Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Computers & Operations Research 31 (2004) 303–311

computers & operations research

www.elsevier.com/locate/dsw

Note

Cycling in linear programming problems

Saul I. Gass[a,*], Sasirekha Vinjamuri[b]

[a] *Robert H. Smith School of Business, University of Maryland, College Park, MD 20742, USA*
[b] *Department of Mathematics, University of Maryland, College Park, MD 20742, USA*

Received 1 August 2002; accepted 1 November 2002

Abstract

We collected and analyzed a number of linear programming problems that have been shown to cycle (not converge) when solved by Dantzig's original simplex algorithm. For these problems, we were interested in whether or not some of the more popular linear programming solvers would find an optimal solution. This technical note reviews the results of our analysis and some related topics on cycling.

Scope and purpose

Verification of algorithmic based software can often be a difficult matter as test problems for checking anomalous situations are often difficult to construct. For linear programming software, one such situation that has to be guarded against is the nonconvergence of a problem, given that it has a finite or infinite optimal solution. Here we present 11 linear-programming problems that have been shown to cycle when solved by the original algorithmic rules of the simplex method. All of these problems converged to the proper optimal solution when solved by three popular simplex solvers. For those interested in constructing other problems that cycle, we also present necessary conditions that must hold with respect to the problem's dimensions.

Keywords: Linear programming; Cycling; Degeneracy

1. Introduction

The original proof that the simplex algorithm would converge to an optimal solution invoked the famous non-degeneracy assumption (NDA) (Dantzig [1]). That is, given that the linear-programming (LP) problem in question has feasible solutions, then every basic feasible solution has all of its

* Corresponding author. Tel.: +1-301-405-2208; fax: +1-301-405-8655.
E-mail address: sgass@rhsmith.umd.edu (S.I. Gass).

0305-0548/04/$ - see front matter © 2003 Elsevier Ltd. All rights reserved.
doi:10.1016/S0305-0548(02)00226-5

basic variables strictly positive. For such a problem, Dantzig's prescription for the simplex algorithm was guaranteed to converge to an optimal solution or show that the optimum was unbounded. (The reader is referred to [1] or Gass [2] for a description of the simplex algorithm.) If a problem does not satisfy the NDA (and it is difficult to discern this prior to solving a problem), then there is the possibility that the simplex algorithm would not converge to the optimal solution, that is, it would cycle. Here "to cycle" means that 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 NDA 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 Hoffman [3]. This problem, which is discussed below, is an unusual one in that its coefficients are given in terms of trigonometric functions. Hoffman in his article (and when asked in person) does not "… recall any considerations that led to the construction of the example …" [3]. We note that a feasible basis that has a single degeneracy cannot cause a cycle and, thus, the NDA can be weakened to allow for single degenerate basic feasible solutions.

In discussing the simplex algorithm and cycling, we need to differentiate between *classical cycling* and *computer cycling* (Gass [4]). Classical cycling arises when the data of the problem are given in decimal notation and can be expressed as rational fractions. Computations are performed without loss of accuracy or round-off error; that is, the data are always transformed from rational fractions to rational fractions. A standard version of the primal simplex algorithm is used in which the decision rules for selecting a variable to enter or leave the basis are based on Dantzig's original formulation, e.g., when ties exist, select the variable (to enter or leave) that has the smallest index. Computer cycling is an application of the simplex method that does not preserve rational transformations and uses experiential based criteria for selecting entering or leaving variables, tolerance values for what is a zero, scaling, and numerical stability procedures. A problem written on paper and solved by hand (classical mode) is not the same problem entered into a computer and solved by a computer-based simplex algorithm (computer mode). (A similar discussion can be applied to the dual simplex method and for the special case of a transportation problem. See Beale [5] for a problem that cycles under dual simplex rules for selecting the outgoing and incoming variables, and Gassner [6] for examples of cycling in the transportation and assignment problems.)

It should be clear that if a problem exhibits classical cycling, the chances are that it will not exhibit computer cycling, and, vice-versa. In the early days of linear programming, there was concern and interest in finding a "real-world" problem that cycled. The oil refinery problem considered by Charnes, Cooper and Mellon [7] was an early practical problem that exhibited degeneracy, but converged without any difficulty. See [4] for further discussion along these lines.

All commercial LP software that we are aware of apply rules for handling degeneracy, breaking ties, perturbation techniques, and composite primal and dual computations that enable the computer-based simplex algorithm to converge to an optimal solution even if the given problem exhibits classical cycling. In the past, we found one such problem that cycled when operated on by a popular LP software system. When this was brought to the attention of the software developers, the next version of the software solved the problem in question. We were not made privy to the changes that were made. See Benichou et al. [8] for an example of how mathematical programming systems guard against cycling.

