A Counterexample to H. Arsham: "Initialization of the Simplex Algorithm: An Artificial-Free Approach"

Andreas Enge and Petra Huhn
Lehrstuhl für Diskrete Mathematik,
Optimierung und Operations Research
Universität Augsburg
86135 Augsburg
Deutschland
{enge, huhn}@math.uni–augsburg.de

September 28, 1998

Abstract

In [3] and [4] Arsham presents a new Phase 1 algorithm for the simplex method of linear programming, which allegedly obviates the use of artificial variables. He claims in [4] that the new algorithm will terminate successfully or indicate the infeasibility of the problem after a finite number of iterations and precises in [3] that the number of iterations is at most number of constraints.

Providing this claim to be true, we point out some consequences for the complexity of the simplex method. We give a counterexample, where Arsham's algorithm declares the infeasibility of a feasible problem.

Keywords: linear programming, phase 1, complexity
AMS-Subject Classification: 90C05, 90C60, 68Q25

1 Initialization of the Simplex Algorithm

In order to solve a general linear program with any variant of the simplex algorithm [6] it is necessary to find a feasible vertex and hence a feasible basic solution of the polyhedron specified by the constraints, which part of the algorithm is usually referred to as "Phase 1". In [3] and [4] Arsham describes a new Phase 1 algorithm which allegedly gets around the introduction of artificial variables, and moreover needs at most as many pivot steps as there are constraints.[1] For a better comparison with the usual approaches we briefly relate the two most common Phase 1 algorithms in the following subsections before turning our attention to Arsham's technique.

Arsham considers a general LP of the form

$$\max d^T x \text{ subject to } Fx \leq f,\ Gx \geq g,\ Hx = h,\ x \geq 0$$

with f, g, $h \geq 0$. He uses well-known transformation techniques to obtain a problem in *standard form*

$$\text{(LP)} \quad \max c^T x \text{ subject to } Ax = b,\ x \geq 0,$$

where A is an $m \times n$ matrix of full row rank and $b \geq 0$. Hence we can assume in the following that an LP is given in standard form, and Phase 1 consists of finding a feasible basic solution to such a problem.

1.1 Solving an auxiliary problem

Probably the most common approach to Phase 1 is introducing *artificial variables*, one for each equality constraint, and solving the auxiliary problem

$$\text{(AP)} \quad \max \mathbb{1}^T Ax \text{ subject to } Ax + y = b,\ x, y \geq 0,$$

where $\mathbb{1} = (1, \ldots, 1)^T$ is the all-one-vector. Note that a feasible solution to (AP) is given by $x = 0$, $y = b$, and that the objective "max $\mathbb{1}^T Ax$" is equivalent to "min $\mathbb{1}^T y$", so that (AP) is bounded. The auxiliary problem is solved by the usual simplex algorithm. (It is possible, however, to delete the column of an artificial variable as soon as it has left the basis.) (LP) is feasible if and only if the optimal objective value of (AP) is $\mathbb{1}^T b$, which means that all artificial variables are zero. Then, if the optimal basis for (AP) contains no artificial variable, it is a feasible basis for (LP). Otherwise, since the artificial variables are zero and the matrix A has full rank, it is possible to pivot all artificial variables out of the basis to obtain an initial basis for (LP).

1.2 The Big-M-Method

An obvious drawback of the auxiliary problem of the previous section is that it does not take into account the objective function we are actually interested in maximising, and outputs an arbitrary vertex of the feasibility region. It would be desirable, however, to find in Phase 1 a basic solution which has

[1] This complexity result is announced in the abstract of [3]: "This phase terminates successfully (or indicates the infeasibility of the problem) with a finite number of iterations, which is at most equal to the number of constraints." and Lemma 1 in [3]: "By following Steps 3–5 a complete basic variable set can always be genereated (with at most m iterations), provided the feasible region is not empty."

