{Bi} > 0} c^+_{Bi} y_{Bi}.$$

Both are monotonic in $\theta$: $d^+_{Nj}(\theta_1) \leq d^+_{Nj}(\theta_2)$ and $d^-_{Nj}(\theta_1) \geq d^-_{Nj}(\theta_2)$ for any $\theta_1 < \theta_2$.

**A simplex algorithm for piecewise-linear programs**

Let $\bar{x}$ be a basic feasible solution defined by a basis matrix $B$ and breakpoint values $\gamma_N$, and let $c_B$ be the corresponding vector of basic slopes. One iteration of the piecewise-linear simplex algorithm proceeds as follows:

1. **Solve** $\pi B = c_B$.
2. **Test for optimality:** If the nonbasic variables satisfy

$$c^+_{Nj} - \pi a_{Nj} \leq c^+_{Nj} \quad \text{for all } j,$$

then the basic solution is optimal. **Stop.**
3. **Select an entering variable:** Choose a variable $x_{Np}$ such that

$$d^+_{Np} = c^+_{Np} - \pi a_{Np} < 0 \quad \text{or} \quad d^-_{Np} = c^-_{Np} - \pi a_{Np} > 0.$$

4. **Solve** $By_B = a_{Np}$ if $d^+_{Np} < 0$, or $By_B = -a_{Np}$ if $d^-_{Np} > 0$.
5. **Test for unboundedness:** If

$$\gamma_{Bi}^{(r-1)} = +\infty \quad \text{for all } y_{Bi} < 0,$$

$$\gamma_{Bi}^{(r-1)} = -\infty \quad \text{for all } y_{Bi} > 0,$$
then the objective can decrease without bound. Stop.

(6) Select a leaving variable: Choose any variable $x_{Bq}$ and breakpoint $\gamma_{Bq}^{+\{r\}}$ or $\gamma_{Bq}^{-\{r\}}$ such that
   
   a) $(\bar{x}_{Bq} - \gamma_{Bq}^{+\{r\}})/y_{Bq} = \bar{\theta}$ and $y_{Bq} < 0$, or
   b) $(\bar{x}_{Bq} - \gamma_{Bq}^{-\{r\}})/y_{Bq} = \bar{\theta}$ and $y_{Bq} > 0$;

   or choose $x_{Np}$ and any breakpoint $\gamma_{Np}^{+\{r\}}$ or $\gamma_{Np}^{-\{r\}}$ such that
   c) $-(\gamma_{Np} - \gamma_{Np}^{+\{r\}}) = \bar{\theta}$ and $d_{Np}^+ < 0$, or
   d) $(\gamma_{Np} - \gamma_{Np}^{-\{r\}}) = \bar{\theta}$ and $d_{Np}^- > 0$;

   provided $\bar{\theta}$ is small enough that
   $d_{Np}^+ < 0$ and $d_{Np}^+(\bar{\theta}) < 0$, or $d_{Np}^- > 0$ and $d_{Np}^-(\bar{\theta}) > 0$.

(7) Update the basic solution: Reset

   $\bar{x}_B \leftarrow \bar{x}_B - \bar{\theta} y_B$, 

   $\bar{x}_{Np} \leftarrow \gamma_{Np} + \bar{\theta}$ if $d_{Np}^+ < 0$, 

   $\bar{x}_{Np} \leftarrow \gamma_{Np} - \bar{\theta}$ if $d_{Np}^- > 0$, 

   and set $c_{Bi}$ accordingly for $i \neq q$:

   $c_{Bi} = c_{Bi}^{+\{r\} - 1}$ for $y_{Bi} < 0$, 

   $c_{Bi} = c_{Bi}^{-\{r\} - 1}$ for $y_{Bi} > 0$.

(8) Change the basis, according to the choice made in step (6):

   If (a) or (b): Replace $a_{Bq}$ by $a_{Np}$ in $B$.
   Set $c_{Np} = c_{Np}^{+\{r\}}$ (if $d_{Np}^+ < 0$) or $c_{Np} = c_{Np}^{-\{r\}}$ (if $d_{Np}^- > 0$).
   Set $\gamma_{Bq} = \gamma_{Bq}^{+\{r\}}$ (if $y_{Bq} < 0$) or $\gamma_{Bq} = \gamma_{Bq}^{-\{r\}}$ (if $y_{Bq} > 0$).
   If (c) or (d):
   Set $\gamma_{Np} = \gamma_{Np}^{+\{r\}}$ (if $d_{Np}^+ < 0$) or $\gamma_{Np} = \gamma_{Np}^{-\{r\}}$ (if $d_{Np}^- > 0$).

   The piecewise-linear simplex algorithm carries out a series of these iterations, each starting from the $B$, $c_B$, $\gamma_N$, and $\bar{x}$ determined at the end of the preceding iteration. The algorithm stops when, at some iteration, the termination conditions of step (2) or step (5) are satisfied.

2. Degeneracy

Suppose that the nondegeneracy assumption is dropped. Then the algorithm defined above may encounter a basic feasible solution in which certain basic variables
lie exactly at finite breakpoints. For each $i$ such that $\bar{x}_B = \gamma^{(s_i)}_B$, the variable $x_B$ cannot be associated with a unique slope $c_B$; the slope of $[c/\gamma] B x_B$ is either $c^{(s_i)}_B$ or $c^{(s_i-1)}_B$ depending on whether $x_B$ increases or decreases from $\bar{x}_B$. The vector $\pi$ and the reduced costs $d^+_N$ and $d^-_N$ are thus ill-defined in steps (1)-(3), and the algorithm cannot proceed.

To deal with this difficulty, the piecewise-linear simplex algorithm may arbitrarily choose $c_B$ as either $c^{(s_i)}_B$ or $c^{(s_i-1)}_B$ whenever $\bar{x}_B = \gamma^{(s_i)}_B$. The first part of this section shows that the algorithm can then be applied unambiguously, even though it may encounter degenerate iterations at which the objective value fails to decrease. The termination conditions of steps (2) and (5) are also seen to remain valid.

The remainder of this section takes a closer look at the consequences of degeneracy in P-L simplex steps. An iteration is shown to compute both true reduced costs that determine whether the objective can be reduced, and nominal reduced costs that determine optimality. The arrangements for computing these quantities can have a substantial effect upon both the complexity of an iteration and the path of iterations.

Further modifications are necessary to insure that the algorithm cannot fall into an endless cycle of degenerate iterations. Criteria for the prevention of cycling are derived in Section 3 below.

**Accommodation of degenerate bases**

Suppose the that P-L simplex algorithm is modified so that a finite slope $c_B$ is assigned arbitrarily to each degenerate basic variable $\bar{x}_B = \gamma^{(s_i)}_B$, as described above. The slopes and breakpoints associated with $x_B$ are then as follows:

$$
\begin{align*}
\text{If } c_B = c^{(s_i-1)}_B & > -\infty: & \text{If } c_B = c^{(s_i)}_B & < +\infty: \\
 c^+_B & = c^{(s_i-1)}_B & c^+_B & = c^{(s_i)}_B \\
c^-_B & = c^{(s_i-2)}_B & c^-_B & = c^{(s_i-1)}_B \\
\gamma^+_B & = \gamma^{(s_i-1)}_B & \gamma^+_B & = \gamma^{(s_i)}_B \\
\gamma^-_B & = \gamma^{(s_i-2)}_B & \gamma^-_B & = \gamma^{(s_i-1)}_B
\end{align*}
$$

If both $c^{(s_i-1)}_B = -\infty$ and $c^{(s_i)}_B = +\infty$ then no finite $c_B$ can be assigned. In such a case, however, every feasible solution must have $\bar{x}_B = \gamma^{(s_i)}_B$, so that $x_i$ can be fixed and dropped from the problem in advance.

Once $c_B$ is chosen, $\pi$ can be computed in step (1). If the conditions of step (2) are satisfied, then the basis must be optimal; the proof of optimality in Part I of this paper [8] is applicable regardless of degeneracy.

If the optimality conditions do not hold, the iteration may proceed to select some $x_{N_p}$ in step (3) and to compute $y_B$ in step (4). The meaning of $y_B$ is preserved under degeneracy: when $x_{N_p}$ is increased (if $d^+_{N_p} < 0$) or decreased (if $d^-_{N_p} > 0$) by $\theta$ from $\gamma_{N_p}$, each $x_B$ changes by $\theta y_B$. Thus the proof of unboundedness in Part I is also applicable. If the conditions of step (5) are satisfied, then the P-LP must have no finite minimum.
If the unboundedness conditions do not hold, then a leaving variable can be selected. The meanings of the effective reduced costs are also preserved under degeneracy: for any positive \( \theta \), either \( d^+_{Np}(\theta) \) is the rate of increase in the objective as \( x_{Np} \) approaches \( \gamma_{Np} + \theta \) from below, or \( d^-_{Np}(\theta) \) is the rate of decrease in the objective as \( x_{Np} \) approaches \( \gamma_{Np} - \theta \) from above. Thus if step (6) can find a \( \theta > 0 \) such that \( d^+_{Np}(\theta) < 0 \) or \( d^-_{Np}(\theta) > 0 \), the iteration can be completed as before, and the new basic solution will yield a lower value of the objective.

In the presence of degeneracy, however, the algorithm may not be able to choose a positive \( \theta \). For each \( \bar{x}_{Bi} = \gamma^{(s)}_{Bi} \), one of the ratios in step (6) may be zero:

\[
(\bar{x}_{Bi} - \gamma^{(1)}_{Bi})/y_{Bi} = 0 \quad \text{if} \quad y_{Bi} < 0 \quad \text{and} \quad c_{Bi} = e^{(s-1)}_{Bi}, \quad \text{or} \quad \ \\
(\bar{x}_{Bi} - \gamma^{-1}_{Bi})/y_{Bi} = 0 \quad \text{if} \quad y_{Bi} > 0 \quad \text{and} \quad c_{Bi} = e^{(s)}_{Bi}.
\]

The proof of termination in Part I argued that \( \tilde{\theta} \) could always be chosen positive, because it could always be at least as large as the smallest ratio. If either of the above situations occurs, however, then the smallest ratio is zero, and the reasoning of the proof breaks down. Indeed, it can happen at a degenerate basis that

\[
 d^+_{Np} < 0 \quad \text{but} \quad d^-_{Np}(\theta) \geq 0, \quad \text{or} \quad d^-_{Np} > 0 \quad \text{but} \quad d^-_{Np}(\theta) \leq 0,
\]

for all \( \theta > 0 \). Then even though the computed reduced cost is favorable, the objective value must increase when \( x_{Np} \) is pushed up or down from \( \gamma_{Np} \).

The P-L simplex algorithm can accommodate this situation provided that step (6) is interpreted to allow a choice of \( \tilde{\theta} = 0 \). The variable \( x_{Np} \) then enters the basis at its breakpoint value \( \gamma_{Np} \), and some degenerate variable \( x_{Bq} \) leaves the basis at its current breakpoint value. This is a degenerate iteration; the basis changes but the basic solution and the objective value remain the same.

**Reduced costs at degenerate bases**

At any basic solution, regardless of degeneracy, it is meaningful to imagine that a nonbasic variable \( x_{Nj} \) is pushed up or down by a small amount \( \delta \) from its current breakpoint value \( \gamma_{Nj} \), and that the basic variables are adjusted accordingly to maintain \( Ax = b \). From formula (1.1), the resulting values of \( x_{Nj} \) and the basic variables may be written as

\[
x_{Nj} = \gamma_{Nj} + \delta \quad \text{or} \quad x_{Nj} = \gamma_{Nj} - \delta \quad \text{where} \quad Bw_{Nj:B} = a_{Nj}.
\]

For each \( x_{Nj} \) there is a different vector \( w_{Nj:B} \), whose element \( w_{Nj:B} \) determines how \( x_{Bi} \) will change as \( x_{Nj} \) is pushed. For the entering variable \( x_{Np} \) in particular, \( w_{Np:B} \) is just either \( y_{B} \) (if \( d^+_{Np} < 0 \)) or \( -y_{B} \) (if \( d^-_{Np} > 0 \)) as computed in step (4) of the algorithm.

