Making Progress During a Stall in the Simplex Algorithm

George B. Dantzig*
Department of Operations Research
Stanford University
Stanford, California 94305-4022

Dedicated to Alan J. Hoffman on the occasion of his 65th birthday.

Submitted by Uriel G. Rothblum

ABSTRACT

All the standard methods for avoiding cycling in the simplex algorithm use row selection rules for resolving degeneracy. The method proposed in this paper does not. Instead, the Gass-Saaty parametric method applied to the objective (cost) form is used to choose the incoming column. The pivot-row choice among blocking rows, i.e., those tied for pivot, is arbitrary. A simple anticycling device is used which avoids dual degeneracy of the parametrized objective with "probability one." Tests were run on nine highly degenerate practical test problems ranging in size from small to large. Using MINOS software, the standard simplex method required 31,195 iterations, of which 20,504 (or 66%) were blocked pivots. Using the parametric scheme required 13,812 iterations, of which 67% were blocked. The reduction in the number of iterations for this set of highly degenerate test problems was 56%. The CPU-time reduction was 48%.

1. INTRODUCTION

The linear program which we wish to solve using the simplex method is

$$\text{FIND} \quad \min z, \quad Ax = b, \quad z = cx, \quad x \geqslant 0.$$

The algorithm is said to *stall* (or be blocked) when one or more pivot steps

*Research of this report was partially supported by the National Science Foundation Grants DMS-8420623, ECS-8617905, and SES-8518662; U.S. Department of Energy Grant DE-FG03-87ER25028; Office of Naval Research Contract N00014-85-K-0343.

655 Avenue of the Americas, New York, NY 10010 0024-3795/89/$3.50

results in no change in the objective z. All the well-known methods for resolving ties during a stall select the "right" row to pivot on among the blocking rows of the canonical form of iteration t. Thus, the ϵ-perturbation as implemented using the lexicographic rule [2], the inductive method as implemented by Wolfe's rule [8], and the random-row choice rule [2] are all examples of row-selection rules for avoiding cycling. Bland's rule is a scheme that selects both the column and the row [1]. From the point of view of reducing the numbers of iterations or the computation time during a stall, little good can be said about any of them in the author's opinion.

The method presented here is a rule for selecting the incoming column only. Once the column is selected, any row among the set of blocking rows of the canonical form may be used for pivoting. For numerical stability, pivoting on the largest coefficient among the blocking rows is recommended. Except for column selection, all other steps are the same as the standard simplex method. We will therefore not review these and will assume they are known to the reader.

2. GASS-SAATY PARAMETRIC METHOD

The rule proposed for choosing the incoming column is the one used in the parametric method of Gass and Saaty [4]; it reduces dual infeasibility (as measured by how much a certain vector used to perturb the objective coefficients must be scaled up to make the dual solution feasible). This parametrized objective is defined below in such a way as to avoid dual degeneracy with "probability one." We will first state the Gass-Saaty column-selection scheme in the framework of the canonical form. We then discuss how the updates can be easily computed from the data of the original problem using the framework of the revised simplex method and LU factorization of the basis.

We assume that a starting feasible basis B^0 is given and $\beta^0 = \{j_1^0, \ldots, j_m^0\}$ is the set of basic column indices j of the initial basis. An additional equation $dx = w$ is used to parametrize the cost equation $z = cx$. Let

$$\sum d_j x_j = w, \qquad \text{where} \quad d_j = \begin{cases} 0 & \text{for} \quad j \in \beta^0, \\ \|A_{\cdot j}\|(1+\epsilon_j) & \text{for} \quad j \notin \beta^0, \end{cases}$$

where $\|A_{\cdot j}\|$ is the Euclidean norm, some other norm, or any other fixed positive function of $A_{\cdot j}$; the value of ϵ_j is chosen from a table of random numbers $0.0 < \epsilon_j < 0.1$. The author recommends the use of a column norm so

that the choice of entering column j will be independent of the units used to measure the level of activity j.

Denote by B^{t-1} the feasible basis at the start of iteration t, and denote by c_β and d_β the subsets of components of c and d corresponding to basic column indices $\beta = \beta^{t-1}$. Let π and σ be found by solving

$$c_\beta = \pi B^{t-1} \quad \text{and} \quad d_\beta = \sigma B^{t-1}.$$

On iteration t, the parametrized reduced cost form is

$$(\bar{c} + \theta \bar{d})x = (z - z_{t-1}) + \theta(w - w_{t-1}),$$

where

$$\bar{c} = c - \pi A \quad \text{and} \quad \bar{d} = d - \sigma A.$$

If $\bar{c} \geqslant 0$, then $\theta = 0$, and the iterative process STOPS with the current basic feasible solution optimal.

