Some experimental results of station cone algorithm in comparison with simplex algorithm for linear programming

C N Nguyen¹ and H T Le²

¹ Hanoi Institute of Mathematics, 18 Hoang Quoc Viet, Hanoi, Vietnam
² Faculty of Informatic Technology, University of Mining and Geology, Hanoi, Vietnam.

Email ¹ dr.chu.vga@gmail.com; ² lethanhue.humg@gmail.com

Abstract. In this paper we introduce a new variant of station cone algorithm to solve linear programming problems. It uses a series of interior points Ok to determine the entering variables. The number of these interior points is finite and they move toward the optimal point. At each step, the calculation of new vertex is a simplex pivot. The proposed algorithm will be a polynomial time algorithm if the number of points Ok is limited by a polynomial function. The second objective of this paper is to carry out experimental calculations and compare with simplex methods and dual simplex method. The results show that the number of pivots of the station cone algorithm is less than 30 to 50 times that of the dual algorithm. And with the number of variables n and the number of constraints m increasing, the number of pivots of the dual algorithm is growing much faster than the number of pivots of the station cone algorithm. This conclusion is drawn from the computational experiments with n ≤ 500 and m ≤ 2000. In particular we also test for cases where n = 2, m = 100 000 and n = 3, m = 200 000. For case where n = 2 and m = 100 000, station cone algorithm is given no more than 16 pivots. In case of n = 3, m = 200 000, station cone algorithm has a pivot number less than 24.

1. Introduction

Linear programming (LP) is considered as one of the greatest inventions of mathematics in the 20th century. And there are two mathematicians who are regarded as the founders of the LP: Soviet mathematician Leonid Kantorovich (19 January 1912 – 7 April 1986) and American mathematician George Dantzig (November 8, 1914 – May 13, 2005).

In 1939, for the first time, Leonid Kantorovich studied the problem of planning production. And he came up with a mathematical model approach. He set up the mathematical model for the production planning problem along with the solution. The Kantorovich work - “Mathematical methods of organizing and planning production” [17] is recorded as the original appearance of linear programming.

But the important milestone of linear programming as a new field of mathematics was in 1947, when George Dantzig introduced the simplex algorithm. After its discovery by Dantzig in 1947 [6] the simplex method was unrivaled, until the late 1980s, for its utility in solving practical linear programming problems. The computational experiments show that the simplex method is efficient in practice [2,3,6,7]. Nevertheless, there exists a class of linear programming problems for which the simplex method takes an exponential number of steps [10].

In 1979 [9] Khachiyan introduced the ellipsoid method which run in polynomial time (a bound of $O(n^3L)$ arithmetic operations on number with $O(nL)$ digits). Khachiyan's algorithm was of landmark importance for establishing the polynomial time solvability of linear programs. Despite its major theoretical advance, the ellipsoid method had little practical impact as the simplex method is more efficient for many classes of linear programming problems [1,14].
In 1984 [8] Kamarkar proposed a new projective method for linear programming problems which not only improved Khachiyan’s theoretical worst-case polynomial bound but in fact promised dramatically practical performance improvement over simplex method. Karmarkar’s algorithm falls within the class of interior point methods. In contrast to the simplex method, which finds the optimal solution among the vertices of the feasible set, the interior point method moves through the interior of the feasible region and reaches the optimal solution only asymptotically. Stimulated by Karmarkar’s algorithm a variety of interior point methods were developed for linear programming [12,16].

There are several important open problems in the theory of linear programming, the solution of which would represent fundamental breakthrough in mathematics. In the recent survey on linear programming [15] M.J. Todd has mentioned unsolved problems: Is there a polynomial pivot rule for the simplex method? The immense efficiency of the simplex method in practice, despite its exponential time theoretical performance, hints that there may be variations of simplex algorithm that run in polynomial time.

Therefore, we set ourselves the following 3 purposes: The first purpose is to search the new algorithm more efficiently than the simplex algorithm; The second purpose is to find the polynomial pivot rule for the variation of simplex algorithm; The third purpose is to conduct experimental calculations to compare the newly found algorithm with the simplex algorithm.

In this paper, we present an algorithm, which can be considered a variant of the dual simplex method. In the next section, we introduce the station cone concept, which plays a key role in our algorithm. How to select the leaving variable is presented in the section 3. In section 4, we show how to choose the entering variable - this is an important key to the efficiency of the algorithm. Section 5 devoted to algorithm description. The result of experimental calculation is presented in section 6. A few comments are given in section 7.