All relevant reduced costs for \( x_{Nj} \) can be expressed in terms of \( w_{Nj:B} \). Given any choice of slopes \( c_{Bi} \), formula (1.2) implies that the values \( d^+_{Nj} \) and \( d^-_{Nj} \) computed
in step (2) are
\[
\begin{align*}
\frac{d^+}{N_j} &= c_{N_j}^{[1]} - \sum_i c_{Bi} w_{N_j:Bi}, \\
\frac{d^-}{N_j} &= c_{N_j}^{[1]} - \sum_i c_{Bi} w_{N_j:Bi},
\end{align*}
\]
(2.1)

where \(c_{N_j}^{[1]}\) and \(c_{N_j}^{[1]}\) have been defined as the slopes in \([c/\gamma]_{N_j}\) to the left and right, respectively, of \(\gamma_{N_j}\). Let \(c_{Bi}^-\) and \(c_{Bi}^+\) similarly denote the slopes in \([c/\gamma]_{Bi}\) to the left and right of \(x_{Bi}\). Then for all small enough \(\delta > 0\), the effective reduced costs must be
\[
\begin{align*}
\frac{d^+}{N_j}(\delta) &= c_{N_j}^{[1]} - \sum_{w_{N_j:Bi} < 0} c_{Bi}^+ w_{N_j:Bi} - \sum_{w_{N_j:Bi} > 0} c_{Bi}^- w_{N_j:Bi}, \\
\frac{d^-}{N_j}(\delta) &= c_{N_j}^{[1]} - \sum_{w_{N_j:Bi} < 0} c_{Bi}^+ w_{N_j:Bi} - \sum_{w_{N_j:Bi} > 0} c_{Bi}^- w_{N_j:Bi}.
\end{align*}
\]
(2.2)

If all basic variables lie between breakpoints then \(c_{Bi}^- = c_{Bi} = c_{Bi}^-\), so that \(d_{N_j}^+ = d_{N_j}(\delta)\) and \(d_{N_j}^- = d_{N_j}(\delta)\) as expected. If any \(x_{Bi} = \gamma_{Bi}^{(s_i)}\), however, then \(c_{Bi} = c_{Bi}^{(s_i)^{-1}}\) but \(c_{Bi}^+ = c_{Bi}^{(s_i)}\), and there must be some differences:

\[
\begin{align*}
\text{Coefficients of } w_{N_j:Bi} &\text{ differ in . . . if . . . and . . .} \\
d_{N_j}^+ \text{ and } d_{N_j}^+ (\delta) &\quad w_{N_j:Bi} < 0 \quad c_{Bi} = c_{Bi}^{(s_i)^{-1}} \\
&\quad w_{N_j:Bi} > 0 \quad c_{Bi} = c_{Bi}^{(s_i)}
\end{align*}
\]
\[
\begin{align*}
d_{N_j}^- \text{ and } d_{N_j}^- (\delta) &\quad w_{N_j:Bi} < 0 \quad c_{Bi} = c_{Bi}^{(s_i)} \\
&\quad w_{N_j:Bi} > 0 \quad c_{Bi} = c_{Bi}^{(s_i)^{-1}}
\end{align*}
\]

To express the total difference concisely, let \(\Omega_{N_j}^+\) and \(\Omega_{N_j}^-\) be the disjoint sets of indices \(i\) for which the coefficients of \(w_{N_j:Bi}\) differ in \(d_{N_j}^+, d_{N_j}^+ (\delta)\) and \(d_{N_j}^-, d_{N_j}^- (\delta)\), respectively. Then (2.1) and (2.2) together imply that
\[
\begin{align*}
d_{N_j}^+ (\delta) &= d_{N_j}^+ - \sum_{i \in \Omega_{N_j}^+} (c_{Bi}^{(s_i)} - c_{Bi}^{(s_i)^{-1}}) |w_{N_j:Bi}|, \\
d_{N_j}^- (\delta) &= d_{N_j}^- - \sum_{i \in \Omega_{N_j}^-} (c_{Bi}^{(s_i)} - c_{Bi}^{(s_i)^{-1}}) |w_{N_j:Bi}|.
\end{align*}
\]
(2.3)

Since convexity requires \(c_{Bi}^{(s_i)} \geq c_{Bi}^{(s_i)^{-1}}\), these relationships confirm that \(d_{N_j}^- (\delta) \geq d_{N_j}^+\) and \(d_{N_j}^- (\delta) \leq d_{N_j}^+\) for all \(j\).

The quantities \(d_{N_j}^+ (\delta)\) and \(d_{N_j}^- (\delta)\), for sufficiently small positive \(\delta\), are properly regarded as the \textit{true reduced costs} of \(x_{N_j}\), since they describe the actual change in the objective value when \(x_{N_j}\) is pushed up or down. The quantities \(d_{N_j}^+\) and \(d_{N_j}^-\) are \textit{nominal reduced costs} which, as shown above, can only overstate the true reduced costs. (Even so, at most one of \(d_{N_j}^+\) and \(d_{N_j}^-\) can be favorable, since \(d_{N_j}^+ - d_{N_j}^- = c_{N_j}^{[1]} - c_{N_j}^{[1]} \geq 0\) just as in the nondegenerate case.)
Because the nominal reduced costs can only overstate the true reduced costs, the signs of the reduced costs at a given basis must be related in one of the following three ways:

<table>
<thead>
<tr>
<th>Nominal reduced costs</th>
<th>True reduced costs</th>
<th>Implication</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) All $d^+<em>{Nj} \geq 0$, $d^-</em>{Nj} \leq 0$.</td>
<td>All $d^+<em>{Nj}(\delta) \geq 0$, $d^-</em>{Nj}(\delta) \leq 0$.</td>
<td>Basis is optimal.</td>
</tr>
<tr>
<td>(b) Some $d^+<em>{Np} &lt; 0$, $d^-</em>{Np} &gt; 0$.</td>
<td>All $d^+<em>{Nj}(\delta) \geq 0$, $d^-</em>{Nj}(\delta) \leq 0$.</td>
<td>Basis may not be optimal. Must take degen. iter.</td>
</tr>
<tr>
<td>(c) Some $d^+<em>{Np} &lt; 0$, $d^-</em>{Np} &gt; 0$.</td>
<td>Some $d^+<em>{Np}(\delta) &lt; 0$, $d^-</em>{Np}(\delta) &gt; 0$.</td>
<td>Basis is not optimal. Can take nondegen. iter.</td>
</tr>
</tbody>
</table>

The different roles of the nominal and true reduced costs are clear. Optimality must be confirmed by the signs of $d^+_{Nj}$ and $d^-_{Nj}$, while the possibility of a nondegenerate iteration depends on the signs of $d^+_{Np}(\delta)$ and $d^-_{Np}(\delta)$.

When the P-L simplex algorithm is modified to accommodate degeneracy as described earlier in this section, the first part of an iteration computes only the nominal reduced costs, which suffice to identify case (a) above. The remainder of an iteration effectively determines the sign of the true reduced cost for just the entering variable $X_Np$. If a $\delta$ is found such that $d^+_{Np}(\delta) < 0$ or $d^-_{Np}(\delta) > 0$ then case (c) is identified, and a nondegenerate iteration is taken. Otherwise cases (b) and (c) cannot be distinguished; indeed, (b) can be confirmed only by checking all of the true reduced costs.

**Selection of an entering variable under degeneracy**

The preceding analysis suggests that the P-L simplex algorithm might usefully employ the true as well as the nominal reduced costs in selecting an entering variable. If step (2) does not detect optimality, then step (3) may look for an $x_{Np}$ that has a favorable true reduced cost; if no such variable can be found, then $x_{Np}$ may instead be chosen to have favorable nominal reduced cost:

(2) **Test for optimality:** If the nonbasic variables satisfy

$$d^+_{Nj} \geq 0 \text{ and } d^-_{Nj} \leq 0$$

for all $j$, then the basic solution is optimal. Stop.

(3) **Select an entering variable:** If the nonbasic variables satisfy

$$d^+_{Nj}(\delta) \geq 0 \text{ and } d^-_{Nj}(\delta) \leq 0$$

for all $j$, then begin a degenerate iteration by choosing a variable $x_{Np}$ such that

$$d^+_{Np} < 0 \text{ or } d^-_{Np} > 0;$$
otherwise, begin a nondegenerate iteration by choosing a variable $x_{Np}$ such that

$$d_{Np}^+(\delta) < 0 \quad \text{or} \quad d_{Np}^-(\delta) > 0.$$  

This procedure distinguishes cases (b) and (c) above, and has the advantage that it leads to a nondegenerate iteration whenever possible.

From a computational standpoint, however, the use of true reduced costs in step (3) is problematical. Since an iteration must examine nominal reduced costs in any case, to test for optimality in step (2), the determination of true reduced costs can only add to an iteration's inherent complexity. In particular, the algorithm cannot compute the true reduced cost of $x_{Nj}$ until it knows the sign of $w_{Nj:Bi}$ for each degenerate $x_{Bi}$.

One possible arrangement is to compute all reduced costs from the vectors $w_{Nj:Bi}$, using some combination of formulas (2.1), (2.2) and (2.3). Then the vector $\pi$ of step (1) is not required, and the vector $y_B$ of step (4) is $\pm w_{Np:B}$ as previously noted. To determine the necessary vectors $w_{Nj:Bi}$, an iteration could solve $Bw_{Nj:Bi} = a_{Nj}$ for each $x_{Nj}$ whose reduced costs were to be computed; but the solution of even a few such systems would add significantly to the cost of an iteration. If instead all vectors $w_{Nj:Bi}$ were updated at each iteration to reflect the change of basis, then the algorithm would perform the equivalent of a simplex "tableau" update [5, 7] whose drawbacks of inefficiency and instability are well known.

An alternative is to calculate $\pi$ and the nominal reduced costs as in Section 1’s statement of the algorithm, and to calculate true reduced costs from the nominal ones by formula (2.3). Then only the values $w_{Nj:Bi}$ corresponding to degenerate variables $x_{Bi}$ need be computed. Writing $e_{Bi}$ for the unit vector such that $Be_{Bi} = a_{Bi}$, it follows that

$$w_{Nj:Bi} = \sigma_{Bi}a_{Nj} \quad \text{where} \quad \sigma_{Bi}B = e_{Bi}.$$  

Thus it suffices to determine $\sigma_{Bi}$ for each degenerate $x_{Bi}$. To avoid solving $\sigma_{Bi}B = e_{Bi}$ at each iteration, $\sigma_{Bi}$ may be updated by

$$\sigma_{Bq} \leftarrow \frac{\sigma_{Bq}}{w_{Np:Bq}},$$  

$$\sigma_{Bi} \leftarrow \frac{\sigma_{Bi} - w_{Np:Bi}\sigma_{Bq}}{\sigma_{Bq}} \quad \text{for all} \ i \neq q.$$  

Thus the principal computations are the inner products $\sigma_{Bi}a_{Nj}$ for each $i$ and $j$ such that $x_{Bi}$ is degenerate and $d_{Nj}^+ < 0$ or $d_{Nj}^- > 0$. In addition, $\sigma_{Bi}B = e_{Bi}$ must be solved for each newly degenerate $x_{Bi}$.

These analyses suggest that the P-L simplex algorithm cannot compute many true reduced costs if it is to retain the essential computational features of the iteration presented in Section 1 (and of comparable linear simplex iterations). Instead, the algorithm must choose some $c_B$ and rely on the nominal reduced costs. Different choices of $c_B$ necessarily yield different values of $d_{Nj}^-$ and $d_{Nj}^+$, however, which overstate $d_{Nj}(\delta)$ and $d_{Nj}(\delta)$ by differing amounts. Hence certain strategies for choosing $c_B$ might affect the algorithm’s performance.
In formula (2.3), the discrepancies between the nominal and true reduced costs are expressed as sums over the sets $\Omega^+_N$ and $\Omega^-_N$. For each $\bar{x}_{Bi} = \gamma_{(s_i)}$ such that $w_{Nj;Bi} \neq 0$, choosing $c_{Bi} = c_{Bi}^{(s_i-1)}$ places $i$ into exactly one of $\Omega^+_N$ or $\Omega^-_N$, and choosing $c_{Bi} = c_{Bi}^{(s)}$ places $i$ into the other. Thus it is possible to choose $c_B$ to make either $\Omega^+_N$ or $\Omega^-_N$ empty, and hence to make either $d^+_N = d^+_N(\delta)$ or $d^-_N = d^-_N(\delta)$. Unfortunately, different choices of $c_B$ are generally necessary to achieve these equalities for different $j$. The most one can hope is that some choice of $c_B$ will tend to make $\Omega^+_N$ small for those $x_{Nj}$ such that $d^+_N < 0$, and $\Omega^-_N$ small for those such that $d^-_N > 0$.

Such a choice cannot be made efficiently in any precise sense, since membership in $\Omega^+_N$ or $\Omega^-_N$ also depends on the unknown signs of the values $w_{Nj;Bi}$. Nevertheless, there may be some merit in a heuristic choice. Intuitively, $\Omega^+_N$ and $\Omega^-_N$ represent “wrong guesses” about the behavior of the basic variables: they contain the indices $i$ for which $x_{Bi}$ will increase although $c_{Bi}$ has been taken as the slope to the left of $\bar{x}_{Bi}$, or for which $x_{Bi}$ will decrease although $c_{Bi}$ has been taken as the slope to the right. Thus it might help to choose $c_{Bi}$ as $c_{Bi}^{(s_i)}$ if $x_{Bi}$ is expected to increase at the next iteration, or as $c_{Bi}^{(s_i-1)}$ if $x_{Bi}$ is expected to decrease. Several heuristics for predicting the expected behavior of $x_{Bi}$ are described in Part III [9] of this paper.

