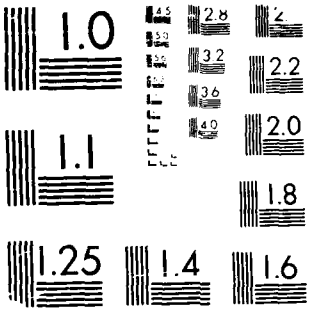

MICROCOPY RESOLUTION TEST CHART

Systems
Optimization
Laboratory

MAKING PROGRESS DURING A STALL IN THE SIMPLEX ALGORITHM

by

George B. Dantzig

TECHNICAL REPORT SOL 88-5

February 1988

Department of Operations Research
Stanford University
Stanford, CA 94305

(4)

SYSTEMS OPTIMIZATION LABORATORY
DEPARTMENT OF OPERATION RESEARCH
STANFORD UNIVERSITY
STANFORD, CALIFORNIA 94305-4022

MAKING PROGRESS DURING A STALL IN THE SIMPLEX ALGORITHM

by

George B. Dantzig

TECHNICAL REPORT SOL 88-5

February 1988

Research and reproduction of this report were 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.

MAKING PROGRESS DURING A STALL IN THE SIMPLEX ALGORITHM

George B. Dantzig

Abstract

All the standard methods for avoiding cycling in the simplex algorithm set up a subprogram related to the blocking rows in the canonical form and look for an improving homogeneous solution. 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 [2]. The pivot-row choice among blocking rows, i.e., those tied for pivot, is arbitrary. A simple anti-cycling device is used which avoids dual degeneracy of the parameterized objective with probability one. Tests were run on nine highly degenerate practical test problems ranging in size from small to large. Using MINOS software [5], the standard simplex method required 31,195 iterations of which 20,504 (or 66%) were blocked pivots. Using the parametric scheme required 14,379 iterations, of which 9,481 or 66% were also blocked. The reduction in the number of iterations for this set of highly degenerate test problems is over 50%. CPU time reduction under the current way of implementation is 40%.

1. Introduction

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

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

The algorithm is said to *stall* (or be blocked) when one or more pivot steps results in no change in the objective z. It is generally believed (but this turns out to be wrong) that the way to make progress during a stall is to 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, the inductive method as implemented by Wolfe's rule, and the random-row choice rule are all examples of row-selection rules for avoiding cycling. Bland's rule is a scheme that selects both the column and the row. From the point of view of reducing the number of iterations during a stall, little good can be said about any of them.

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 [2]; it reduces dual infeasibility. A perturbation of the parametrized objective is introduced to avoid dual degeneracy with probability one. We will first state their 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 parameterize the cost equation $cx = z$. Let

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

where ϵ_j is chosen from a table of random numbers $0 < \epsilon_j < 1$ and $\|A_{\cdot j}\|$ is the Euclidean norm, some other norm, or any other arbitrarily chosen positive number.

Denote by B^{t-1} the feasible basis at the start of iteration t and denote by c_β and d_β the subset 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 parameterized 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 $\bar{c} \geq 0$, 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

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

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

$$\begin{aligned} \bar{c}_j(\theta^t) &= \bar{c}_j + \theta^t \bar{d}_j > 0 \qquad \text{for all } j \notin \beta \text{ except } j = s\,, \\ \bar{c}_s(\theta^t) &= \bar{c}_s + \theta^t \bar{d}_s = 0 \qquad \text{for some } s \notin \beta\,. \end{aligned}$$