chances of being closer to the optimum. This is exactly the purpose of the Big-M method. Again, artificial variables are introduced, but the objective function is more or less preserved. Indeed we consider the problem

$$(\mathrm{LP}(M)) \quad \max c^T x - M\mathbb{1}^T y \text{ subject to } Ax + y = b,\ x, y \geq 0.$$

Here, M can either be set numerically as a "sufficiently big" constant or be treated symbolically as a constant "bigger than any real number" such that $\alpha_1 M + \beta_1 \leq \alpha_2 M + \beta_2$ holds exactly if (α_1, β_1) is lexicographically less or equal (α_2, β_2) for real numbers α_1, β_1, α_2, β_2. Again, (LP(M)) is feasible by letting $x = 0$, $y = b$. If (LP(M)) has an optimal solution with all artificial variables equal to zero, then this is an optimal solution to (LP); if (LP(M)) is unbounded with an extremal free direction in which the objective grows and which does not involve any artificial variables, then (LP) is unbounded; in any other case, (LP) is infeasible.

The high negative coefficients in front of the artificial variables assure that the basic aim of this Phase 1 algorithm is to eliminate these variables from the basis, without losing the original objective function out of view. Again, it is possible to delete the column corresponding to an artificial variable as soon as it has left the basis.

1.3 Arsham's Approach

The idea of the new Phase 1 algorithm is to start with an empty basis, represented by the tableau

$$\begin{array}{c|c} \overline{A} & \overline{b} \\ \hline \overline{c} & \end{array}$$

with $\overline{A} = A$, $\overline{b} = b$ and $\overline{c} = c$. Then as long as this is possible, the basis is augmented by a new variable. We describe the process in detail in our words:

1. If the basis is full, Phase 1 is finished successfully, and the optimization process of Phase 2 can be started. Otherwise mark all columns as "unexamined" and go to Step 2).

2. Pivot column choice: Choose the as yet unexamined non-basic column s with the biggest reduced cost coefficient $\overline{c}_s$, be it negative or not. If there are ties, choose the first possible column. If none such column exists, state the infeasibility of the problem and stop.

3. Pivot row choice: If there is a row r, not yet occupied by a basic variable, such that $\overline{a}_{rs} > 0$ and

$$\frac{\overline{b}_r}{\overline{a}_{rs}} = \min \left\{ \frac{\overline{b}_i}{\overline{a}_{is}} \ :\ \overline{a}_{is} > 0,\ i = 1, \ldots, m \right\},$$

then the basis is augmented by the variable x_s in row r, and Gauß–Jordan row operations turn the s-th column into the r-th unit vector. Then go to Step 1). (If there are ties for the pivot row, choose the first possible one.) If no row exists with the properties mentioned above mark the column s as "examined" and go to Step 2).

Unfortunately, there are cases where the algorithm declares a feasible problem infeasible; we shall give a counterexample in Section 3.

1.4 Where Are the Artificial Variables?

Arsham claims that his algorithm "obviates the use of artificial variables" ([3] and [4], in the abstracts). This is true in a very literal sense, but the artificial variables are still present implicitly: Notice that the initial tableau

$$\begin{array}{c|c} A & b \\ \hline c & \end{array}$$

corresponds to the tableau obtained from the problem

$$\text{(LP')} \quad \max c^T x \text{ subject to } Ax + y = b,\ x, y \geq 0$$

for the basis made up by the artificial variables, except that the unit matrix in the basic columns is omitted. Now augmenting the basic variable set in Arsham's algorithm corresponds to pivoting an artificial variable out of the basis for (LP') and deleting the corresponding column afterwards. Hence Arsham's algorithm is very similar to Phase 1 via an auxiliary problem or the Big-M method with the pivoting rule: "Use Dantzig's column rule as far as possible, but don't bother about negative reduced cost coefficients, and perform pivot steps only that change a basic artificial variable against a non-basic original variable." The main difference is that the original objective function is kept instead of choosing an objective that encourages the elimination of artificial variables; this is made up for by not allowing pivot steps that exchange two original or two artificial variables.