A final possibility is compromise. If the expected movement of $x_{Bi}$ cannot be usefully predicted, $c_{Bi}$ may be chosen as any value between $c_{Bi}^+$ and $c_{Bi}^-$, such as their average if both are finite. An extension of the above analysis shows that such a choice contributes $(c_{Bi} - c_{Bi}^-)w_{Nj;Bi}$ to the overstatement of one of the nominal reduced costs, and $c_{Bi}^+ - c_{Bi}^-w_{Nj;Bi}$ to the overstatement of the other. The proof of the optimality conditions also extends to nominal reduced costs determined in this way.

3. Prevention of degenerate cycling

It remains to show that the piecewise-linear simplex algorithm can always reach an optimal solution, even though degenerate iterations are unavoidable. This can be done, as in the linear case, by demonstrating a selection rule for the entering and leaving variables that prevents the algorithm from repeatedly visiting a cycle of degenerate bases.

This section examines anti-cycling selection rules of two kinds: those based on perturbation (or equivalent lexicographic tie-breaking) and those based on certain combinatorial analyses. Either kind of rule is sufficient to prevent cycling, but both are of interest for the insight they offer into the algorithm’s behavior.

The P-L anti-cycling rules predictably resemble their linear counterparts, but assume some of the flexibility and generality of the P-L simplex algorithm. An essential aspect of these rules is their unambiguous specification of the basic slopes $c_B$ at degenerate bases. If an arbitrary choice of $c_B$ were allowed at each iteration, then the algorithm could be made to cycle between even a pair of bases.
Cycling can also be prevented by making certain degenerate basis changes according to supplementary criteria, which do not employ the usual reduced costs in selecting the entering variable or the usual ratio test in selecting the leaving variable. The related methods of Balinski and Gomory [1] and of Graves [11] were devised for linear programs, but rely on properties that can be extended to the piecewise-linear case. Rockafellar [16] derives a very general method that can be specialized to P-LPs. Since these methods go beyond the purely primal simplex steps that are the subject of this paper, they are more properly the subject of a future study.

Perturbational and lexicographic selection rules

Since the constraints of a piecewise-linear program have the same form as those of a linear program, they can be perturbed in the familiar way, and to much the same effect. Thus the presentation in this section assumes a knowledge of the usual perturbational arguments [5, 7] and outlines the extension of these arguments to the P-L case.

Given $m$ linearly independent $m$-vectors $v_1, \ldots, v_m$, an $\varepsilon$-perturbed P-LP can be defined by

\[
\begin{align*}
(P_\varepsilon) \quad \text{Minimize} \quad [c/\gamma]x \\
\text{subject to} \quad Ax = b + v_1\varepsilon + v_2\varepsilon^2 + \cdots + v_m\varepsilon^m.
\end{align*}
\]

The original P-LP is $(P_0)$. Any basis matrix $B$ and nonbasic breakpoint vector $\gamma_N$ define a basic solution $\bar{x}(\varepsilon)$ for $(P_\varepsilon)$ that is perturbed from the basic solution $\bar{x}$ for $(P_0)$.

A basis is said to be $\varepsilon$-feasible if $\bar{x}(\varepsilon)$ is feasible for all sufficiently small nonnegative $\varepsilon$; $\varepsilon$-feasibility implies feasibility (taking $\varepsilon = 0$) but the converse need not hold if $\bar{x}$ is degenerate. By a suitable perturbation of the P-LP, however, any feasible basis can be made $\varepsilon$-feasible and, in a certain sense, $\varepsilon$-nondegenerate:

**Property 1.** Given any feasible basis for $(P_0)$, linearly independent vectors $v_1, \ldots, v_m$ may be chosen so that it is $\varepsilon$-feasible.

**Property 2.** For each basic variable $x_{Bi}$ in an $\varepsilon$-feasible basic solution, there exists an index $s_i$ such that $\gamma_{Bi}^{(s_i-1)} < \bar{x}_{Bi}(\varepsilon) < \gamma_{Bi}^{(s_i)}$ for all sufficiently small positive $\varepsilon$.

**Proof.** Property 1 is elementary even in the P-L case. For Property 2, any basic value may be expressed as a polynomial function of $\varepsilon$,

$$
\bar{x}_{Bi}(\varepsilon) = \bar{x}_{Bi} + (\sigma_i v_1) \varepsilon + (\sigma_i \varepsilon^2) + \cdots + (\sigma_i v_m) \varepsilon^m,
$$

where $\sigma_i B = e_{Bi}$ (the unit vector such that $Be_{Bi} = a_{Bi}$). Because the vectors $v_1, \ldots, v_m$ are independent, at least one of the coefficients $(\sigma_i v_i) \neq 0$. Hence $\bar{x}_{Bi}(\varepsilon)$ can equal $\gamma_{Bi}^{(s_i-1)}$ or $\gamma_{Bi}^{(s_i)}$ at only a finite number of $\varepsilon$ values, and the Property follows. \(\square\)
Suppose now that some \( \varepsilon \)-feasible basis is at hand for selected \( v_1, \ldots, v_m \). The key property is the ability of the algorithm to move to a strictly better \( \varepsilon \)-feasible basis:

**Property 3.** Given any \( \varepsilon \)-feasible basis, the P-L simplex algorithm selects entering and leaving variables to produce a new basis that is also \( \varepsilon \)-feasible and that has a lower objective value for all sufficiently small positive \( \varepsilon \).

**Proof.** Consider how the P-L simplex algorithm will behave when applied to any \((P_\varepsilon)\). Property 2 implies that the same vector \( c_B \) of basic slopes may be associated with \( \bar{x}(\varepsilon) \) for all small enough \( \varepsilon \geq 0 \). Since steps (1) through (5) of the algorithm are otherwise unaffected by the perturbation of \( b \), they must be valid for all sufficiently small nonnegative \( \varepsilon \).

In step (6), the "ratios" of interest have the form

\[
\frac{(\bar{x}_{B_i}(\varepsilon) - \gamma_{B_i}^+(\varepsilon))}{y_{B_i}} \quad \text{for } y_{B_i} < 0,
\]

\[
\frac{(\bar{x}_{B_i}(\varepsilon) - \gamma_{B_i}^-(\varepsilon))}{y_{B_i}} \quad \text{for } y_{B_i} > 0,
\]

and \(- (\gamma_{N_p} - \gamma_{N_p}^+(\varepsilon)) \) or \((\gamma_{N_p} - \gamma_{N_p}^-(\varepsilon))\). Extending the proof of Property 2, no two of these ratios can be equal except at a finite number of \( \varepsilon \) values, and hence the ratios lie in the same order for all sufficiently small positive \( \varepsilon \). Then it is not hard to show that any selection of leaving variable is valid for all sufficiently small \( \varepsilon \geq 0 \). Finally, steps (7) and (8) are the same for all such \( \varepsilon \).

Since the selection of entering and leaving variables is valid for all small enough \( \varepsilon \geq 0 \), the new basis is necessarily \( \varepsilon \)-feasible. Furthermore, since the old basic solution was \( \varepsilon \)-nondegenerate, the new basic solution must yield a lower objective value for all small enough \( \varepsilon > 0 \).

The P-L simplex algorithm can thus be guaranteed to optimize the original P-LP when applied to a suitably perturbed variant:

**Theorem 1.** Given a basic feasible solution for \((P_0)\), \( \varepsilon \) and \( v_1, \ldots, v_m \) can be chosen so that the P-L simplex algorithm applied to \((P_\varepsilon)\) terminates after finitely many iterations with an optimal basis (or an indication of unboundedness) for \((P_0)\).

**Proof.** Follows from Properties 1 and 3 by the same arguments used in the linear case.

Also as in the linear case, the behavior of the simplex algorithm for sufficiently small \( \varepsilon \) can be determined without knowing the actual value of \( \varepsilon \). This observation leads to a lexicographic variant outlined below.

Let the coefficients of the polynomial \( \bar{x}_{B_i}(\varepsilon) \) be represented as a vector,

\[
\bar{X}_{B_i} = (\bar{x}_{B_i}, \sigma_1 v_1, \sigma_2 v_2, \ldots, \sigma_m v_m),
\]
and let the distance from this vector to a breakpoint be
\[ X_{B_i} - \gamma_{B_i}^h = (\tilde{x}_{B_i} - \gamma_{B_i}^h, \sigma_1 v_1, \sigma_2 v_2, \ldots, \sigma_m v_m). \]

As previously remarked, the independence of \( v_1, \ldots, v_m \) insures that this vector is not all-zero. Either its first nonzero element is positive (\( \tilde{x}_{B_i} > \gamma_{B_i}^h \)) in the usual notation of lexicographic ordering) or its first nonzero element is negative (\( \tilde{x}_{B_i} < \gamma_{B_i}^h \)).

Using this notation, the proper slope \( c_{B_i} \) for a degenerate \( \tilde{x}_{B_i} = \gamma_{B_i}^h \) in step (1) is given by the following rule:

\[
c_{B_i} = \begin{cases} 
  c_{B_i}^{(r,-1)} & \text{if } \tilde{x}_{B_i} < \gamma_{B_i}^h, \\
  c_{B_i}^{(r)} & \text{if } \tilde{x}_{B_i} > \gamma_{B_i}^h.
\end{cases}
\]

This represents a departure from the linear case, in which lexicographic rules only come into play after the entering variable has been selected.

Once \( c_B \) is chosen, steps (1)-(5) may be carried out as before. In step (6), however, all of the ratios become vectors, with \( \tilde{x}_{B_i} \) substituted for \( \tilde{x}_{B_i} \); as an example,

\[
\frac{X_{B_i} - \gamma_{B_i}^h}{y_{B_i}} = \left( \frac{\tilde{x}_{B_i} - \gamma_{B_i}^h}{y_{B_i}}, \frac{\sigma_1 v_1}{y_{B_i}}, \ldots, \frac{\sigma_m v_m}{y_{B_i}} \right).
\]

Scalar “ratios” are regarded as vectors by extending them with zeroes; for instance, \( \gamma_{N^p} - \gamma_{N^p}^{-r} \) is treated as \( \gamma_{N^p} - \gamma_{N^p}^{-r} \). A leaving variable is then chosen as follows:

(6) Select a leaving variable: Choose any variable \( x_{B_q} \) and breakpoint \( \gamma_{B_q}^{(r)} \) or \( \gamma_{B_q}^{(-r)} \) such that

\[ (a) \quad (\tilde{x}_{B_q} - \gamma_{B_q}^{(r)})/y_{B_q} = 0 \quad \text{and} \quad y_{B_q} < 0, \quad \text{or}
\]

\[ (b) \quad (\tilde{x}_{B_q} - \gamma_{B_q}^{(-r)})/y_{B_q} = 0 \quad \text{and} \quad y_{B_q} > 0; \]

or choose \( x_{N^p} \) and any breakpoint \( \gamma_{N^p}^{(r)} \) or \( \gamma_{N^p}^{(-r)} \) such that

\[ (c) \quad - (\gamma_{N^p} - \gamma_{N^p}^{(r)}) = 0 \quad \text{and} \quad d_{N^p}^+ < 0, \quad \text{or}
\]

\[ (d) \quad (\gamma_{N^p} - \gamma_{N^p}^{(-r)}) = 0 \quad \text{and} \quad d_{N^p}^- > 0; \]

provided \( \tilde{0} \) is lexicographically small enough that \( d_{N^p}^- < 0 \) and

\[
d_{N^p}^+ (\tilde{0}) = c_{N^p}^{(r)} - \sum_{y_{B_i} < 0} c_{B_i}^{(r,-1)} y_{B_i} - \sum_{y_{B_i} > 0} c_{B_i}^{(r,-1)} y_{B_i} < 0,
\]

or \( \tilde{0} \) is lexicographically small enough that \( d_{N^p}^- > 0 \) and

\[
d_{N^p}^- (\tilde{0}) = c_{N^p}^{(r)} + \sum_{y_{B_i} < 0} c_{B_i}^{(r,-1)} y_{B_i} + \sum_{y_{B_i} > 0} c_{B_i}^{(r,-1)} y_{B_i} > 0;
\]

where the indices \( r_i \) (for \( i \neq q \)) and \( r \) satisfy

\[ (\tilde{x}_{B_i} - \gamma_{B_i}^{(r,-1)})/y_{B_i} < \tilde{0} < (\tilde{x}_{B_i} - \gamma_{B_i}^{(r)})/y_{B_i} \quad \text{for} \ y_{B_i} < 0, \]

\[ (\tilde{x}_{B_i} - \gamma_{B_i}^{(-r,-1)})/y_{B_i} < \tilde{0} < (\tilde{x}_{B_i} - \gamma_{B_i}^{(-r)})/y_{B_i} \quad \text{for} \ y_{B_i} > 0, \]

\[ -(\gamma_{N^p} - \gamma_{N^p}^{(r-1)}) < \tilde{0} < -(\gamma_{N^p} - \gamma_{N^p}^{(r)}) \quad \text{if} \ d_{N^p}^- < 0, \]

\[ (\gamma_{N^p} - \gamma_{N^p}^{(r-1)}) < \tilde{0} < (\gamma_{N^p} - \gamma_{N^p}^{(r)}) \quad \text{if} \ d_{N^p}^- > 0. \]
Steps (7) and (8) can then proceed as before, taking $\bar{\theta}$ as the first element of $\bar{\Theta}$.