The choice of s will be unique with probability one because of the random selection of ϵ_j. 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 [2], see also [1], 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 $j \neq j_r$, $j \notin \beta^{t-1}$, $\theta \leq \theta^t$:

$$\begin{aligned} \bar{c}_j(\theta) &= (\bar{c}_j - \bar{a}_{rj}\bar{c}_s/\bar{a}_{rs}) + \theta(\bar{d}_j - \bar{a}_{rj}\bar{d}_s/\bar{a}_{rs}) \\ &= (\bar{c}_j - \bar{a}_{rj}\bar{c}_s/\bar{a}_{rs}) + (\theta - \theta^t)(\bar{d}_j - \bar{a}_{rj}\bar{d}_s/\bar{a}_{rs}) + \theta^t(\bar{d}_j - \bar{a}_{rj}\bar{d}_s/\bar{a}_{rs}) \\ &= (\bar{c}_j + \theta^t \bar{d}_j) + (\theta - \theta^t)(\bar{d}_j - \bar{a}_{rj}\bar{d}_s/\bar{a}_{rs})\,, \end{aligned}$$

where we have dropped the second term from the first and 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 $j \neq s, \bar{c}_j(\theta) > 0$. For $j = j_r$, we have for all $\theta < \theta^t$:

$$\begin{aligned} \bar{c}_{j_r}(\theta) &= -(\bar{c}_s + \theta \bar{d}_s)/\bar{a}_{rs} \\ &= (\theta - \theta^t)(-\bar{d}_s/\bar{a}_{rs}) > 0 \end{aligned}$$

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 $t + r$ 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 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 r-th row of the inverse of B^{t-1} is used. This requires one to solve

$\rho B^{t-1} = U_r$ where U_r is unit vector r. Columns are "priced out" by computing $\bar{a}_{rj} = \rho A_{\cdot j}$. The update formulas are

$$\begin{array}{lrcl} \text{updated} & \bar{c}_j & = & \bar{c}_j - \lambda \bar{a}_{rj} \quad \text{where } \lambda = \bar{c}_s/\bar{a}_{rs} \\ \text{updated} & \bar{d}_j & = & \bar{d}_j - \mu \bar{a}_{rj} \quad \text{where } \mu = \bar{d}_s/\bar{a}_{rs} \, . \end{array}$$

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,

$$\begin{array}{lrcl} \text{updated} & \bar{c}_j & = & c_j - \pi^t A_{\cdot j} \\ \text{updated} & \bar{d}_j & = & \bar{d}_j - (\mu/\lambda)(\bar{c}_j - \text{updated } \bar{c}_j) \end{array}$$

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

Third way: Edward Klotz (private communication) recommends a *double pricing* scheme that requries *one solve* and *no storing* of $\bar{c}$ and $\bar{d}$,

$$\begin{array}{lrcl} \text{updated} & \bar{c}_j & = & c_j - \pi^t A_{\cdot j} \\ \text{updated} & \bar{d}_j & = & \bar{d}_j - \sigma^t A_{\cdot j} \end{array}$$

where σ^t is generated without a solve by

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

which requires only a temporary storing of components π_i of π as it is computed to compute the corresponding components of σ_i^t. The above formula for updated $\sigma = \sigma^t$ is obtained by eliminating ρ from the relations:

$$\begin{array}{ll} \pi^t & = \pi^{t-1} + (\bar{c}_s/\bar{a}_{rs})\rho \\ \sigma^t & = \sigma^{t-1} + (\bar{d}_s/\bar{a}_{rs})\rho \, . \end{array}$$

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 j-th 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 $\max_i A_{ij}$ providing some $A_{ij} > 0$, or $\max_i \mid A_{ij} \mid$. Any of these will render column selection unit free.

2. The choice of incoming column s by the parameterization scheme appears to be at least as good a choice as the usual rule $s = \text{argmin } \bar{c}_j$.

3. Near dual degeneracy does not appear likely to cause the same "treading of water and getting nowhere" as does degeneracy in the primal because all near tieing columns are promising columns to enter the basis and should be considered.

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

5. Drawbacks

1. 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 could cause the value of the θ_{i+1} of the next partition to be reinitiated at a higher value than the last reduction of θ_{i+1} and therefore it is doubtful that one can prove convergence when the primal is stalled. Convergence for unstalled steps is, of course, guaranteed.

2. If the first or second way is used to do the updating, the cost per iteration requires only four more instructions per column to maintain row vectors $\bar{c}$ and $\bar{d}$ plus the cost of initially storing two starting vectors $\bar{c} = c$ and $\bar{d} = d$ so there should be only an insignificant increase in CPU time.