2 The Complexity of Linear Programming

Shortly after Dantzig's discovery of the simplex method researchers set for the quest of the polynomial time pivot rule. Strange enough, the apparent practicability of the simplex algorithm is in striking contrast to the theoretical results obtained so far: A great deal of pivot rules have been proved to require an exponential number of pivot steps for special problem classes (see [10]), and the behavior of the other rules is essentially unknown. Part of the mystery is resolved by probabilistic analyses (see [5, 2, 1, 8]).

It was not until 1979 that the polynomial time solvability of linear programs was shown, and remarkably by algorithms that have not much in common with the simplex algorithm [7, 9]. Their running times are polynomial in the coding length of the problem, but they depend on the size of the numbers involved.

Arsham claims that his Phase 1 algorithm "terminates successfully (or indicates the infeasibility of the problem) after a finite number of iterations, which is at most equal to the number of constraints" ([3], in the abstract and Lemma 1), respectively, "is finite" ([4], Lemma 1), and indeed his "proof" would imply that the number of pivot steps is bounded by the number of constraints.

We propose to show that this would lead to a strongly polynomial pivot rule for the simplex algorithm, i. e. the number of pivot steps would be bounded by a polynomial in m and n, independently of the size of the occurring numbers. Precisely, if m is the number of constraints and n the number of variables in (LP), then $O(m+n)$ pivot steps would be required.

We assume that (LP) is feasible; a possible infeasibility could be detected by Arsham's algorithm in at most m steps. Considering the dual problem

$$\text{(DP)} \quad \min b^T y \text{ subject to } A^T y \leq c,$$

which can easily be rewritten in standard form, we set up a new problem (LDP) by joining the constraints of (LP) and (DP) and adding the constraint

$$c^T x - b^T y = 0.$$

The choice of objective function does not matter. By the Strong Duality Theorem, if (LDP) is infeasible, then (LP) is unbounded; a free direction on which the objective grows infinitely can be obtained as the optimal solution of the auxiliary problem

$$\max c^T d \text{ subject to } Ad = 0,\ d \geq 0,\ c^T d \leq 1.$$

If (LDP) is feasible, then the x-coordinates of a feasible point of (LDP) yield an optimal solution of (LP). But the feasibility of (LDP) could be checked by the new Phase 1 algorithm in at most $m+n+1$ pivot steps.

3 A Counterexample

An example of a feasible problem which is declared infeasible by Arsham's algorithm is given by the following problem:

$$\begin{aligned}&\max 3x_1 + x_2 - 4x_3\\ &\text{ subject to } x_1 + x_2 - x_3 = 1\,,\ x_2 \geq 2, x \geq 0\,.\end{aligned}$$

After adding a surplus variable we get:

$$\begin{aligned}&\max 3x_1 + x_2 - 4x_3\\ &\text{ subject to } x_1 + x_2 - x_3 = 1\,,\ x_2 - s_1 = 2, x_i, s_1 \geq 0\,.\end{aligned}$$

The algorithm starts with the following tableau:

$$\begin{array}{c|cccc|cl} & x_1 & x_2 & x_3 & s_1 & & \\ \hline ? & (1) & 1 & -1 & 0 & 1 & \leftarrow \\ ? & 0 & 1 & 0 & -1 & 2 & \\ \hline & 3 & 1 & -4 & 0 & & \\ & \uparrow & & & & & \end{array} \tag{1}$$

As the variable x_1 has the biggest reduced cost coefficient, the first column is chosen as the pivot column. Its entry in the second row is 0, so the first row becomes the pivot row. Hence, x_1 enters the basis and we perform the corresponding Gauss-Jordan row operations to obtain the following tableau:

$$\begin{array}{c|cccc|c} & x_1 & x_2 & x_3 & s_1 & \\ \hline x_1 & 1 & 1 & -1 & 0 & 1 \\ ? & 0 & 1 & 0 & -1 & 2 \\ \hline & 0 & -2 & -1 & 0 & \\ & & \uparrow & \uparrow & \uparrow & \end{array} \tag{2}$$