Otherwise at least one $\bar{c}_j < 0$. We make the *inductive assumption*, for all θ in some range $0 < \theta^t < \theta < \theta^{t-1}$, that

$$\bar{c}_j(\theta) = \bar{c}_j + \theta \bar{d}_j > 0 \qquad \text{for all} \quad j \notin \beta.$$

Moreover, we make the uniqueness assumption that at $\theta = \theta^t$,

$$\bar{c}_j(\theta^t) = \bar{c}_j + \theta^t \bar{d}_j > 0 \qquad \text{for all} \quad j \notin \beta \text{ except } j = s,$$

$$\bar{c}_s(\theta^t) = \bar{c}_s + \theta^t \bar{d}_s = 0 \qquad \text{for some} \quad s \notin \beta.$$

The choice of s will be unique with probability one because of the random selection of ϵ_j. For proof of this assertion, see [6]. Note that the choice of column s is "unit free," i.e., it would not be affected if all columns $A_{\cdot j}$ were rescaled by positive factors when d_j is initially chosen proportional to some norm of $A_{\cdot j}$.

It is easy to see that $\bar{d}_s > 0$, because for $\theta^t < \theta < \theta^{t-1}$ we have $\bar{c}_s + \theta \bar{d}_s > 0$; subtracting $\bar{c}_s + \theta^t \bar{d}_s = 0$ yields $(\theta - \theta^t)\bar{d}_s > 0$, where $\theta - \theta^t > 0$. Moreover, $\bar{c}_s = -\theta^t \bar{d}_s < 0$. We will make use of the fact that $\bar{d}_s > 0$ and $\bar{c}_s < 0$.

According to the theory of Gass and Saaty [4] (see also [2]), after a pivot in the selected column s, it is possible once again to decrease θ by a positive amount. The proof is as follows: After pivoting on some row r, for $j \neq j_r$, $j \notin \beta^{t-1}$, $\theta \leqslant \theta^t$, we have

$$\bar{c}_j(\theta) = \left(\bar{c}_j - \frac{\bar{a}_{rj}\bar{c}_s}{\bar{a}_{rs}}\right) + \theta\left(\bar{d}_j - \frac{\bar{a}_{rj}\bar{d}_s}{\bar{a}_{rs}}\right)$$

$$= \left(\bar{c}_j - \frac{\bar{a}_{rj}\bar{c}_s}{\bar{a}_{rs}}\right) + (\theta - \theta^t)\left(\bar{d}_j - \frac{\bar{a}_{rj}\bar{d}_s}{\bar{a}_{rs}}\right) + \theta^t\left(\bar{d}_j - \frac{\bar{a}_{rj}\bar{d}_s}{\bar{a}_{rs}}\right)$$

$$= \left(\bar{c}_j + \theta^t\bar{d}_j\right) + (\theta - \theta^t)\left(\bar{d}_j - \frac{\bar{a}_{rj}\bar{d}_s}{\bar{a}_{rs}}\right),$$

where we have dropped the second term from the first and the third parenthesis because $\bar{c}_s + \theta^t\bar{d}_s = 0$, by the definition of θ^t. Note $\bar{c}_j + \theta^t\bar{d}_j > 0$ by the uniqueness assumption for $j \neq \beta^t$, $j \neq s$. Therefore, for some range $\theta < \theta^t$ for $j \notin \beta^{t-1}$ and $j \neq s$, we have $\bar{c}_j(\theta) > 0$. For $j = j_r$, we have for all $\theta < \theta^t$

$$\bar{c}_{j_r}(\theta) = -\frac{\bar{c}_s + \theta\bar{d}_s}{\bar{a}_{rs}}$$

$$= (\theta - \theta^t)\frac{-d_s}{\bar{a}_{rs}} > 0,$$

because $\bar{a}_{rs} > 0$ and $\bar{d}_s > 0$, as we have shown earlier. Let β^t be the updated set of basic indices. Since now $\bar{c}_j + \theta^t\bar{d}_j > 0$ for $j \notin \beta^t$, it follows for some *new range* $\theta^{t+1} < \theta < \theta^t$ that $\bar{c}_j(\theta) > 0$. We are now ready to repeat the iterative process.