**Combinatorial selection rules**

The combinatorially motivated anti-cycling rules of Bland [3] also extend to piecewise-linear programming. Following some preliminary development of ideas below, a piecewise-linear equivalent of Rule I in [3] is stated and proved in detail, and the features of a P-L Rule II are outlined.

**Preliminaries.** Imagine that, prior to the start of some iteration, all slopes of $[c/\gamma]_k$ are reduced by $\gamma a_k$, for $k = 1, \ldots, n$ and an arbitrary vector $v$. The new objective can be expressed as

$$[\tilde{c}/\gamma]x = [c/\gamma]x - (vA)x.$$ 

Over all $x$ that satisfy $Ax = b$, the two objectives differ only by a constant $vb$, and hence have the same minimizers. Indeed, the change has little effect on the quantities $c_B$, $d_N$, and $d_N$ that are computed in steps (1)-(3). Since $[\tilde{c}/\gamma]$ has the same breakpoints as $[c/\gamma]$, the basic slopes can become $\tilde{c}_B = c_B - vb$. Then $\tilde{\pi}B = \tilde{c}_B$ implies $\tilde{\pi} = \pi - v$. The reduced costs are

$$\tilde{d}^+ = \tilde{c}^{+\{1\}} - \pi a_N,$$

and similarly for $\tilde{d}^-$. They are all unchanged.

Of particular interest is the special case in which $v$ is taken to be $\pi$ itself. In the resulting canonical objective $[\tilde{c}/\gamma]$, $\tilde{\pi}a_B = c_B$ is subtracted from every slope of $[c/\gamma]_B$, and $\tilde{\pi}a_N$ is subtracted from every slope of $[c/\gamma]_N$. Thus $[\tilde{c}/\gamma]$ has slopes as follows:

$$\tilde{c}_B = 0,$$

$$\tilde{c}_B = c_B - c_B > 0,$$

$$\tilde{c}_B = c_B - c_B < 0,$$

$$\tilde{c}^{+\{1\}} = d_N^+,$$

$$\tilde{c}_N = (c_B - c_B) > d_N^+,$$

$$\tilde{c}_N = d_N^-,$$

$$\tilde{c}_N = d_N^- < d_N^-.$$ 

From (3.1)-(3.3), the slopes of $[\tilde{c}/\gamma]_B$ are nonpositive at all points to the left of $\bar{x}_B$ and are nonnegative at all points to the right. Similarly, from (3.4)-(3.7), if $x_N$ is ineligible to enter the basis (because $d_N^+ \geq 0$ and $d_N^- \leq 0$) then the slopes of $[\tilde{c}/\gamma]_N$ are nonpositive at all points to the left of $\bar{x}_N$ and are nonnegative at all points to the right.
Finally, consider a series of degenerate iterations; let $x_{Nl}$ be the variable that enters the basis at the first iteration of the series, and let $[\bar{c}/\gamma]$ be the canonical objective relative to the $\bar{\pi}$ computed at the first iteration. Since the degenerate iterations leave $\bar{x}$ unchanged, the above observations about the slopes to the left and the right of $\bar{x}$ remain true.

Moreover, if $x_{Bi}$ has been basic throughout the degenerate series and $c_{Bi}$ has not been changed, then (3.1) says that $\bar{c}_{Bi}$ remains zero. Similarly, if $x_{Nl}$ entered with $d_{Nl}^+ < 0$ and if $c_{Nl}$ was fixed at $c_{Nl}^{+[1]}$, then (3.4) implies that $\bar{c}_{Bi}$ remains negative at subsequent iterations. If $x_{Nl}$ entered with $d_{Nl}^- > 0$ and if $c_{Nl}$ was fixed at $c_{Nl}^{-[1]}$, then (3.6) implies that $\bar{c}_{Bi}$ remains positive.

**Rule I.** A P-L extension of Bland’s first rule in [3] can be stated as follows. The entering variable $x_{Np}$ must be the first one available:

$$p = \min \{j : d_{Nj}^- < 0 \text{ or } d_{Nj}^+ > 0\}.$$  \hfill (3.8)

Thus the nominal reduced cost for $x_{Np}$ must be favorable:

$$d_{Np}^+ = c_{Np}^{+[1]} - \sum_i c_{Bi} y_{Bi} < 0 \quad (By_B = a_{Np}), \quad \text{or}$$

$$d_{Np}^- = c_{Np}^{-[1]} + \sum_i c_{Bi} y_{Bi} > 0 \quad (By_B = -a_{Np}).$$  \hfill (3.9)

If the true reduced cost $d_{Np}^+(\delta)$ or $d_{Np}^-(\delta)$ is also favorable, a leaving variable may be chosen in the usual way. Otherwise, writing $c^-_{Bi}$, $c^+_{Bi}$ for the slopes of $[c/\gamma]_{Bi}$ to the left and right of $\bar{x}_{Bi}$ as in Section 2,

$$d_{Np}^+(\delta) = c_{Np}^{+[1]} - \sum_{y_{Bi} < 0} c_{Bi} y_{Bi} - \sum_{y_{Bi} > 0} c_{Bi} y_{Bi} \geq 0, \quad \text{or}$$

$$d_{Np}^-(\delta) = c_{Np}^{-[1]} + \sum_{y_{Bi} < 0} c_{Bi} y_{Bi} + \sum_{y_{Bi} > 0} c_{Bi} y_{Bi} \leq 0.$$  \hfill (3.10)

In this situation, the rule requires that a degenerate leaving variable $x_{Bq}$ be chosen “early enough” in the following sense:

either $c_{Bq} = c_{Bq}^-$ and $y_{Bq} < 0$,

or $c_{Bq} = c_{Bq}^+$ and $y_{Bq} > 0$; and

$$d_{Np:Bq}^+(\delta) = c_{Np}^{+[1]} - \sum_{i \leq q, y_{Bi} < 0} c_{Bi} y_{Bi} - \sum_{i > q, y_{Bi} > 0} c_{Bi} y_{Bi} \leq 0, \quad \text{or}$$

$$d_{Np:Bq}^-(\delta) = c_{Np}^{-[1]} + \sum_{i \leq q, y_{Bi} < 0} c_{Bi} y_{Bi} + \sum_{i > q, y_{Bi} > 0} c_{Bi} y_{Bi} \geq 0.$$  \hfill (3.11)

Steps (7) and (8) then proceed as before, but updating $c_B$ as follows:

$$c_{Bi} \text{ unchanged for } i > q,$$

$$c_{Np} = c_{Np}^{+[1]} \text{ if } d_{Np}^+ < 0,$$

$$c_{Np} = c_{Np}^{-[1]} \text{ if } d_{Np}^- > 0.$$  \hfill (3.12)

For $i < q$, the algorithm may set $c_{Bi}$ as before.
The nature of the leaving-variable part of the rule may be seen more clearly by
defining the set of indices of "wrong" slopes in $c_B$:
\[
\Omega = \{ i : \bar{x}_n = \gamma_i^{(s)}, \text{ and either } c_{B_i} = c^{-}_{B_i} \text{ but } y_{B_i} < 0, \text{ or } c_{B_i} = c^{+}_{B_i} \text{ but } y_{B_i} > 0 \}.
\]
A comparison of terms in (3.9) and (3.10) shows that both can hold only if $\Omega$ is
nonempty. Then, comparing (3.9) and (3.11), a degenerate variable $x_{Bq}$ is seen to
be early enough provided that
\[
q \in \Omega \text{ and } d_{Np,Bq}^+(\delta) = d_{Np}^+ + \sum_{i < q, t \in \Omega} (c_i^{(x)} - c_i^{(x-1)}) |y_{B_i}| \leq 0, \text{ or }
\]
\[
d_{Np,Bq}^-(\delta) = d_{Np}^- - \sum_{i < q, t \in \Omega} (c_i^{(x)} - c_i^{(x-1)}) |y_{B_i}| \geq 0.
\]
In words, for $x_{Bq}$ to leave its slope must be "wrong" in $c_B$, but its reduced cost
must still be favorable or neutral when all lower-numbered slopes are "corrected".
This criterion is always satisfied by the lowest-numbered $q \in \Omega$, and may be satisfied
by other early enough indices $q$.

**Theorem 2.** If the piecewise-linear simplex algorithm selects the entering variable by
criterion (3.8) and the leaving variable by criterion (3.11), and updates $c_B$ by formula
(3.12), then it must terminate after finitely many iterations.

**Proof.** Suppose that entering and leaving variables are chosen as hypothesized, but
that nevertheless some basis is repeated after a cycle of degenerate iterations.
Following the reasoning in [3], let $x_l$ be the highest-numbered variable that both
enters and leaves the basis in the course of the cycle. Suppose that $x_{Nl}$ enters with
$d^+_{Nl} < 0$; the case of $d^-_{Nl} > 0$ is entirely analogous.

Let $[\gamma / \gamma]$ be the canonical objective relative to the $\pi$ computed when $x_{Nl}$ is
chosen to enter, and consider the subsequent iteration at which $x_{Bl}$ leaves
and some $x_{Np}$ enters. Several assertions follow from the fact that $x_l$ is the highest-numbered
in the cycle, together with the entering-variable and updating rules and the previous
observations about $[\gamma / \gamma]$.

- $x_{Np}$: Since the $p$th variable participates in the cycle, $p < l$. By the entering-
  variable rule, when $x_{Nl}$ entered, the $p$th variable either was basic or was nonbasic
  and ineligible to enter. Thus the slope of $[\gamma / \gamma]_{Np}$ is nonpositive at all points
to the left of $x_{Np}$ and nonnegative at all points to the right.
- $x_{Bi}, l < i$: By the entering-variable rule, when $x_{Nl}$ entered, the $i$th variable either
  was basic or was nonbasic and ineligible to enter. Thus the slope of $[\gamma / \gamma]_{Bi}$ is
  nonpositive at all points to the left of $x_{B^i}$ and nonnegative at all points to the right.
- $x_{Bl}$: Because the $l$th variable is the highest numbered in the cycle, the updating
  rule insures that $c_{Nl}$ was set to $c^{+}_{Nl}^{(1)}$ when $x_{Nl}$ entered the basis, and that $c_{Bl}$
  was not changed at subsequent iterations. Thus $\tilde{c}_{Bl}$ remains negative.
• $x_{Bl}, i > l$: These variables do not participate in the cycle, and so must remain basic throughout it. By the updating rule, $c_{Bi}$ is left unchanged at every iteration in the cycle. Thus $\hat{c}_{Bi}$ remains zero.

