VOL. 15, NO. 3, FEBRUARY 2020 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

ANTI-CYCLING PROCEDURE IN DEGENERATE LINEAR PROGRAMMING PROBLEMS

Saad M. Salman
Department of Mathematics and Computer Application, Collage of Science, Al-Nahrain University, Baghdad, Iraq
E-Mail: saadmohsin58@yahoo.com

ABSTRACT

In some linear programming problems there are some situations are often difficult to solve, one such situation is the non-convergence of the problem, given that it has a finite or infinite optimal solution. Here we present a procedure for solving degenerate linear programming problem, based on the philosophy of constraints activities, by constructing a new rule for identifying an active pivot degenerate constraint, to perform the simplex iteration, preventing any cycling problem to be occur, in obtaining an optimal solution. A theorem has been presented with its proof, to show the convergence of the proposed approach after a finite number of simplex iterations. Good results had been obtained in terms of number of simplex iterations, compared with the other procedures, in solving such problems, by implementing several tested examples.

Keywords: degeneracy, cycling, simplex method, redundancy.

1. INTRODUCTION

An interesting question is raised in [3] about the role of degeneracy in the worst-case complexity of the randomized simplex algorithm, would not converge to the optimal solution, that is, it would be cycle, in which the simplex algorithm would keep repeating a degenerate basic feasible solution. If the repeated basis happened to be the optimal one, the simplex method would not so indicate. Problems that did not satisfy the NDP were easy to construct, but to find one that did not converge took some effort. The first instance of a linear-programming problem that was shown to cycle is the one constructed by [10]. All commercial LP software that we are applying rules for handling degeneracy, breaking ties, perturbation techniques, computations that enable the computer-based simplex algorithm to converge to an optimal solution even if the given problem exhibits classical cycling.

2. PREVIOUS METHODS

The following degeneracy linear programming problem has been considered

Minimize $c^T X$,

subject to: $AX \le B$,

$X \ge 0$.

Where $A \equiv m \times n$ matrix, $X = (x_1, x_2, \ldots, x_n)^T$, $B = (b_1, b_2, \ldots, b_m)^T$, with $b_r = 0$ for some r.

Degeneracy may become evident in the simplex method, when leaving variable is being selected in the iterative process, under the pivot column which determines the leaving variable if there is a tie (i.e. minimum ratio is the same for two or more rows), arbitrary selection of one of these variables may result in one or more variables becoming zero in the next iteration and the problem become degenerate, and in this case, it is usual that one or more of the subsequent pivots will be degenerate, and return to a case that has appeared before, in which case the simplex enters an infinite loop and never attains to the optimal solution, and this behavior is called "*Cycling*". Therefore, if the simplex method cycles, then all the pivots within the cycle must be degenerate, since the objective function value never changes. Hence, it follows that all the pivots within the cycle must have the same objective function value, i.e., all of these pivots must be degenerate. In practice, degeneracy is very common, but cycling is rare. In fact, it is so rare that most efficient implementations do not take precautions against it.

In [4], the cycling case was illustrated by considering the following example:

Minimize $Z = -\frac{3}{4}x_1 + 150x_2 - \frac{1}{50}x_3 + 6x_4$

Subject to: $\frac{1}{4}x_1 - 60x_2 - \frac{1}{25}x_3 + 9x_4 \le 0$

$\frac{1}{2}x_1 - 90x_2 - \frac{1}{50}x_3 + 3x_4 \le 0$

$x_3 \le 1$

all $x_j \ge 0$, $j = 1, 2,3,4$

with optimal values are: $x_1 = \frac{1}{25}, x_2 = 0, x_3 = 1 \text{ and } x_4 = 0$

Starting solving this problem, it had been taken 7 simplex iterations to reach to the same first solution again. See [4].

In [14], the perturbation method, by adding fixed and small random perturbation for each constraint, and smaller on each succeeding constraint, in which it turns out that the method produces a variant of the simplex method that never cycles.

In [6] & [7], a modified simplex method that do not cycle, by making a new pivoting rules for which the simplex method will definitely either reach an optimal solution or prove that no such solution exists, are presented. One of the modifications rules, is in the selecting choices, when there are ties in selecting the variable x_k to enter the basic solution and ties occur in selecting the variable x_j to be removed from the basic solution. In selecting x_k, we can choose any variable that will improve the value of the objective function, while the choice of x_j must correspond to a basis change that preserves feasibility, in which it does require a different and more computationally involved process for determining x_l.