Convergence is guaranteed in a finite number of iterations because repetition of a canonical form for some iteration $t + \tau$ with a $\theta < \theta^{t+1}$ would imply a lowering of θ below the calculated minimum θ^{t+1}, a contradiction. On all iterations, the incoming column has $\bar{c}_s < 0$, so that if there is no stalling, there will be a positive decrease in z. On all iterations θ is strictly decreasing.

3. UPDATING

First Way

If the vectors $\bar{c}$ and $\bar{d}$ of iteration $t-1$ are stored, then those of iteration t can be computed by a *single solve* for "prices" and a *single "pricing out"* of columns. For pricing-out vector ρ, the rth row of the inverse of B^{t-1} is used. This requires one to solve

$$\rho B^{t-1} = U_r, \qquad \bar{a}_{rj} = \rho A_{\cdot j},$$

where U_r is unit vector r. Columns j are "priced out" by using ρ as a "pricing" vector and computing $\bar{a}_{rj} = \rho A_{\cdot j}$. The update formulas are

$$\text{updated } \bar{c}_j = \bar{c}_j - \lambda \bar{a}_{rj}, \qquad \text{where} \quad \lambda = \bar{c}_s / \bar{a}_{rs},$$

$$\text{updated } \bar{d}_j = \bar{d}_j - \mu \bar{a}_{rj}, \qquad \text{where} \quad \mu = \bar{d}_s / \bar{a}_{rs}.$$

Second Way

Instead, the updating of $\bar{c}_j$ can be done in the usual way by determining π by a *single solve* $B^t \pi^t = c_\beta$, where $c_\beta = (c_{j_1}, c_{j_2}, \ldots, c_{j_m})$ and $\beta = \beta^t$. This way also requires a single pricing out:

$$\text{updated } \bar{c}_j = c_j - \pi^t A_{\cdot j},$$

$$\text{updated } \bar{d}_j = \bar{d}_j - \frac{\mu}{\lambda}\left(\bar{c}_j - \text{updated } \bar{c}_j\right),$$

which can easily be verified by eliminating $\bar{a}_{rj}$ from the first way of updating. This requires not overlaying updated $\bar{c}_j$ on stored $\bar{c}_j$ until after updated $\bar{d}_j$ is computed.

Third Way

Edward Klotz [5] recommends a *double pricing* scheme that requires *one solve and no storing* of $\bar{c}$ and $\bar{d}$:

$$\text{updated } \bar{c}_j = c_j - \pi^t A_{\cdot j},$$

$$\text{updated } \bar{d}_j = d_j - \sigma^t A_{\cdot j},$$

where σ' is generated without a solve by

$$\text{updated } \sigma = \sigma + \frac{\bar{d}_s}{\bar{c}_s}(\text{updated } \pi - \pi),$$

which requires only a temporary storing of components π_i of π as it is computed to find the corresponding components of updated σ_i. The above formula for updated σ is obtained by eliminating ρ from the relations

$$\text{updated } \pi = \pi + \frac{\bar{c}_s}{\bar{a}_{rs}}\rho,$$

$$\text{updated } \sigma = \sigma + \frac{\bar{d}_s}{\bar{a}_{rs}}\rho.$$

This third way of updating was the one Klotz implemented for the experiments presented at the end of this paper.

4. BENEFITS

(1) As noted earlier, $\|A_{\cdot j}\|$ can be any positive number. If it is a column *norm* of some kind, however, then the particular choice of initial $d_j = \|A_{\cdot j}\|(1+\epsilon_j)$ makes the selection of s, the incoming column, independent of the unit for measuring the jth activity. The random choice of ϵ_j avoids dual degeneracy with probability 1. The choice of norm can be the Euclidean norm $\|A_{\cdot j}\|_2$, or the sum of positive components of $A_{\cdot j}$ or $\max_i A_{ij}$ provided some $A_{ij} > 0$, or $\max_i |A_{ij}|$. Any of these will render column selection unit free.

(2) The choice of incoming column s by the parametrization scheme appears to be at least as good a choice as the usual rule $s = \text{argmin}\, \bar{c}_j$. This comment is based on a comparison of the number of iterations and computation time required to solve 62 problems drawn from practical sources used by the Systems Optimization Laboratory at Stanford for testing the efficiency of various linear-programming algorithms. Using the geometric mean of ratios of CPU times for comparison, on a set of highly degenerate problems the parametric scheme was 29% faster; on the "PILOT" set it was 70% faster; on the "Staircase" set it was 5% faster; on a set of six "Ship" problems it was

30% slower; and on the remaining set of 29 problems, it was no better than the standard simplex rule.