We look again for a candidate to enter the basis. First, we try s_1, which has the biggest reduced cost coefficient, but we find no positive entry in this pivot column and thus no pivot element. We mark this column as "examined". The same happens if we choose the third column corresponding to x_3. Finally we select the second column as pivot column, but here we find the smallest column ratio in an already occupied row, namely the first one. At this point all non-basic columns are marked as "examined". The algorithm has not found any non-basic variable which could enter the basis and therefore terminates and states that the problem is infeasible.[2]

However, this problem is feasible, as

$$x_1 = 0\,,\ x_2 = 2\,,\ x_3 = 1\,,\ s_1 = 0$$

is a feasible basic solution.

The algorithm terminates with a wrong result! What is the reason for its failure?

Let us have a look at the basic solutions of this problem:

[2] In the proof of the finiteness and correctness of the algorithm Arsham mentions that one may have to try all possibilities of performing Gauss–Jordan row operations, but he gives no explicit instructions how and when to do it; furthermore there is no hint to this necessity in the description of the algorithm itself. Particularly, Step 4 of the algorithm declares the problem to be infeasible if there is no other nonbasic variable which is a candidate to enter the basic variable set. Notice that testing all possible pivot sequences could require an exponential amount of work before a problem can actually be declared infeasible.

Basis	x_1	x_2	x_3	s_1	
$\{x_1, x_2\}$	-1	2	0	0	not feasible
$\{x_1, x_3\}$	–	0	–	0	no solution
$\{x_1, s_1\}$	1	0	0	-2	not feasible
$\{x_2, x_3\}$	0	2	1	0	feasible
$\{x_2, s_1\}$	0	1	0	-1	not feasible
$\{x_3, s_1\}$	0	0	-1	-2	not feasible

As we see there is only one feasible basis, which does not contain x_1. But the algorithm chooses exactly x_1 as the first basic variable. Then it tries to occupy the other rows, i. e. to complete the basis by other variables without ever giving the variable x_1 a chance of leaving the basis again. This causes the erroneous behavior of the algorithm. Allowing pivot steps that exchange a basic variable against a non-basic variable would remedy this problem, but destroy the polynomial time bound — notice that the only pivot step in Tableau (2), the one that exchanges x_1 and x_2, would even lead to cycling in our example.

So the quest for a polynomial time pivot row is not yet finished!

References

[1] I. Adler and N. Meggido. A simplex algorithm whose average number of steps is bounded between two quadratic functions of the smaller dimension. *Journal of the Association for Computing Machinery*, 32:871–895, 1985.

[2] I. Adler, N. Meggido, and M. J. Todd. New results on the average behavior of simplex algorithms. *Bulletin of the American Mathematical Society*, 11:378–382, 1984.

[3] H. Arsham. An artificial-free simplex-type algorithm for general LP models. *Mathematical and Computer Modelling*, 25(1):107–123, 1997.

[4] H. Arsham. Initialization of the simplex algorithm: An artificial-free approach. *SIAM Review*, 39(4):736–744, December 1997.

[5] Karl H. Borgwardt. *The Simplex Method — A Probabilistic Analysis.* Springer-Verlag, Heidelberg, 1987.

[6] George B. Dantzig. *Linear Programming and Extensions.* Princeton University Press, 1974.

[7] L. G. Hačijan. A polynomial algorithm in linear programming. *Soviet Mathematics Doklady*, 20(1):191–194, 1979.

[8] Petra Huhn. *Schranken für die durchschnittliche Laufzeit des Simplexverfahrens und von Innere-Punkte-Verfahren.* PhD thesis, Universität Augsburg, 1997. Augsburger Mathematisch-naturwissenschaftliche Schriften, Bd. 16.

[9] N. K. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 26:373–395, 1984.

[10] Victor Klee and George J. Minty. How good is the simplex algorithm? In *[11]*, pages 159–175, 1972.

[11] Oved Shisha, editor. *Inequalities-III*, New York, 1972. Academic Press.