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 [4]. 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 [3].

All the experiments were run by Klotz on a DEC MicroVax 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 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 comparing iterative count. The CPU time shown for the parametric method may be misleading because none of the ways recommended earlier for doing the updating were used. Instead a slower (but easier to implement) double-solve double-pricing scheme was used. A better implementation is expected to show CPU time per iteration about equal to the iteration time per iteration of the regular simplex. If so, then the iteration ratio shown for the

test problems would be a better measure for comparing the relative efficiency of the two methods for very degenerate problems.

Comparative Iteration Count of Regular vs. Parametric Simplex Method on Nine Highly Degenerate Problems

PROBLEM NAME	Problem Size # ROWS	Problem Size #COLS.*	Iterations "REGULAR" SIMPLEX	Iterations PARAMETRIC SCHEME
KB2	46	41	65	80
DEGEN1	67	72	15	23
TUFF	371	587	1407	521
DEGEN2	445	534	1264	1053
WOODIP	486	2594	564	768
NZFRI	624	3521	10970	2268
WOODW	1089	8405	2381	2375
DEGEN3	1504	1818	11096	4828
CYCLE	2234	2857	3433	2403
	* excludes slacks			
Total Iterations			31195	14379
% Blocked			66%	66%

Parametric Iterations/Regular Iterations = .46

Parametric CPU / Regular CPU = .60

REFERENCES

1 Dantzig, George B., **Linear Programming and Extensions**, Princeton University Press, 1963, page 246, Theorems 2 and 3.

2 Gass, Saul I. and T.L. Saaty, "The Computational Algorithm for the Parametric Objective Function," **Naval Res. Logist. Quart.**, Vol. 2, No. 1, June 1955, pp. 39-45.

3 Klotz, Edward, Ph.D. thesis in preparation, 1988.

4 Lustig, Irvin, "Comparisons of Composite Simplex Algorithms", (Ph.D. thesis), TR SOL 87-8, Department of Operations Research, Stanford University, June 1987.

5 Murtagh, B.A. and M.A. Saunders, "MINOS 5.0 User's Guide," TR SOL 83-20, Department of Operations Research, Stanford University, December 1983.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: SOL 88-5	2. GOVT ACCESSION NO.: ADA193084
3. RECIPIENT'S CATALOG NUMBER:	
4. TITLE (and Subtitle): Making Progress During a Stall in the Simplex Algorithm	5. TYPE OF REPORT & PERIOD COVERED: Technical Report
	6. PERFORMING ORG. REPORT NUMBER:
7. AUTHOR(s): George B. Dantzig	8. CONTRACT OR GRANT NUMBER(s): N00014-85-K-0343
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Department of Operations Research - SOL Stanford University Stanford, CA 94305	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 1111MA
11. CONTROLLING OFFICE NAME AND ADDRESS: Office of Naval Research - Dept. of the Navy 800 N. Quincy Street Arlington, VA 22217	12. REPORT DATE: February 1988 13. NUMBER OF PAGES: pp. 7
	14. SECURITY CLASS. (of this report): UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:

16. DISTRIBUTION STATEMENT (of this Report)

This document has been approved for public release and sale; its distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Linear Programming, Parametric Simplex Method, Degeneracy Resolution

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

(please see other side)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Abstract

All the standard methods for avoiding cycling in the simplex algorithm set up a subprogram related to the blocking rows in the canonical form and look for an improving homogeneous solution. 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 [2]. The pivot-row choice among blocking rows, i.e., those tied for pivot, is arbitrary. A simple anti-cycling device is used which avoids dual degeneracy of the parameterized objective with probability one. Tests were run on nine highly degenerate practical test problems ranging in size from small to large. Using MINOS software [5], the standard simplex method required 31,195 iterations of which 20,504 (or 66%) were blocked pivots. Using the parametric scheme required 14,379 iterations, of which 9,481 or 66% were also blocked. The reduction in the number of iterations for this set of highly degenerate test problems is over 50%. CPU time reduction under the current way of implementation is 40%.

END

DATE

FILMED

DTIC

JULY 88