(3) Near dual degeneracy does not appear likely (in the author's opinion) to cause the same "treading of water and getting nowhere" as does degeneracy in the primal, because all the near-tying columns are promising columns to enter the basis; if not considered on a particular iteration, they are likely to be considered soon for entry into the basis.

(4) Decreasing dual infeasibility, as measured by the monotonic decrease of θ during the iterations when decrease in objective in z is stalled, seems to be a better strategy than the artificial schemes (such as the lexico min ratio, Bland's rule, Wolfe's rule, or the random rule) that have been proposed to avoid cycling.

5. DRAWBACKS

When *partial pricing* is used, it is recommended that a parametric scheme be imposed on each partition separately with θ replaced with separate $\theta_1, \theta_2, \ldots$ for each partition. The reduction of θ_i in one partition may require the value of the θ_{i+1} of the next partition to be reinitiated at a higher value than its last reduction, and therefore it is doubtful that one can prove convergence under partial pricing when the primal is stalled. Convergence for unstalled steps is, of course, guaranteed.

6. TESTS ON PRACTICAL PROBLEMS

The Systems Optimization Laboratory of Stanford University, Operations Research Department, has collected a number of test problems drawn from practical sources. These are used in systematic trials comparing various proposed techniques for solving linear programs. See for example Irvin Lustig [6]. As part of his forthcoming Ph.D. thesis, Edward Klotz tested a number of proposed methods, including some of his own invention, for reducing the number of iterations and the CPU time. Among them is the method proposed in this paper [5].

All the experiments were run by Klotz on a DEC Micro Vax using MINOS 5.1 on nine test problems specially selected because a high percentage of their iterations were blocked using the "regular" simplex method. The proposed "parametric" method was compared with the latter. Neither method used the scaling or partial-pricing options of MINOS. Except for column selection, all features of MINOS were identical for both methods. See Table 1 comparing performance on a set of very degenerate practical problems.

TABLE 1

COMPARATIVE PERFORMANCE OF REGULAR (REG) VS. PARAMETRIC (PAR) SIMPLEX METHOD ON NINE HIGHLY DEGENERATE PROBLEMS

Problem name	Problem size: No. of rows	Problem size: No. of cols.[a]	Iterations: REG	Iterations: PAR	CPU time: REG	CPU time: PAR
KB2	46	41	65	80	6.41	7.43
DEGEN1	67	72	15	23	4.64	5.18
TUFF	371	587	1407	524	527	222
DEGEN2	445	534	1264	1062	518	463
WOOD1P	486	2594	564	745	1383	1975
NZFRI	624	3521	10970	2268	11164	2703
WOODW	1089	8405	2381	1841	5591	4836
DEGEN3	1504	1818	11096	4921	18028	8114
CYCLE	2234	2857	3433	2348	6963	4612
Total			31195	14379	44185	22938

[a] Excludes slacks.

$$\text{(Parametric iterations)}/\text{(regular iterations)} = 14379/31195 = 0.44$$

$$\text{(Parametric CPU)}/\text{(regular CPU)} = 22938/44185 = 0.52$$

Geometric mean of the nine problem ratios (parametric/regular):

Iteration	0.68
CPU	0.71

Proportion of iterations stalled:

Regular simplex	0.66
Parametric scheme	0.67

REFERENCES

1 R. G. Bland, New finite pivot rules for the simplex method, *Math. Oper. Res.* 2:103–107 (1977).

2 George B. Dantzig, *Linear Programming and Extensions*, Princeton U.P., 1963, p. 228; p. 140; Theorems 2, 3, p. 246.

3 B. C. Eaves, The linear complementarity problem, *Management Sci.* 17(9):632 (1971).
4 Saul I. Gass and T. L. Saaty, The computational algorithm for the parametric objective function, *Naval Res. Logist. Quart.* 2(1):39–45 (June 1955).
5 Edward Klotz, Ph.D. Thesis, in preparation.
6 Irvin Lustig, Comparisons of Composite Simplex Algorithms, Ph.D. Thesis, TR SOL 87-8, Dept. of Operations Research, Stanford Univ., June 1987.
7 B. A. Murtagh and M. A. Saunders, MINOS 5.0 User's Guide, TR SOL 83-20, Dept. of Operations Research, Stanford Univ., Dec. 1983.
8 P. Wolfe, A technique for resolving degeneracy in linear programming, *J. Soc. Indust. and Appl. Math.* 11:205–211 (1963).

Received 1 March 1988; final manuscript accepted 15 July 1988