In sum, $[\hat{c}/\gamma]$ has the following properties at the iteration where $x_{Bl}$ leaves:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Slope in $[\hat{c}/\gamma]$ satisfies...</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_{Np}$</td>
<td>$\hat{c}<em>{Np}^{\pm} \leq 0$, $\hat{c}</em>{Np}^{\mp} \geq 0$</td>
</tr>
<tr>
<td>$x_{Bi}, i &lt; l$</td>
<td>$c_{Bi}^{\pm} &lt; 0$, $c_{Bi}^{\mp} \geq 0$</td>
</tr>
<tr>
<td>$x_{Bi}$</td>
<td>$\hat{c}_{Bi} &lt; 0$</td>
</tr>
<tr>
<td>$x_{Bi}, i &gt; l$</td>
<td>$\hat{c}_{Bi} = 0$</td>
</tr>
</tbody>
</table>

It remains only to observe that (3.13) is unaffected by substituting $[\hat{c}/\gamma]$ for $[c/\gamma]$, since neither the reduced costs $d_{Np}^+$, $d_{Np}^-$ nor the differences $c_{Bi}^{(x)} - c_{Bi}^{(y+1)}$ are changed. Thus, to satisfy the leaving-variable rule, the iteration must have $y_{Bi} > 0$ (since $\hat{c}_{Bi}$ was fixed at $c_{Bi}$) and

$$d_{Np; Bi}(\delta) = \hat{c}_{Np}^{\pm} - \sum_{i < l, y_{Bi} < 0} \hat{c}_{Bi}^{\pm} y_{Bi} - \sum_{i < l, y_{Bi} > 0} \hat{c}_{Bi}^{\mp} y_{Bi} - \sum_{i > l} \hat{c}_{Bi}^{\pm} y_{Bi} \leq 0,$$

or

$$d_{Np; Bi}(\delta) = \hat{c}_{Np}^{\pm} + \sum_{i < l, y_{Bi} < 0} \hat{c}_{Bi}^{\pm} y_{Bi} + \sum_{i < l, y_{Bi} > 0} \hat{c}_{Bi}^{\mp} y_{Bi} + \sum_{i > l} \hat{c}_{Bi}^{\pm} y_{Bi} \geq 0.$$

Yet every term for $d_{Np; Bi}(\delta)$ is nonnegative, and specifically $-\hat{c}_{Bi} y_{Bi} > 0$; every term for $d_{Np; Bi}(\delta)$ is nonpositive, and $\hat{c}_{Bi} y_{Bi} < 0$. Thus the assumption of a degenerate cycle leads to a contradiction. 

**Rule II.** A piecewise-linear version of Bland's second rule can also be demonstrated by reference to the canonical objective. Suppose that $x_{Np}$ is selected to enter at the basic solution $\bar{x}$ defined by $B$ and $\gamma_N$, and let $\mathcal{F}^+ = \{ j \neq p : d_{Nj}^+ < 0 \}$ and $\mathcal{F}^- = \{ j \neq p : d_{Nj}^- > 0 \}$ be the sets of other nonbasic variables that could have entered. The key to Rule II is the following P-L extension of Observation 2.1 in [3]:

**Property 4.** If a series of iterations takes the P-L simplex algorithm from the basic solution $\bar{x}$ to a basic solution $x'$ such that $x'_{Nj} \leq \gamma_{Nj}$ for all $j \in \mathcal{F}^+$ and $x'_{Nj} \geq \gamma_{Nj}$ for all $j \in \mathcal{F}^-$, then $x_{Np} \geq \gamma_{Np}$ (if $d_{Np}^+ < 0$) or $x_{Np} \leq \gamma_{Np}$ (if $d_{Np}^- > 0$).

**Proof.** Let $[\hat{c}/\gamma]$ be the canonical objective relative to the vector $\bar{\pi}$ computed when $\bar{x}$ was the current solution and $x_{Np}$ was chosen to enter. From (3.1)–(3.7), the slopes of individual terms $[\hat{c}/\gamma]_k$ in the neighborhood of $\bar{x}_k$ must have the following signs:
Suppose that \( x'_{Nj} \leq \gamma_{Nj} \) for all \( j \in \mathcal{F}^+ \) and \( x'_N \geq \gamma_N \) for all \( j \in \mathcal{F}^- \), as hypothesized, but that \( x'_{Np} < \gamma_{Np} \) (if \( d^+_{Np} < 0 \)) or \( x'_N > \gamma_N \) (if \( d^-_{Np} > 0 \)). Since \( \bar{x}_{Nj} = \gamma_{Nj} \), and particularly \( \bar{x}_{Np} = \gamma_{Np} \), the above table implies that \( [\hat{c}/\gamma]x' \geq [\hat{c}/\gamma] \bar{x} \). Since the canonical objective differs from the true objective by only a constant, it follows that \( [c/\gamma]x' \geq [c/\gamma] \bar{x} \), contradicting the assumption that \( x' \) follows by a series of iterations from \( \bar{x} \). □

To make use of this property, imagine that when \( x_{Np} \) is brought into the basis, \( [c/\gamma] \) is replaced by a simplified objective \( [\hat{c}/\hat{\gamma}] \):

\[
\hat{c}^{+(1)}_{Nj} = +\infty \quad \text{for } j \in \mathcal{F}^+, \quad \hat{c}^{-(1)}_{Nj} = -\infty \quad \text{for } j \in \mathcal{F}^-; \tag{3.14}
\]

\[
\hat{\gamma}_{Np} = -\infty \quad \text{if } d^+_{Np} < 0, \quad \hat{\gamma}_{Np} = +\infty \quad \text{if } d^-_{Np} > 0. \tag{3.15}
\]

The restrictions (3.14) force all subsequent basic feasible solutions \( x' \) to satisfy the hypothesis of Property 4. As a result, the parts of the objective where \( x'_{Np} < \gamma_{Np} \) (if \( d^+_{Np} < 0 \)) or \( x'_{N} > \gamma_N \) (if \( d^-_{Np} > 0 \)) can be ignored, which is exactly the effect of the changes specified by (3.15). Every feasible basis for \( [\hat{c}/\hat{\gamma}] \) is also a feasible basis for \( [c/\gamma] \), and the minimization of \( [\hat{c}/\hat{\gamma}] \) subject to \( Ax = b \) is a restricted subproblem of the original P-LP.

Suppose that the P-L simplex algorithm terminates finitely when applied to \( [\hat{c}/\hat{\gamma}] \) starting from \( \bar{x} \). Either it finds an unbounded solution, in which case \( [c/\gamma] \) is unbounded as well, or it finds a basic optimal solution \( \hat{x} \). In the latter case \( \hat{x} \) is also a basic feasible solution for \( [c/\gamma] \), and \( \hat{x}_{Nj} \) has unfavorable reduced costs relative to \( [c/\gamma] \) for all nonbasic \( j \notin \mathcal{F}^+ \cup \mathcal{F}^- \). Thus a new entering variable \( x_{Np} \) can be chosen, and a new restricted subproblem can be constructed, so that the union of the new \( \mathcal{F}^+ \) and \( \mathcal{F}^- \) is strictly smaller than before. After only finitely many such subproblems, \( \mathcal{F}^+ \) and \( \mathcal{F}^- \) must be empty and an optimal solution must be at hand.

The same approach may be applied recursively to solve any subproblem. At each level of recursion, the number of finite breakpoints is strictly reduced, so that the number of recursion levels—as well as the number of subproblems at each level—must be finite. Hence, unless unboundedness is detected, the top-level iterations must eventually stop at an optimal basis.
By construction, every basic feasible solution for a subproblem at any level is also a basic feasible solution for the problem at the next higher level. Thus any basic feasible solution for any subproblem must be a basic feasible solution for \([c/\gamma]\). In fact a careful analysis can show, as in [3], that every iteration on a subproblem can be interpreted as a valid iteration on the full P-LP. Hence the recursion effectively describes a cycle-avoiding rule for the P-L simplex algorithm.

4. Feasibility

Like other simplex methods, the piecewise-linear simplex algorithm requires a basic feasible solution for a starting point. Fortunately, the existence of such solutions is a reasonable assumption for P-LPs, just as for LPs.

A P-LP can fail to have basic feasible solutions because it has no basic solutions at all. The difficulties of this situation are easily circumvented, as observed in the first half of this section, by implicitly or explicitly dropping certain linearly dependent columns of the constraint matrix.

Once a basic solution has been constructed, the P-L simplex algorithm can find a basic feasible solution — or determine that no feasible solutions of any kind exist — by following either a “two-phase” or a “penalty” approach as described in the second half of this section. Both approaches can be viewed as extensions of common linear-programming techniques that replace or supplement the true objective with a simple piecewise-linear function. Successful application of the penalty approach requires that certain sufficiently large penalty values be chosen; the optimality conditions for P-LPs provide a simple demonstration that the minimum sizes of these penalties are determined by the dual values \(\bar{\pi}\) at the optimal basis.

Finding basic solutions

If every piecewise-linear term \([c/\gamma]\)_k has at least one finite breakpoint, then a basic (though possibly infeasible) solution is easily constructed. First, any \(m\) linearly independent columns \(a_B\) of \(A\) are chosen to define the basic variables. Then the remaining nonbasic variables are fixed at arbitrary finite breakpoints \(\gamma_{N_j}\).

If some P-L terms do not have finite breakpoints, then the corresponding free variables can only be basic. Hence bases can exist, and the above construction can be carried through, if and only if the coefficient columns of the free variables are linearly independent. P-LPs that violate this requirement need not be contrived; as an example, consider the linear \(l_1\) estimation problem defined for an \(m \times n\) matrix \(G\) and \(m\)-vector \(g\) as follows:

\[
\text{Minimize } \sum_{i=1}^{m} |u_i| \\
\text{subject to } Gx + u = g.
\]
The variables $u_1, \ldots, u_m$ have breakpoints at zero, but $x_1, \ldots, x_n$ have a slope of zero and no finite breakpoints. Thus basic solutions do not exist if the columns of $G$ are linearly dependent.

In keeping with the notation for basic and nonbasic variables, let the free variables be $x_{Fi}$, and let $c_F$ be the one finite slope defining $[c/\gamma]_{Fi}$. Let $F$ be the matrix of columns $a_{Fi}$ that correspond to the free variables. The following elementary relationship between $F$ and $c_F$ carries over from linear programming:

**Property 5.** If a P-LP has a finite optimal value, then there exists a vector $\pi$ such that $\pi F = c_F$.

**Proof.** If no such $\pi$ exists, then $c_F$ cannot be in the row space of $F$. Hence $c_F$ cannot be orthogonal to the nullspace of $F$: there must exist a vector $\tilde{y}_F$ such that $F\tilde{y}_F = 0$ but $c_F\tilde{y}_F \neq 0$. From the existence of $\tilde{y}_F$ the P-LP is easily seen to be unbounded if it has any feasible solutions at all, contradicting the hypothesis. ∎

If the columns of $F$ are not linearly independent, a maximal independent subset can be chosen. Let $x_{F_{Bi}}$ denote the variables corresponding to any such subset, and let $x_{F_{Nj}}$ denote the remaining free variables. Thus for any values $\tilde{y}_{F_{Nj}}$ that might be chosen, there are unique values $\tilde{y}_{F_{Bi}}$ such that $F\tilde{y}_F = 0$. In these terms, a converse to the above property, can be stated as follows:

**Property 6.** Suppose there does exist $\pi$ such that $\pi F = c_F$, and let $x_{F_{Nj}}^*$ be any arbitrary values for the nonindependent free variables $x_{F_{Nj}}$. If the P-LP has a finite minimum, then it has an optimal solution in which $x_{F_{Nj}} = x_{F_{Nj}}^*$. If the P-LP has an unbounded solution (or no feasible solution) then it continues to have an unbounded solution (or no feasible solution) subject to the additional constraints $x_{F_{Nj}} = x_{F_{Nj}}^*$.

**Proof.** If there are no feasible solutions, then there can be no feasible solutions with $x_{F_{Nj}} = x_{F_{Nj}}^*$. Otherwise, let $\bar{x}$ be any feasible solution to $Ax = b$, and let $\bar{y}_F$ be the solution to $F\bar{y}_F = 0$ such that

$$\bar{y}_{F_{Nj}} = x_{F_{Nj}}^* - \bar{x}_{F_{Nj}}.$$

Extend $\bar{y}_F$ to a $\bar{y}$ that solves $Ay = 0$, by letting $\bar{y}_k = 0$ for all $x_k$ that are not free variables. Then a new solution to $Ax = b$ is given by

$$x' = \bar{x} + \bar{y}.$$

Since $x'$ differs from $\bar{x}$ only in the free variables, $x'$ is also feasible. Also $x_{F_{Nj}}' = \bar{x}_{F_{Nj}} + \bar{y}_{F_{Nj}} = x_{F_{Nj}}^*$, and $cx' = c\bar{x} + c_F\bar{y}_F$.

Since $c_F$ lies in the row space of $F$ by hypothesis, and since $\bar{y}_F$ is in the nullspace, $c_F\bar{y}_F = 0$, from which it follows that $cx' = c\bar{x}$. Thus, for every feasible solution $\bar{x}$, there is an associated feasible solution $x'$, achieving the same objective value, in
which the free variables $x_{F_{Nj}}$ take the values $x^*_{F_{Nj}}$. The assertions of the property are a direct consequence. 

For practical purposes, Gaussian elimination on $F$ can serve both to construct a maximal independent subset of columns $a_{FBi}$ and to determine whether $\pi F = c_F$ can be satisfied. If $\pi$ exists then the variables $x_{F_{Nj}}$ can be fixed at any values (most conveniently, zero) and the resulting contracted P-LP may be solved in place of the original one.