2. Facts about LP problems with respect to cycling

To date, a number of LP problems have been constructed that exhibit classical cycling. We list and discuss those that we have found in Section 4 (Vinjamuri [9]). If one attempts to construct such a problem, certain characteristics must be kept in mind. The form of the LP problem is

$$\begin{aligned}&\text{Minimize}\quad cy\\&\text{subject to}\\&\quad Ix + Ay = b\\&\quad (x, y) \geqslant 0\end{aligned} \tag{1}$$

where I is an $(m \times m)$ identity matrix, A is an $[(m \times n - m)]$ matrix, $(m < n)$, x and y are column vectors, c is a row vector, and b is a column vector. The basic simplex algorithm is applied to (1) in which ties for the pivot column are broken by selecting the leftmost (smallest index) column, and ties for the pivot row are broken by selecting the topmost row (lowest index).

Under the above rules, using the primal simplex, for cycling to occur we must have $m \geqslant 2$, $n \geqslant m+3$, and $n \geqslant 6$. For cycling to occur at a nonoptimal point we must have $m \geqslant 3$, $n \geqslant m+3$, and $n \geqslant 7$. All bounds are sharp (Marshall and Suurballe [10]). An LP problem with only two nonbasic variables cannot cycle; a cycling example must have at least six variables, at least two equations, and at least three nonbasic variables [10]. A basic feasible solution with a single degeneracy cannot cause a cycle (due to L. Goldstein, Gass [11]). The minimum length of a cycle is six iterations (Yudin and Gol'shtein [12]).

3. The Hoffman problem [3]

Conceived by Hoffman in 1951, this is the first linear-programming problem that was shown to cycle under the standard simplex iteration rules. It thus demonstrated that anti-cycling or anti-degeneracy procedures would be necessary to ensure that the theoretical simplex method could be interpreted as an algorithm that always returned a solution to the problem, that is, stopped after a finite number of steps with a statement of infeasibility or optimal bounded or unbounded solution.

Hoffman's problem:

Minimize

$$[(\cos\varphi - 1)/\cos\varphi]x_4 + wx_5 + 2wx_7 + 4\sin^2\varphi x_8 + w(2 - 4\cos^2\varphi)x_9 + 4\sin^2\varphi x_{10}$$
$$+w(1 - 2\cos\varphi)x_{11}$$

subject to

$$x_1 = 1$$

$$x_2 + \cos\varphi x_4 - w\cos\varphi x_5 + \cos 2\varphi x_6 - 2w\cos^2\varphi x_7 + \cos 2\varphi x_8 + 2w\cos^2\varphi x_9 + \cos\varphi x_{10}$$
$$+w\cos\varphi x_{11} = 0$$

$$x_3 + [(\tan\varphi \sin\varphi)/w]x_4 + \cos\varphi x_5 + [(\tan\varphi \sin 2\varphi)/w]x_6 + \cos 2\varphi x_7 - [2\sin^2\varphi/w]x_8$$
$$+\cos 2\varphi x_9 - [(\tan\varphi \sin\varphi)/w]x_{10} + \cos\varphi x_{11} = 0$$
$$x_j \geqslant 0 \text{ for } j = 1, \ldots, 11.$$

Here we set $\varphi = 2/5\pi$; w is any number greater than $(1 - \cos\varphi)/(1 - 2\cos\varphi)$.

The standard simplex algorithm applied to the above example cycles in 10 iterations without indicating that the starting basic feasible solution $(x_1 = 1, x_2 = 0, x_3 = 0)$ and all subsequent bases yield the minimum value of zero for the objective function. Lee [13] discusses and suggests a possible genesis of Hoffman's problem from both algebraic and geometric points-of-view. As noted in [3], Gass and Jacobs showed the set of 2×2 matrices formed by the second and third rows and successive pairs of columns form a cyclic set. That is, with

$$A = \begin{pmatrix} \cos\varphi & -w\cos\varphi \\ (\tan\varphi \sin\varphi)/w & \cos\varphi \end{pmatrix}$$

formed by the second and third rows with the fourth and fifth columns, we have A^2 corresponding to (2×2) matrix associated with the sixth and seventh columns, A^3 with the eighth and ninth columns, A^4 with the tenth and eleventh columns, and $A^5 = I$ [9]. It would appear that such matrices of finite order can be used to construct other cycling problems.

4. Cycling problems

We collected a number of LP problems that exhibit classical cycling and ran each one using three popular and readily available LP software packages: LINDO, C-Plex, and the Excel spreadsheet solver. 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 iterations it takes to cycle when the problem is solved by hand using the standard simplex rules. When finding an optimal solution for these cycling problems using LP software, the iteration count does not reflect the cycle count. For example, the problem of Section 4.3 cycles in 6 iterations, but is solved by LINDO in one iteration.