In contrast, a very-easy-to-use anti-cycling procedure, requires simple decision rules. The modified rules, see [5], are the following:

- Among all candidates to enter the basic solution, select the variable has the lowest index.
- Among all candidates to leave the basic solution, select the variable has the lowest index.

One possible use of the lowest-index rules as an anti-degeneracy procedure; that is apply the rules to a problem after a number of iterations (say 2m) have been completed without a change in the value of the objective function. The test showed the standard rule was still much better on highly degenerate problems. In degenerate problems, we may stay at the same solution for a number of iterations, changing only the basis, and eventually moving to a better solution without cycling. In the above example, it had been taken 6 simplex iterations to reach an optimal solution $x_1 = \frac{1}{25}, x_2 = 0,\ x_3 = 1 \text{ and } x_4 = 0,$ havin minimum value of the objective function is -1/20.

3. PROPOSED APPROACH

Although the lowest-index rules are easy to apply, but still inefficient from the point of the number of simplex iterations in solving the problem, in which, tests had been done, where the standard rules and the lowest-index rules were compared showed that the standard rules required less number of iterations on their test problems, see [8].

Before we propose our approach, first, we present some essential definitions that are required in our approach, see [1], [2] & [6].

We are consider the feasible region Ω define as

$\Omega=\{X \in R^n\colon\ A_i^T X \le b_i, i \in I\}$, where $A_i^T X \le b_i$ is the i-th constraint.

The region represented by all but the *k-th* constraint is given by

$$\Omega_k = \{X \in R^n\colon\ A_i^T X \le b_i,\ \ i \in I\backslash\{k\}\}\ .$$

Definition 1: The constraint $A_k^T X \le b_k$ is said to be *inactive* in the description of Ω if $\Omega = \Omega_k$, otherwise is said to be active.

Definition 2: The *k-th* constraint is an $active\ at\ X_r$ if its distance from $\boldsymbol{X_r}$, is at minimum, i.e. *dis* $(X_r, A_k) = \min_i \parallel A_i - X_r \parallel,$ for all i.

In this paper, we are implemented the philosophy of active constraint amoning number of degenerate constraints, in order to overcome the cycling problem, by identifying such constraint, in order to be selected as a pivot constraint, in performing the simplex method. The correct identification of active constraints is important from both theoretically and practically point of view. Theoretically, the identification such active constraints is not difficult. However, as far as we are aware of, to date no technique can successfully to identify all active constraints, see [1], [2], [6], & [13]. To do this, we are presenting the following definitions.

Definition 3: The projection $P_i(X_r)$ of the point X_r onto the hyper plane $H_i = \{X \in R^n\colon\ A_i^T X_r = b_i\}$, is defined by

$$P_i(X_r) = X_r + A_i(b_i - A_i^T X_r).$$

Consequently, we have $\parallel P_i(X_r) - X_r \parallel = A_i(b_i - A_i^T X_r) = \mathrm{dis}(X_r, H_i)$.

In [2], a definition of local inactive nonlinear constraint is presented, which is of no use, in identifying whether the constraint is active or not, since local active constraint may be inactive in another local feasible region. Therefore, we prefer to define a local active constraint, which its existence is necessary to keep the hole feasible region of the problem unchanged, by setting $\boldsymbol{\delta}$>0, and define

$$U_\delta(x) = \{y \in R^n\colon \parallel y - X_r \parallel \le \delta\},$$

by using the usual definition of the distance function *dis(.,.)* between the point and set of linear equations.

Suppose that "ε" the infmum distance between the solution point X_r and the set of the constraints Ω at a local region, we can present the following definitions:

Definition 4: The constraint $A_k X_r \le 0$ is *locally active* at $\boldsymbol{X_r}$ if there exist an open set $\delta(\varepsilon)$, such that $A_k \in \Omega \cap \bar{\boldsymbol{\delta}}(\boldsymbol{\varepsilon})$ & $\Omega_{\boldsymbol{k}} \cap \bar{\boldsymbol{\delta}}(\boldsymbol{\varepsilon}) = \emptyset$,

where $\bar{\delta}(\varepsilon)$ is the closuer of δ.

The above definition can be stated into another way:

Definition 5: The constraint $A_k X_r \le 0$ is locally active at X_r, if for some ε, $\Omega \cap \bar{\delta}(\varepsilon) \ne \Omega_r \cap \bar{\delta}(\varepsilon)$. Otherwise, it is inactive.

If we denote the following:
x_j= Real variables (j=1, …,n),
s_i = Slack variables (i=1, …,m),
int (X_r)= Interior feasible point at the current iteration r,
dist (*int*(X_r),A_i)= The distance between the *int*(X_r) and the *i-th* constraint A_i .