Alternatively, each objective term $[c/\gamma]_{Fi}$ may be transformed to an equivalent P-L function that has an arbitrary “artificial” breakpoint $\gamma^*_{Fi}$ surrounded by equal slopes:

$$
\begin{align*}
\gamma^{0}_{Fi} &= -\infty \\
c^{0}_{Fi} &= c_{Fi} \\
\gamma^{1}_{Fi} &= \gamma^*_{Fi} \\
c^{1}_{Fi} &= c_{Fi} \\
\gamma^{2}_{Fi} &= +\infty
\end{align*}
$$

To define an initial basic solution, some of the $x_{Fi}$ may be made nonbasic at their artificial breakpoints. Once any iteration brings $x_{Fi}$ into the basis, however, its breakpoint may be ignored at all subsequent iterations.

When artificial breakpoints are employed in this way, only an independent subset of the columns $a_{Fi}$ can ever enter the basis. Hence this approach effectively contracts the P-LP, by fixing free variables at arbitrary values $\gamma^*_{Fi}$, although it does not decide in advance which variables to fix. (A specialization of this idea has long been used in implementations of the bounded-variable linear simplex algorithm, to handle variables that have a lower bound of $-\infty$ and an upper bound of $+\infty$; the artificial breakpoint is customarily taken to be zero.)

The vector $\pi$ determined in step (1) of the P-L simplex algorithm must satisfy $\pi a_{Fi} = c_{Fi}$ for all basic free variables $x_{Fi}$. If the optimality conditions in step (2) are satisfied, then also $c_{Fi} \leq \pi a_{Fi} \leq c_{Fi}$ for all $x_{Fi}$ that are nonbasic at their artificial breakpoints. Thus, in finding a finite optimum, the algorithm automatically determines a solution to the condition $\pi F = c_F$ of Property 5.

**Finding basic feasible solutions**

Finding a feasible basic solution may or may not be more difficult than finding just a basic solution. If $[c/\gamma]$ is finite everywhere then all basic solutions must be feasible. On the other hand, if any finite breakpoint $\gamma^{(h)}_k$ is adjacent to an infinite slope $c^{(h-1)}_k = -\infty$ or $c^{(h)}_k = +\infty$ then many basic solutions may be infeasible, and no basic feasible solution may be evident.

In the latter case, the piecewise-linear simplex algorithm may itself be used to search for a basic feasible solution from a basic infeasible starting point. A simple
"two-phase" approach defines a new P-L function $[f/\gamma]$ whose slopes are derived from those of $[c/\gamma]$ as follows:

$$
c^{(s)}_k = -\infty \Rightarrow f^{(s)}_k = -1, $n$
c^{(t)}_k = +\infty \Rightarrow f^{(t)}_k = +1, $n$
c^{(h)}_k \text{ finite} \Rightarrow f^{(h)}_k = 0.
$$

Each term $[f/\gamma]_k x_k$ is finite everywhere. Thus any basic solution to $Ax = b$ can be used by the P-L simplex algorithm as a starting point for the "phase one" P-LP,

$$
\text{Minimize } [f/\gamma]x \\
\text{subject to } Ax = b.
$$

Since $[f/\gamma]_k x_k$ either becomes constant or increases to infinity as $x_k \rightarrow -\infty$ or $x_k \rightarrow +\infty$, this problem cannot be unbounded. The algorithm must find a basic optimal solution, and the following result applies:

**Property 7.** If any optimal solution for $[f/\gamma]$ is infeasible for $[c/\gamma]$, then every solution to $Ax = b$ is infeasible for $[c/\gamma]$.

**Proof.** Each of the terms $[f/\gamma]_k$ is minimal precisely on the interval where $[c/\gamma]_k x_k$ is finite. Hence if $\bar{x}$ and $\hat{x}$ are a feasible and an infeasible solution for $[c/\gamma]$, then $\hat{x}$ must be optimal for $[f/\gamma]$, with $[f/\gamma] \hat{x} < [f/\gamma] \bar{x}$. In other words, the infeasible $\hat{x}$ cannot be optimal for $[c/\gamma]$ unless no feasible $\bar{x}$ exists. □

Property 7 implies that, if there does exist a feasible solution for $[c/\gamma]$, then the P-L simplex algorithm must stop at a basic optimal solution for $[f/\gamma]$ that is also basic feasible for $[c/\gamma]$. Then a starting point is at hand for the minimization of $[c/\gamma]$ (in "phase two").

An alternative "penalty" approach replaces the infinite slopes in $[c/\gamma]$ by large finite ones:

$$
c^{(s)}_k = -\infty \Rightarrow \hat{f}^{(s)}_k = -M^-_k, $n$
c^{(t)}_k = +\infty \Rightarrow \hat{f}^{(t)}_k = +M^+_k, $n$
c^{(h)}_k \text{ finite} \Rightarrow \hat{f}^{(h)}_k = c^{(h)}_k.
$$

The resulting P-L function $[\hat{f}/\gamma]$ is convex for all sufficiently large $M^-_k$ and $M^+_k$. Again it is everywhere finite, so that the P-L simplex algorithm can minimize it, subject to $Ax = b$, starting from any initial basic solution.

**Property 8.** If there exists a feasible solution for $[c/\gamma]$, and if $M^-_k$ and $M^+_k$ are chosen sufficiently large for all $k$, then a basic solution is optimal for $[c/\gamma]$ if and only if it is optimal for $[\hat{f}/\gamma]$. 
Proof. Let $\bar{x}$ be any feasible solution and $\bar{x}$ an infeasible solution for $[c/\gamma]$. If the function $[\hat{f}/\gamma]$ is varied by letting $M_k \to -\infty$ and $M_k^+ \to +\infty$ for all $k$, then $[\hat{f}/\gamma] \bar{x} \to \infty$ while $[\hat{f}/\gamma] \bar{x}$ remains unchanged. Thus $\bar{x}$ cannot be optimal for $[\hat{f}/\gamma]$ if $M_k^-$ and $M_k^+$ are large enough. Since the number of basic infeasible solutions is finite, $M_k^-$ and $M_k^+$ can be chosen sufficiently large so that every basic optimal solution for $[\hat{f}/\gamma]$ is feasible for $[c/\gamma]$.

Suppose that $M_k^-$ and $M_k^+$ are so chosen; then it need only be shown that a basic feasible solution $x^*$ for $[c/\gamma]$ is optimal for $[c/\gamma]$ if and only if it is optimal for $[\hat{f}/\gamma]$. Imagine applying the P-L simplex algorithm to either $[c/\gamma]$ or $[\hat{f}/\gamma]$ at $x^*$. The basic slopes $c_B$ and $f_B$ may be chosen identically, so that the same $\pi$ is defined by step (1). The optimality conditions of step (2) may differ, if any nonbasic variables lie at breakpoints adjacent to infinite slopes:

$$
\begin{align*}
\text{If...} & & \text{then optimality for} & & \text{but optimality for} \\
 c_{Nj}^{(s)} = -\infty & & [c/\gamma] \text{ requires...} & & [\hat{f}/\gamma] \text{ requires...} \\
 \gamma_{Nj} = \gamma_{Nj}^{(s+1)} & & \bar{\pi} a_{Nj} \leq c_{Nj}^{(s+1)} & & -M_{Nj}^- \leq \bar{\pi} a_{Nj} \leq c_{Nj}^{(s+1)} \\
 c_{Nj}^{(t)} = +\infty & & \pi a_{Nj} \leq c_{Nj}^{(t-1)} & & +M_{Nj}^+ \geq \bar{\pi} a_{Nj} \geq c_{Nj}^{(t-1)} \\
 \gamma_{Nj} = \gamma_{Nj}^{(t)} & & & & \\
\end{align*}
$$

The optimality conditions are the same, however, for all other nonbasic variables.

Suppose first that $x^*$ is optimal for $[c/\gamma]$. Thus there is some $\bar{\pi}$ such that the optimality conditions for $[c/\gamma]$ are satisfied. The above comparison shows clearly that the optimality conditions for $[\hat{f}/\gamma]$ will also be satisfied provided that $M_{Nj}^-$ and $M_{Nj}^+$ are chosen large enough for all nonbasic $x_{Nj}$. Since there can be only finitely many basic optimal solutions, $M_k^-$ and $M_k^+$ can be chosen large enough for all $k$ so that every basic optimal $x^*$ for $[c/\gamma]$ is optimal for $[\hat{f}/\gamma]$.

Conversely, suppose that $x^*$ is not optimal for $[c/\gamma]$. Then for any $\bar{\pi}$ computed in step (1), some optimality condition for $[c/\gamma]$ is violated in step (2). However, as the above comparison of conditions makes clear, every optimality condition for $[c/\gamma]$ is also an optimality condition for $[\hat{f}/\gamma]$. Hence $x^*$ cannot be optimal for $[\hat{f}/\gamma]$. □

A finite minimum for $[c/\gamma]$ can thus be found, if any exists, by minimizing $[\hat{f}/\gamma]$ with sufficiently large penalties $M_k^-$ and $M_k^+$. Clearly, moreover, if there is no finite minimum for $[c/\gamma]$ then there can be none for $[\hat{f}/\gamma]$ with any choice of penalties. (If the P-LP has no feasible solutions for $[c/\gamma]$, however, it must still have feasible solutions for $[\hat{f}/\gamma]$ no matter how large the penalties. Only the two-phase approach can prove an absence of feasible points.)
A precise characterization of sufficiently large penalties can be given in terms of any optimal basis:

**Property 9.** If a basic solution is optimal for \([c/\gamma]\), then it is optimal for \([\hat{f}/\gamma]\) provided that

\[-M_k \leq \min\{\pi^* a_k, c_k^{(s+1)}\} \quad \text{and} \quad +M_k^+ \geq \max\{\pi^* a_k, c_k^{(t-1)}\},\]

where \(\pi^*\) is the vector of dual values from step \((1)\), \(c_k^{(s+1)}\) is the smallest slope of \([c/\gamma]_k\) greater than \(-\infty\), and \(c_k^{(t-1)}\) is the largest slope less than \(+\infty\).

**Proof.** For a basic variable \(x_{Bi}\), \(\pi^* a_{Bi}\) equals some slope \(c_{Bi}\), and the conditions of the property merely ensure convexity of \([\hat{f}/\gamma]_{Bi}\). For a nonbasic variable \(x_{Nj}\) that lies at a breakpoint adjacent to two finite slopes, optimality requires \(c_{Nj}^{(s+1)} \leq \pi^* a_{Nj} \leq c_{Nj}^{(t-1)}\), so again the conditions just ensure convexity.

For a nonbasic \(x_{Nj}\) that lies adjacent to \(c_{Nj}^{(s)} = -\infty\), optimality for \([c/\gamma]\) requires \(\pi^* a_{Nj} \leq c_{Nj}^{(s+1)}\). Thus the conditions of the property imply \(-M_k \leq \pi^* a_{Nj} \leq c_{Nj}^{(s+1)}\), which is exactly the condition for optimality of \([\hat{f}/\gamma]\) as shown in the proof of Property 8. The argument for an \(x_{Nj}\) that lies adjacent to \(c_{Nj}^{(t)} = +\infty\) is analogous. \(\square\)

Property 9 can be regarded as a special case of a fundamental result for exact penalty functions [12, pages 387-390]. Although it is customary to think of \(M_k^-\) and \(M_k^+\) as large numbers, the conditions of the property require only that they be “large enough”; even negative values may suffice for some of them.

Since \(\pi^*\) cannot be known in advance, the choice of \(M_k^-\) and \(M_k^+\) in practice must be based on an estimate of \(\pi^*\), or of \(\pi^* a_{Nj} = c_{Nj}^{(s+1)} - d_{Nj}^{(s)} = c_{Nj}^{(t-1)} - d_{Nj}^{(t)}\). If the penalties are chosen too small, the minimum of \([f/\gamma]\) will not be feasible for \([c/\gamma]\), and the algorithm will have to continue with larger values. As \(M_k \to \infty\) and \(M_k^+ \to \infty\), however, the penalty approach becomes indistinguishable from the two-phase approach. Thus the penalties must be “medium sized” for the penalty approach to be most effective.

Both approaches can be viewed as generalizations of well-known linear programming techniques. Large-scale linear simplex codes have traditionally used certain limited versions of the P-L simplex algorithm to solve the phase one problem [17]; a more general P-L simplex algorithm for phase one was proposed by Rarick [15]. The P-L penalty approach for linear programming has been investigated by Conn [6] and Bartels [2].

5. Finiteness

The assumption of a finite number of finite breakpoints (and finite slopes) is seldom restrictive. If each slope or breakpoint is regarded as a separate piece of data, then such an assumption serves only to insure that every piecewise-linear
R. Fourer / A simplex method for piecewise-linear programming 307

program has a description of finite length, to which an algorithm can meaningfully be applied. Moreover, in practice the breakpoint sets are often inherently small; many of the applications surveyed in part III of this paper [9], for example, require no more than three finite breakpoints in any term of the objective function.