Some of the problems have multiple optimal solutions; we list only one.

4.1. Hoffman—(See Section 3 for original statement of the problem)

$$\phi = \tfrac{5}{2}\pi$$
$$w > (1 - \cos(\phi))/(1 - 2)\cos(\phi)) \approx 1.8090$$
$$w = 2$$

Minimize

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

subject to

$$x_1 = 1$$

$$x_2 + 0.3090x_4 - 0.6180x_5 - 0.8090x_6 - 0.3820x_7 + 0.8090x_8 + 0.3820x_9 + 0.3090x_{10}$$
$$+0.6180x_{11} = 0$$

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

$$x_j \geqslant 0$$

Solution : $x_1 = 1,\ x_j = 0\ (j = 2, \ldots, 11)$; Minimum $= 0$; Cycle $= 10$.

4.2. *Beale example (Gass [2])*

Minimize $\quad -3/4x_1 + 150x_2 - 1/50x_3 + 6x_4$

subject to

$$\tfrac{1}{4}x_1 - 60x_2 - \tfrac{1}{25}x_3 + 9x_4 + x_5 = 0$$

$$\tfrac{1}{2}x_1 - 90x_2 - \tfrac{1}{50}x_3 + 3x_4 + x_6 = 0$$

$$x_3 + x_7 = 1$$

$$x_j \geqslant 0$$

Solution : $x_1 = \frac{1}{25},\ x_3 = 1;\ x_5 = \frac{3}{100}$; Minimum $= -\frac{1}{20}$; Cycle $= 6$.

4.3. *Yudin and Gol'shtein [12]*

Maximize $\quad x_3 - x_4 + x_5 - x_6$

subject to

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

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

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

$$x_j \geqslant 0$$

Solution : $x_1 = 2.5,\ x_2 = 1.5,\ x_5 = 0.5$; Maximum $= 0.5$; Cycle $= 6$.

4.4. Yudin and Gol'shtein [12]

$$\text{Maximize} \quad x_3 - x_4 + x_5 - x_6$$

subject to

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

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

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

$$x_j \geqslant 0$$

Solution : $x_1 = 3,\ x_2 = 2,\ x_5 = 1$; Maximum $= 1$; Cycle $= 6$.

4.5. Kuhn example (Balinski and Tucker [14])

$$\text{Minimize} \quad -2x_4 - 3x_5 + x_6 + 12x_7$$

subject to

$$x_1 - 2x_4 - 9x_5 + x_6 + 9x_7 = 0$$

$$x_2 + 1/3x_4 + x_5 - 1/3x_6 - 2x_7 = 0$$

$$x_3 + 2x_4 + 3x_5 - x_6 - 12x_7 = 2$$

$$x_j \geqslant 0$$

Solution : $x_1 = 2,\ x_4 = 2,\ x_6 = 2$; Minimum $= -2$; Cycle $= 6$.

Note: The third equation has been added as a bound on the objective function. Without it, the two equation problem is unbounded and cycles in six iterations.

4.6. Marshall and Suurballe [10]

$$\text{Minimize} \quad 2x_1 + 4x_4 + 4x_6$$

subject to

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

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

$$x_j \geqslant 0$$

Solution : All variables $= 0$; Minimum $= 0$; Cycle $= 6$.

4.7. Marshall and Suurballe [10]

$$\text{Minimize} \quad -0.4x_5 - 0.4x_6 + 1.8x_7$$

subject to

$$x_1 + 0.6x_5 - 6.4x_6 + 4.8x_7 = 0$$

$$x_2 + 0.2x_5 - 1.8x_6 + 0.6x_7 = 0$$

$$x_3 + 0.4x_5 - 1.6x_6 + 0.2x_7 = 0$$

$$x_4 + x_6 = 1$$

$$x_j \geqslant 0$$

Solution : $x_1 = 4$, $x_2 = 1$, $x_5 = 4$, $x_6 = 1$; Minimum $= -2$; Cycle $= 6$.

4.8. Solow [15]

$$\text{Minimize} \quad -2x_3 - 2x_4 + 8x_5 + 2x_6$$

subject to

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

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

$$x_j \geqslant 0$$

Solution : All $x_i = 0$; Minimum $= 0$; Cycle $= 6$.

4.9. Sierksma [16]

$$\text{Maximize} \quad 3x_1 - 80x_2 + 2x_3 - 24x_4$$

subject to

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

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

$$x_j \geqslant 0$$

Solution : Unbounded; Cycle $= 6$.