Our procedure, start at any iteration with degenerate solution, with more than one degenerate constraint, by constructing the feasible interior point *int*(X_r) near the current extreme solution, as in the case of degenerate active, when the basis at the iteration *r* does not have full rank, and the active constraints are linearly dependent, see [2].

Now, we can state the following rule:

VOL. 15, NO. 3, FEBRUARY 2020 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

At any degenerate solution, if the constraint A_k corresponding to $\min_{b_i=0}\{dist(int(X_r), A_i)\}$, then the constraint A_k is an active.

The original proof that the simplex algorithm would converge to an optimal solution invoked the nondegenercy assumption, see [7], in which the simplex algorithm was guaranteed to converge to an optimal solution or show that the optimum was unbounded. The proof of our presented theorem is in the manner, presented in [14]:

Theorem

The modified simplex method for degeneracy problems, always terminated in less number of iterations, provided that the leaving variable is selected according to the above rule.

Proof:

The constructing of an interior feasible solution and applying the above rule in order to identify the pivot constraint, is prevented the simplex pivot iteration from cycling, and the iterations will terminated. while the other remain tests, is prevented any inefficient variable from entering the basis, since such variables are enter the basis and leaving the basis later, therefore the iterations of our modification method is converged either to optimal or unbounded solution, and terminated after a finite less number of iterations. ■

To demonstrate the performance of our simplex pivot rule, we are considering the same example used by other procedures as following:

Basis	b	x_1	x_2	x_3	x_4
		-3/4	150	-1/50	6
s_1	0	¼	-60	-1/25	9
s_2	0	½	-90	-1/50	3
s_3	1	0	0	1	0
$z_j\_c_j$	0	¾	-150	1/50	-6

I

Basis	b	x_1	x_2	x_3	x_4
		-3/4	150	-1/50	6
s_1	0	0	-15	-3/100	15/2
x_1	0	1	-180	-1/15	6
s_3	1	0	0	1	0
$z_j\_c_j$	0	0	-15	1/20	-21/2

II

Basis	b	x_1	x_2	x_3	x_4
		-3/4	150	-1/50	6
S_1	0	0	-15	0	15/2
x_1	1/25	1	-180	0	6
x_3	1	0	0	1	0
$z_j\_c_j$		0	-15	0	-21/2

Which is the same optimality values of variables.

4. NUMERICAL TESTED PROBLEMS

Now, for more performance satisfaction to our proposed approach, number of collected *degenerate LP* problems that exhibit classical cycling and ran each one using three popular and readily available *LP* software packages: *"MathLab* Version 6", to solved. All problems were solved correctly by each package. The problem by Beale was the first one given in terms of rational coefficients. For each problem, we note the number of iterations that returns the problem to its original form, that is, the number of simplex iterations it takes to cycle when the problem is solved by hand using the standard simplex pivot rules.

Problem 1 [7]

Minimize

$-2.2361x_4+2x_5+4x_7+3.6180x_8+3.236x_9+3.6180x_{10}$
$+0.764x_{11}$

Subject to

$x_1=1$

x_2 $+0{:}3090x_4$ $-0{:}6180x_5$ $-0{:}8090x_6$ $-0{:}3820x_7$ $+0{:}8090x_8$ $+0{:}3820x_9$
$+0.3090x_{10}$ $+0{:}6180x_{11}=0$

x_3 $+1.4635x_4$ $+0.3090x_5$ $+1.4635x_6$ $-0.8090x_7$ $-0.9045x_8$ $-0.8090x_9$ $+0.4635x_{10}$ $+0.309x_{11}=0$

$x_j \geq 0$

Solution: $x_1 = 1$; $x_j = 0$ (j = 2,…..,11); Minimum = 0; Cycle = 10 "MathLab Version 6"; no cycle "proposed method ".

Problem 2 [1]

Maximize $x_3 - x_4 + x_5 - x_6$

Subject to

$x_1 + 2x_3 - 3x_4 - 5x_5 + 6x_6 = 0$

$x_2 + 6x_3 - 5x_4 - 3x_5 + 2x_6 = 0$

$3x_3 + x_4 + 2x_5 + 4x_6 + x_7 = 1$

$x_{j \geq} 0$

Solution: $x_1 = 2.5$; $x_2 = 1.5$; $x_5 = 0.5$; Maximum=0.5; Cycle=6 "MathLab Version 6"; no cycle "proposed method ".