Nevertheless, it is reasonable to consider solving P-LPs under weaker conditions. If the breakpoint and slope sequences are generated by a finite algorithm, then a P-LP can have a finite description even though the sequences are infinite. A P-L simplex method may still be applied, provided that it makes use of only finitely many breakpoints and slopes at each iteration.

Generated slopes and breakpoints are most likely to arise in piecewise-linear approximations to convex functions [4, 7, 13]. As a simple example, a convex function $f_k(x_k)$ can be approximated by letting the breakpoints be $\gamma_k^{(h)} = \Delta h$, for any $\Delta > 0$ and $-\infty < h < +\infty$, and by taking the slopes as

$$c_k^{(h)} = (f(\gamma_k^{(h+1)}) - f(\gamma_k^{(h)}))/(\gamma_k^{(h+1)} - \gamma_k^{(h)}).$$

For a nonlinear program that has a separable convex objective and linear constraints, this kind of P-L approximation yields a piecewise-linear program. Such a P-LP may be more convenient to solve than the original nonlinear program, even though the P-L simplex algorithm cannot guarantee the rapid convergence characteristic of nonlinear methods that exploit information about the curvature of the objective.

This section begins by formulating minimal assumptions under which the P-L simplex algorithm is well-defined when applied to infinite slope and breakpoint sequences. The conditions for optimality, unboundedness and finite termination can then be re-examined. Conditions for optimality depend only on the local nature of a P-L function, and are preserved under even very weak assumptions. Conditions for unboundedness are more problematical, because they are also influenced by the greater variety of asymptotic behavior that is possible when infinite numbers of breakpoints are allowed. Finally, finite termination can be guaranteed only under fairly restrictive assumptions about the nature of the infinite breakpoint sequences.

Minimal assumptions

If absolutely any increasing collection of breakpoints and slopes were allowed to define a convex piecewise-linear function, then all points on the real line could be taken as breakpoints, and any convex function might be regarded as piecewise-linear. Even dense countable sets of breakpoints (such as the rationals) could suffice to describe any convex function. A reasonably restrictive and intuitive notion of piecewise linearity thus requires that each P-L term be linear on segments of positive length. Equivalently, the breakpoints must not be dense in any interval.

A somewhat stronger assumption is needed to ensure that the P-L simplex algorithm can be carried out as described in Section 1. Step (2) requires that there be a well-defined slope to the left and to the right of every breakpoint, and subsequent steps also depend on being able to determine successive slopes and breakpoints to the left or right of $\gamma_{NP}$ and every $c_{ii}$. Thus no breakpoint may be a limit of breakpoints.
Certain other anomalous situations can occur when a subsequence of breakpoints converges to some finite value \( \gamma_k^{(h^*)} \). Under the above assumptions, \([c/\gamma]\gamma_k^{(h^*)}\) may be finite, provided that \(\gamma_k^{(h^*)}\) is not included in the list of breakpoints. Such an arrangement is best avoided, however, for one or more of the following reasons:

- If \([c/\gamma]\) has different slopes to the left and to the right of \(\gamma_k^{(h^*)}\), then \(\gamma_k^{(h^*)}\) is properly included as a breakpoint by any intuitive definition of a piecewise-linear function.

- If \([c/\gamma]\) is finite to the left and to the right of \(\gamma_k^{(h^*)}\), then the sequence of slopes cannot be numbered by consecutive integers, as the statement of the P-L simplex algorithm implicitly assumes.

- If every basic optimal solution has \(x_k = \gamma_k^{(h^*)}\), then the P-L simplex algorithm may not be able to reach an optimum unless \(\gamma_k^{(h^*)}\) is one of the breakpoints.

All of these cases can be ruled out by saying that the function value may not be finite at any limit point of the breakpoints.

All of the above conditions will be assumed in the sequel. Subject to these restrictions, a convex P-L function \([c/\gamma]_k\) either has a largest finite breakpoint (and largest finite slope), or has increasing sequences of infinitely many breakpoints and slopes. In the latter case, \([c/\gamma]_kx_k\) must fall into one of the following categories as \(x_k\) steps to higher and higher breakpoints:

\[
\begin{array}{ccc}
\lim_{h \to +\infty} \gamma_k^{(h)} & \lim_{h \to +\infty} c_k^{(h)} & \lim_{h \to +\infty} [c/\gamma]_k\gamma_k^{(h)} \\
\text{finite} & \text{finite} & \text{finite} \\
\text{finite} & +\infty & \text{finite} \\
\text{finite} & +\infty & +\infty \\
+\infty & \text{negative} & -\infty \\
+\infty & \text{zero} & -\infty \\
+\infty & \text{positive} & +\infty \\
+\infty & +\infty & +\infty \\
\end{array}
\]

The corresponding table for \(-\infty\) is obtained by reflecting the first two columns: \(+\infty\) becomes \(-\infty\), and "negative" is exchanged with "positive".

The first two rows of the table describe somewhat counterintuitive behavior. As the breakpoints approach their finite limit, the distance between them tends to zero; although the intervening slopes may become progressively larger, they grow slowly enough that the function value is bounded. To preserve continuity, the P-L function could be defined to have

\[
[c/\gamma]_k \left( \lim_{h \to +\infty} \gamma_k^{(h)} \right) = \lim_{h \to +\infty} [c/\gamma]_k\gamma_k^{(h)};
\]

then \(\lim_{h \to +\infty} \gamma_k^{(h)}\) would be a feasible point for \([c/\gamma]_k\) (though in the second case it would not be "regularly feasible" [16, Section 8D]).
However, since by assumption a P-L function may not take a finite value at a limit of breakpoints, \([c/\gamma]_k\) must instead be regarded as taking an infinite value at \(\lim_{h \to +\infty} \gamma_k^{(h)}\). Hence \([c/\gamma]_k\) is not a closed convex function in the first two cases above. The assumptions do guarantee that it is closed in all of the other cases.

**Conditions for optimality and unboundedness**

Under the above assumptions, the fundamentally piecewise-linear nature of the functions \([c/\gamma]_k\) is preserved in any sufficiently small neighborhood. Thus local properties, such as the conditions for feasibility and optimality, remain unchanged. In particular, the conditions for termination with optimality in the P-L simplex algorithm remain valid:

**Property 10.** If the conditions in step (2) of the P-L simplex algorithm are satisfied, then the current basic solution is optimal regardless of the finiteness of the slope and breakpoint sets.

If the conditions in step (2) are not satisfied and if the true reduced cost is \(d_{Np}(\delta) < 0\) or \(d_{Np}(\delta) > 0\) for the variable \(x_{Np}\) chosen in step (3), then the P-L simplex algorithm achieves a reduction in the objective value.

**Proof.** If some basic feasible \(\bar{x}\) satisfies the optimality conditions of step (2), then it also satisfies these conditions for the "local" objective \([\hat{c}/\hat{\gamma}]\) defined by

\[
\begin{align*}
\hat{\gamma}_{Bi}^{-(1)} &= -\infty, & \hat{c}_{Bi} &= c_{Bi}; \\
\hat{\gamma}_{Bi}^{+(1)} &= +\infty, \\
\hat{\gamma}_{Nj}^{-(1)} &= -\infty, & \hat{c}_{Nj}^{-(1)} &= c_{Nj}; \\
\hat{\gamma}_{Nj}^{+(1)} &= +\infty,
\end{align*}
\]

Since \([\hat{c}/\hat{\gamma}]\) has only finitely many breakpoints, \(\bar{x}\) must be optimal for \([\hat{c}/\hat{\gamma}]\) by the proof in [8]. Yet by construction,

- (slope of \([\hat{c}/\hat{\gamma}]_k\)) \(\geq\) (slope of \([c/\gamma]_k\)) at all \(x_k < \bar{x}_k\),
- (slope of \([\hat{c}/\hat{\gamma}]_k\)) \(\leq\) (slope of \([c/\gamma]_k\)) at all \(x_k > \bar{x}_k\).

As a consequence, for any \(x\), \([c/\gamma]x - [c/\gamma]\bar{x} \geq [\hat{c}/\hat{\gamma}]x - [\hat{c}/\hat{\gamma}]\bar{x} \geq 0\). Hence \(\bar{x}\) is also optimal for \([c/\gamma]\).

If the conditions of step (2) are not satisfied, then an entering variable \(x_{Np}\) can be chosen in step (3). As \(x_{Np}\) begins to increase or decrease from \(\gamma_{Np}\), the change in the objective value depends only on the slopes immediately to the left or right of \(\bar{x}_{Np}\) and all \(\bar{x}_{Bi}\). In particular, the objective value decreases if and only if the effective reduced cost is favorable, regardless of how many breakpoints \([c/\gamma]\) has in all. □
Unboundedness is much more problematical. Conditions for an unbounded solution depend on the asymptotic, rather than local, properties of \([c/\gamma]\). The assumption of a finite breakpoint set in [8] assures that \([c/\gamma]\_k x_k\) either becomes infinite or behaves like a linear function as \(x_k \rightarrow +\infty\) or \(x_k \rightarrow -\infty\). When this assumption is relaxed, a convex P-L function can exhibit any of the other kinds of limiting behavior allowed of convex functions.

Thus the conditions in step (5) remain sufficient for unboundedness in the infinite-breakpoint case, since they imply that the objective decreases along a ray of solutions that crosses no breakpoints. The same conditions are no longer necessary, however, because the objective may also decrease along a ray that crosses infinitely many breakpoints. In such a case the objective value may be truly unbounded below or may have a finite infimum that is nowhere achieved, even though the conditions of step (5) are not satisfied at any basic feasible solution.

Various weaker necessary and sufficient conditions for the infinite-breakpoint case can be formulated in terms of the effective reduced costs and their limits:

**Property 11.** If \([c/\gamma]\) is closed, then the conditions for unboundedness can be summarized as follows, where \(d^+_N(\infty) = \lim_{\theta \rightarrow \infty} d^+_N(\theta)\) and \(d^-_N(\infty) = \lim_{\theta \rightarrow \infty} d^-_N(\theta)\).

<table>
<thead>
<tr>
<th>Condition: at some feasible basis</th>
<th>Unboundedness: the P-LP satisfies</th>
<th>Necessary?</th>
<th>Sufficient?</th>
</tr>
</thead>
<tbody>
<tr>
<td>(d^+_N(\infty) &lt; 0) or (d^-_N(\infty) &gt; 0)</td>
<td>(\inf[c/\gamma]x = -\infty)</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>(d^+_N(\theta) &lt; 0) or (d^-_N(\theta) &gt; 0) for all (\theta &gt; 0)</td>
<td>(\inf[c/\gamma]x) is nowhere achieved</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>(d^-_N(\infty) \leq 0) or (d^+_N(\infty) \geq 0)</td>
<td>({x: [c/\gamma]x \leq [c/\gamma]x}) is unbounded for all feasible (\bar{x})</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

The same conditions hold if \([c/\gamma]\) is not closed, except that the second condition also fails to be necessary.

**Proof.** All of these assertions follow directly from established properties of arbitrary convex separable objective functions; see, for example, the analysis by Rockafellar [16, Chapter 11].

The asymptotic effective reduced costs can be expressed conveniently as a function of the asymptotic slopes, provided that the breakpoints have no finite limits. Writing \(\sup_h c^h_k = c^{(+\infty)}_k\) and \(\inf_h c^h_k = c^{(-\infty)}_k\),

\[
\begin{align*}
   d^+_N(\infty) &= \lim_{\theta \rightarrow \infty} c^+_N(\theta) + \sum_{y_{Bi} < 0} \left( \lim_{\theta \rightarrow \infty} c^+_{Bi}(\theta) \right) y_{Bi} + \sum_{y_{Bi} > 0} \left( \lim_{\theta \rightarrow \infty} c^-_{Bi}(\theta) \right) y_{Bi} \\
   &= c^{(+\infty)}_N + \sum_{y_{Bi} < 0} c^{(+\infty)}_{Bi} y_{Bi} + \sum_{y_{Bi} > 0} c^{(-\infty)}_{Bi} y_{Bi}
\end{align*}
\]
and similarly for \( d_{\tilde{N}_p}(\infty) \). These formulas depend on the identity of the basic variables \( x_B \) and the entering variable \( x_{N^p} \), but not on the values \( y_N \) of the nonbasic variables. Thus there are only finitely many quantities \( d^+_{N_p}(\infty) \) and \( d_{\tilde{N}_p}(\infty) \) associated with a P-LP even when the number of breakpoints is infinite.

**Conditions for termination**

Since an infinity of breakpoints permits an infinite number of basic solutions, abandonment of the finiteness assumption allows an infinite number of iterations in the P-L simplex algorithm. The proof of finite termination in [8] remains applicable, however, if only finitely many basic feasible solutions actually exist, or if only finitely many can be visited by the algorithm.