4.10. Chvátal [17]

Minimize $10x_1 - 57x_2 - 9x_3 - 24x_4$

subject to

$$0.5x_1 - 5.5x_2 - 2.5x_3 + 9x_4 + x_5 = 0$$

$$0.5x_1 - 1.5x_2 - 0.5x_3 + x_4 + x_6 = 0$$

$$x_1 + x_7 = 1$$

$$x_j \geqslant 0$$

Solution : $x_1 = 1,\ x_3 = 1,\ x_5 = 2$; Minimum $= 1$; Cycle $= 6$.

4.11. Nering and Tucker [18]

Maximize $-3x_2 + x_3 - 6x_4 - 4x_6$

subject to

$$x_1 + x_2 + 1/3x_5 + 1/3x_6 = 2$$

$$9x_2 + x_3 - 9x_4 - 2x_5 - 1/3x_6 + x_7 = 0$$

$$x_2 + 1/3x_3 - 2x_4 - 1/3x_5 - 1/3x_6 + x_8 = 2$$

$$x_j \geqslant 0$$

Solution : Unbounded; Cycle $= 6$.

5. Summary

In the real world of computer-based solutions, the question of cycling in linear-programming problems has been resolved for all practical purposes. Today's commercial mathematical programming systems (MPS) have been designed to overcome any chance that a degenerate linear-programming problem would cycle. There have been and probably will be instances when an MPS will not converge, but such situations will certainly be due to problems of numerical instability and related difficulties, not degeneracy. Of course, for large problems, unless they are solved in terms of rational arithmetic, we will never really know if a problem that exhibits computer cycling will also be subject to classical cycling.

References

[1] Dantzig G. Linear programming and extensions. Princeton, NJ: Princeton University Press, 1963.
[2] Gass SI. Linear programming: methods and applications, 5th ed. New York: McGraw-Hill Book Company, 1985.
[3] Hoffman AJ. Cycling in the simplex algorithm, 1953. Washington, DC: National Bureau of Standards, 1953.
[4] Gass SI. Comments on the possibility of cycling with the simplex method. Operations Research 1979;27(4):848–52.

[5] Beale EML. Cycling in the dual simplex method. Naval Research Logistics Quarterly 1955;2(4):269–75.
[6] Gassner BJ. Cycling in the transportation problem. Naval Research Logistics Quarterly 1964;11(1):43–57.
[7] Charnes A, Cooper WW, Mellon R. Blending aviation gasolines—a study in programming interdependent activities in an integrated oil company. Econometrica 1952;20(2):135–59.
[8] Benichou M, Gauthier JM, Hentges G, Ribière G. The efficient solution of large-scale linear programming problems—some algorithmic techniques and computational results. Mathematical Programming 1977;13:280–322.
[9] Vinjamuri S. Cycling in linear programming problems. MA Thesis, Department of Mathematics, University of Maryland, College Park, MD, 1998.
[10] Marshall KT, Suurballe JW. A note on cycling in the simplex method. Naval Research Logistics Quarterly 1969;16(1):121–37.
[11] Gass SI. Encounters with degeneracy: a personal view. Annals of Operations Research 1993;47:335–42.
[12] Yudin DB, Gol'shtein EG. Linear programming. Israel Program of Scientific Translations, Jerusalem, 1965.
[13] Lee J. Hoffman's circle updated. SIAM Review 1997;39:98–105.
[14] Balinski ML, Tucker AW. Duality theory of linear programs—a constructive approach with applications. SIAM Review 1997;11:3.
[15] Solow D. Linear programming: an introduction to finite improvement algorithms. Amsterdam: North-Holland, 1984.
[16] Sierksma G. Linear and integer programming, 2nd ed. New York: Marcel Dekker, Inc., 1996.
[17] Chvátal V. Linear programming. New York: W. H. Freeman and Company, 1983.
[18] Nering ED, Tucker AW. Linear programs and related problems. Boston, MA: Academic Press, 1993.

Saul I. Gass received his B.S. in Education and M.A. in Mathematics from Boston University, and his Ph.D. in Engineering Science/Operations Research from the University of California, Berkeley. He is currently Professor Emeritus at the Robert H. Smith School of Business, University of Maryland, College Park. Included in his many publications are the text *Linear Programming* (fifth edition), the book *An Illustrated Guide to Linear Programming*, and the text *Decision Making*, *Models and Algorithms*. He is co-editor of the *Encyclopedia of Operations Research and Management Sciences*. His research interests include: linear programming, large-scale systems, model validation and evaluation, game theory, multi-objective decision analysis, and the application of operations research methodologies.

Sasirekha Vinjamuri received her B.S. and M.A. in Mathematics from the University of Madras. She continued her studies at the University of Maryland, College Park, where she received an M.A. in Applied Mathematics.