Problem 3 [11]

Maximize $x_3 - x_4 + x_5 - x_6$

Subject to

$x_1+x_3-2x_4-3x_5+4x_6=0$

$x_2+4x_3-3x_4-2x_5+x_6=0$

$x_3+x_4+x_5+x_6+x_7=1$

$xj \geq 0$

Solution: $x_1=3$; $x_2=2$; $x_5=1$; Maximum=1; Cycle=6 "MathLab Version 6"; no cycle "proposed method ".

Problem 4 [10]

Minimize $2x_1+4x_4+4x_6$

Subject to

$x_1-3x_2-x_3-x_4-x_5+6x_6=0$

$2x_2+x_3-3x_4-x_5+2x_6=0$

$x_j \geq 0$

 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

Solution: All variables =0;Minimum=0;Cycle=6"MathLab Version 6"; no cycle "proposed method ".

Problem 5 [11]

Minimize $-2x_3-2x_4+8x_5+2x_6$

Subject to

$x_1-7x_3-3x_4+7x_5+2x_6=0$

$x_2+2x_3+x_4-3x_5-x_6=0$

$x_j\geq 0$

Solution: All x_i=0; Minimum=0 ;Cycle=6"MathLab Version 6"; no cycle "proposed method ".

Problem 6 [15]

Maximize $3x_1-80x_2+2x_3-24x_4$

Subject to

$x_1-32x_2-4x_3+36x_4+x_5=0$

$x_1-24x_2-x_3+6x_4+x_6=0$

$x_j\geq 0$

Solution: Unbounded; Cycle=6"MathLab Version 6"; no cycle "proposed method ".

5. CONCLUSIONS

Based on [4], [6] and [7], one can see from tableau I & II, applying lowest-index rules at step 1, no change in the solutions path, step II & III. While applying lowest-index rules at step III, the solutions path in tableau II is different than the solutions path in tableau I, but more simplex iterations are needed to be performed in applying lowest-index rules to reach an optimal solution. While as we have seen from our tableau in this paper, applying our suggestion rule in selecting the pivot constraint to enter the basis, will change the solutions path direction to reach an optimal solution, in two simplex iterations only, comparing with six iterations, by applying the lowest-index rules, after the cycling occurred. We believe that in the degeneracy linear system, the performing of the lowest-index rules, does not considering the selected pivot constraint, whether, active or not, among several degenerate constraints, would cause a different path solution to reach optimal solution, while our simplex pivot rule procedure did better identification of path solution in less number of iterations.. Indeed, such result, required more studies to be done on different structure of degeneracy problems to verify a theory of our pivot rule.

REFERENCES

[1] Audet C. and Dennis Jr. 2003. Analysis of generalized pattern searches. SIAM J. Optim. 13(3): 889-903.

[2] Abramson M. A., *et al.* 2007. Pattern search in the presence of degeneracy. Online at: (mark.abrason@afit.eduhttp).

[3] Balinski, M. L. & *et al.* 1987 Nobs. On the average length of lexicographic paths. Mathematical Programming.

[4] Beale E. M. L. 1955. Cycling in the Dual Simplex Algorithm, Naval Research Logistics, Quarterly. 2(4).

[5] Bland, R. G. 1977. New finite pivoting rules. Mathematics of Operations Research 2, 103-107.

[6] Caron R. J., *et al*. 1989. A degenerate extreme point strategy for the classification of linear constraints as redundant or necessary. Jr. Opt. Theory & Aplcs. 62(22): 225-237.

[7] Gass S. I. 1985. Linear Programming, Methods and Applications. 5-th edition, McGraw-Hill Book Company.

[8] Gill P. & *et al.* 1991. Numerical Algebra and Optimization. Addison-Wesley.

[9] Graves R. L. & Wolfe. 1963 Recent Advances in Mathematical Programming. McGraw-Hill Book Company.

[10] Hoffman A. J. 1953. Cycling in the simplex algorithm. Washington, DC: National Bureau of Standards.

[11] Marshall K. T. & Suurballe J. W. 1969. A note on cycling in the simplex method. Naval Research Logistics Quarterly. 16(1): 121-37.

[12] Megiddo N. 1986. A note on degeneracy in linear programming. Math. Progr. 35, 365-367.

[13] Price C. J. & Coop I. D. 2003. Frames and grids in unconstrained and linearly constrained optimization. A nonsmooth approach. SIAM J. Optim. 14(2): 415-438.

[14] Robert J. V. 2008. Linear Programming, Foundations and Extensions. 3-rd edition. Springer Company.

[15] Solow D. 1984. Linear programming: an introduction to finite improvement algorithms. Amestrdam: North-Holland.