Termination can be assured, for example, if the algorithm visits some \( \bar{x} \) such that \( \frac{[c/\gamma]x}{[c/\gamma]} \geq \frac{[c/\gamma]x'}{[c/\gamma]} \) for only finitely many other basic feasible solutions \( x' \). This is the case, in particular, when \( [c/\gamma] \) is a closed function and the solution set is bounded:

**Property 12.** If \( [c/\gamma] \) is closed and if the set of optimal solutions is nonempty and bounded, then the P-L simplex algorithm must eventually stop.

**Proof.** Consider first any closed and bounded set \( \mathcal{L} \) of feasible solutions; \( [c/\gamma]x < \infty \) for any \( x \in \mathcal{L} \). Let \( \mathcal{L}_k = \{ x_k : (x_1, \ldots, x_k, \ldots, x_n) \in \mathcal{L} \} \) be the closed and bounded interval of values that \( x_k \) takes within \( \mathcal{L} \); \( [c/\gamma]x_k \) is finite for any \( x_k \in \mathcal{L}_k \). Under the hypothesis that \( [c/\gamma] \) is closed, and under the assumption that \( [c/\gamma] \) takes an infinite value at any limit of breakpoints, the interval \( \mathcal{L}_k \) may contain only a finite number of breakpoints. If follows that \( \mathcal{L} \) must contain only finitely many basic solutions.

For any optimal solution \( x^* \), the level set \( \{ x : [c/\gamma]x \leq [c/\gamma]x^* \} \) is nonempty and bounded by hypothesis. If one nonempty level set of a convex function is bounded, however, then all are bounded.

Thus if the algorithm is started at some basic feasible solution \( \bar{x} \), the level set given by \( \mathcal{L} = \{ x : [c/\gamma]x \leq [c/\gamma]\bar{x} \} \) is nonempty and bounded. \( \mathcal{L} \) is also closed, because by hypothesis \( [c/\gamma] \) is closed. Hence by the previous argument \( \mathcal{L} \) can contain only finitely many basic feasible solutions. Yet the algorithm can only visit basic feasible solutions in \( \mathcal{L} \) (since it never permits the objective to increase) and it can visit each basic solution at most once. As a result, it must stop after only finitely many iterations. \( \square \)

Other, more restrictive requirements for termination depend only on the nature of the objective. One result can be stated in terms of the asymptotic behavior of \( [c/\gamma] \):

**Property 13.** If \( [c/\gamma] \) is closed and if \( \lim_{x_k \to -\infty} [c/\gamma]x_k = \lim_{x_k \to +\infty} [c/\gamma]x_k = \infty \) for every \( k \), then the P-L simplex algorithm must eventually stop.

**Proof.** Let \( \bar{x} \) be the basic simplex solution at which the algorithm is started. The level set \( \mathcal{L} = \{ x : [c/\gamma]x \leq [c/\gamma]\bar{x} \} \) is nonempty and convex; the condition that
\[ \lim_{x_k \to -\infty} [c/\gamma]_k x_k = \lim_{x_k \to +\infty} [c/\gamma]_k x_k = \infty \] for every \( k \) implies that \( \mathcal{L} \) is also bounded. Thus, just as in the proof of Property 12, the algorithm cannot visit infinitely many bases. \( \square \)

Alternatively, a condition for termination may be given in terms of the slopes and breakpoints:

**Property 14.** If the differences \( c_k^{(h+1)} - c_k^{(h)} \) between slopes and \( \gamma_k^{(h+1)} - \gamma_k^{(h)} \) between breakpoints are bounded below by some positive value for all \( k \), then the P-L simplex algorithm must eventually stop.

**Proof.** Suppose that the algorithm visits an infinite sequence of basic feasible solutions. The idea of the proof is to first extract a particularly convenient infinite subsequence. Then the objective value along the subsequence can be shown to eventually increase, producing a contradiction.

It is useful to first note that, by the Property's hypothesis, neither the slopes nor the breakpoints may converge. As a result, under the previous assumptions on breakpoint sequences in \([c/\gamma]_k\), either there is a smallest finite slope \( c_k^{(s)} \) (with \( \gamma_k^{(s)} = -\infty \)) or

\[ \lim_{h \to -\infty} c_k^{(h)} = \lim_{h \to -\infty} \gamma_k^{(h)} = -\infty \quad \text{and} \quad \lim_{x_k \to -\infty} [c/\gamma]_k x_k = \infty. \]

Similarly, either there is a largest finite slope \( c_k^{(t)} \) (with \( \gamma_k^{(t)} = +\infty \)) or

\[ \lim_{h \to +\infty} c_k^{(h)} = \lim_{h \to +\infty} \gamma_k^{(h)} = +\infty \quad \text{and} \quad \lim_{x_k \to +\infty} [c/\gamma]_k x_k = \infty. \]

Since there are only finitely many ways to choose a nonsingular submatrix \( B \) from the columns of \( A \), an infinite sequence of basic solutions must contain a subsequence in which the same \( m \) variables are basic and the same \( n - m \) are nonbasic. Within this subsequence, moreover, the values \( \bar{x}_{Bi} \) of a basic variable may or may not fall between some two breakpoints infinitely often. Consequently, taking into account the restrictions of the hypothesis, it must be possible to extract a sub-subsequence along which one of the following holds:

- **\( x_{BV} \):** The objective values \([c/\gamma]_{Bi} \bar{x}_{Bi}\) are strictly increasing and tend to \( \infty \) along the subsequence.
- **\( x_{BW} \):** For some index \( h_i \), \( \gamma_{Bi}^{(h)} \leq \bar{x}_{Bi} \leq \gamma_{Bi}^{(h+1)} \) at every iteration in the subsequence.

By performing the extraction for each \( x_{Bi} \) in turn, a subsequence can be found in which every basic variable satisfies one of these restrictions. Let \( x_{BV} \) and \( x_{BW} \) denote the basic variables of these two types, as indicated above.

By similar reasoning, the nonbasic values \( \bar{x}_{Nj} \) may or may not lie at the same breakpoint infinitely often, and it is possible to further extract a subsequence such
that one of the following holds:

\[ x_{NS} : \text{The values} \ x_{Nj} \ \text{lie at a strictly decreasing series of breakpoints that tend to} \ -\infty \ \text{along the subsequence.} \]

\[ x_{NT} : \text{The values} \ x_{Nj} \ \text{lie at a strictly increasing series of breakpoints that tend to} \ +\infty \ \text{along the subsequence.} \]

\[ x_{NU} : \text{For some index} \ h_j, \ x_{Nj} = \gamma^{(h_j)}_{Nj} \ \text{at every iteration in the subsequence.} \]

By a series of extractions, a subsequence can be found in which every \( x_{Nj} \) satisfies one of the above (and in which the properties of the subsequence relative to the variables \( x_{Bi} \) are preserved). Let \( x_{NSj} \), \( x_{NTj} \) and \( x_{NUj} \) denote the nonbasic variables that fall into each of the three categories. There must exist a subsequence that has at least one variable of type \( x_{NSj} \) or \( x_{NTj} \), since otherwise the algorithm could not visit infinitely many distinct basic solutions.

Let \( \bar{x} \) and \( \hat{x} \) denote two successive basic solutions in the subsequence. The remainder of the proof must show that \( \bar{x} \) can be chosen so that \( [c/\gamma]\bar{x} > [c/\gamma]\hat{x} \), contradicting the assumption that \( \hat{x} \) follows \( \bar{x} \) in a series of simplex iterations.

Since \( \hat{x}_{NU} = \hat{x}_{NU} \) throughout the subsequence, the difference between the objective values at \( \bar{x} \) and \( \hat{x} \) can be broken into just four parts:

\[
[c/\gamma]\bar{x} - [c/\gamma]\hat{x} = ([c/\gamma]_{NS}\bar{x}_{NS} - [c/\gamma]_{NS}\hat{x}_{NS}) + ([c/\gamma]_{NT}\hat{x}_{NT} - [c/\gamma]_{NT}\bar{x}_{NT}) + ([c/\gamma]_{BV}\hat{x}_{BV} - [c/\gamma]_{BV}\bar{x}_{BV}) + ([c/\gamma]_{BW}\hat{x}_{BW} - [c/\gamma]_{BW}\bar{x}_{BW}).
\]

However, because \( [c/\gamma]_{BW} \) has the same slope \( c^{(h_j)}_{BW} \) at both \( \bar{x}_{BWj} \) and \( \hat{x}_{BWj} \), the difference in objective terms for these variables is simply

\[
[c/\gamma]_{BW}\hat{x}_{BWj} - [c/\gamma]_{BW}\bar{x}_{BWj} = (\hat{x}_{BWj} - \bar{x}_{BWj})c^{(h_j)}_{BWj}.
\]

The equations \( A\hat{x} = b \) and \( A\bar{x} = b \) imply \( A(\hat{x} - \bar{x}) = 0 \), or \( B(\hat{x}_B - \bar{x}_B) = -\sum_j (\hat{x}_{Nj} - \bar{x}_{Nj})a_{Nj} \). Letting \( Bw_{NJj:B} = a_{Nj} \) as in Section 2, and dropping the terms \( \hat{x}_{NUj} - \bar{x}_{NUj} = 0 \), it follows that

\[
\hat{x}_{BWj} - \bar{x}_{BWj} = -\sum_j (\hat{x}_{Nj} - \bar{x}_{Nj})w_{NJj:BWj} - \sum_j (\hat{x}_{NTj} - \bar{x}_{NTj})w_{NTj:BWj}.
\]

Substituting in the preceding expressions,

\[
[c/\gamma]\bar{x} - [c/\gamma]\hat{x} = \sum_j (\hat{x}_{NSj} - \bar{x}_{NSj})\left( [c/\gamma]_{NS}\hat{x}_{NSj} - [c/\gamma]_{NS}\bar{x}_{NSj} \right) - \sum_l w_{NSj:BWl}c^{(h_j)}_{BWl} + \sum_j (\hat{x}_{NTj} - \bar{x}_{NTj})\left( [c/\gamma]_{NT}\hat{x}_{NTj} - [c/\gamma]_{NT}\bar{x}_{NTj} \right) - \sum_l w_{NTj:BWl}c^{(h_j)}_{BWl} + \sum_l ([c/\gamma]_{BV}\hat{x}_{BVl} - [c/\gamma]_{BV}\bar{x}_{BVl}).
\]
Consider the expression within the first sum above, for any \( j \). The first factor \((\bar{x}_{N_{S_j}} - \bar{x}_{N_{S_j}})\) is negative and bounded away from zero, by the definition of the subsequence and the hypothesis about breakpoints. Within the second factor, the first term is just the average rate of change of the function \([c/\gamma]_{N_{S_j}}\) when its argument moves from \(\bar{x}_{N_{S_j}}\) to \(\bar{x}_{N_{S_j}}\); this quantity is at least as great as the slope to the left of \(\bar{x}_{N_{S_j}}\), so it must tend to \(-\infty\) as \(\bar{x}\) moves out the subsequence, by the hypothesis about slopes. The second term in the second factor is the same for any \(\bar{x}\) in the subsequence. Thus the first sum — if it is not empty — must grow arbitrarily large as the subsequence proceeds; an entirely analogous argument shows the same for the second sum. Since there must be at least one variable \(x_{N_{S_j}}\) or \(x_{N_{T_j}}\), both sums cannot be empty.

Finally, all of the terms in the third sum are positive by the definition of the subsequence. Thus the total difference \([c/\gamma]\hat{x} - [c/\gamma]\bar{x}\) must be positive for \(\hat{x}\) far enough along the subsequence, which was the contradiction to be shown. \(\square\)

Given some postulated lower-bound value, each iteration of the P-L simplex algorithm can easily check that all newly encountered slopes and breakpoints obey the conditions of the above property. Then the algorithm must eventually stop either with an optimal solution, or with a warning that some pair of breakpoints or slopes differ by less than the lower bound.

Addendum

Two early descriptions of piecewise-linear simplex algorithms predate those surveyed in part I of this paper [8].

Gol'stein [10] states an algorithm that is essentially equivalent to the P-L simplex algorithm of [8]. Nondegeneracy is assumed, so that the usual proof of termination applies, but no proof of optimality is suggested.

Orden and Nalbandian [14] describe a piecewise-linear simplex algorithm in terms of the traditional simplex tableau. Their method steps from one basic solution to another, in the same way as the algorithm in [8]. Their test for optimality and their selection of an entering variable rely on the true reduced costs (as defined in Section 2 above) rather than on the nominal reduced costs used in [8]; hence a nondegeneracy assumption is necessary for optimality of the final basis as well as for finite termination.

References