2. Station Cone
Consider a linear programming problem in the matrix form
$$\text{Max } \{c,x\} \quad x \in P:=\{x|Ax \leq b, x \geq 0\},$$
(2.1)
where $c \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m, \forall x \in \mathbb{R}^n$. Let $A_i, A_2, ..., A_n$ denote the row vectors. Through this paper we suppose that (2.1) and its dual problem are nondegenerated. We also suggest the feasible region $P$ of (2.1) has strict interior points. For simplicity of argument, we assume that the matrix $A$ has full column rank $n$ and $n < m$.

Let $I_n = [i_1, i_2, ..., i_n] \subset \{1, 2, ..., m\}$ such that the vectors $A_{i}, i \in I_n$ are linear independent. This means the vector $A_i$, $i \in I_n$ establish a basis of $\mathbb{R}^n$. Therefore any vector $A_j \in \mathbb{R}^n$ can be expressed as a linear combination of the vectors $A_i$, $i \in I_n$. Let $\lambda_{i_j}$ be the linear coefficient of the vector $A_j$ in the basis $A_j$, $i_k \in I_n$, then
$$a_{j} = \sum_{i=1}^{n} \lambda_{i_j} a_{i_j}, \quad j = 1,2,...,n, \quad l = 1,2,...,m.$$ 
Consider the system of homogeneous linear inequalities
$$A_{i_k} x \leq 0, \quad i_k \in I_n.$$ 
(2.2)

**Definition 1.** The linear inequality
$$A_{i_k} x \leq 0$$
(2.3)
is called the consequent linear inequality of the system (2.2) if and only if all the solutions of the system (2.2) satisfy the linear inequality (2.3).

We need the following well known result in theory of linear inequalities.

**Theorem 2.1.** The linear inequality (2.3) is a consequent linear inequality of the system (2.2) if and only if
$$A_{i_k} = \sum_{i=1}^{n} \lambda_{i_{i_k}} A_{i_k}, \quad \lambda_{i_{i_k}} \geq 0, \quad i_k \in I_n$$
(2.4)
Definition of station cone. Let polyhedral cone $M$ be defined by system of linear inequalities

\[ A_1 x \leq b_1, \]
\[ A_2 x \leq b_2, \]
\[ \ldots \]
\[ A_n x \leq b_n, \]

where $A_1, A_2, \ldots, A_n$ are linear independent. Then $M$ is called a station cone if the vector $c$ is a nonnegative linear combination of the vectors $A_1, A_2, \ldots, A_n$. Then the vertex $x$ of the station cone $M$ is called a station solution and the vectors $A_1, A_2, \ldots, A_n$ is called a basis of a station cone.

Therefore, geometrically it can be seen that all the station cones lie on one side of the objective function $(c, x)$ at their vertices (see fig 1: $M^1, M^2, M^3, M^4, M^5$ are station cones and $M^6, M^7, M^8, M^9$ are not station cones). In other words, the solutions of the system of linear inequalities that create the station cones satisfy the inequality $\langle c, x \rangle \leq \langle c, x^* \rangle$, whereas $x^*$ is the vertex of the station cones. This is equal to the fact that the inequality $\langle c, x \rangle \leq \langle c, x^* \rangle$ is the consequent inequality of the system of the linear inequalities, which formulate the station cone. This also means that the vector $c$ is the nonnegative linear combination of the basic vectors of the station cone.

\[ \text{Figure 1. Station and non station cones} \]

**Theorem 2.2.** If the station solution $x$ satisfies all the constraints of the problem (2.1) then $x$ is an optimal solution.

3. Leaving variable

Let $A_1, A_2, \ldots, A_n$ be the basis of the station cone and

\[ c = \sum_{k=1}^{n} \lambda_{k0} A_k, \]
\[ A_j = \sum_{k=1}^{n} \lambda_{kj} A_k, \quad j = 1, 2, \ldots, m. \]

Then from definition 2.1 follows that: $\lambda_{k0} \geq 0, \quad \forall k = 1, 2, \ldots, n$.

From now on we assume that all $\lambda_{k0}$ are strictly positive, i.e. $\lambda_{k0} > 0, \quad k = 1, 2, \ldots, n$. 

3
It is obvious that \( \lambda_{k0} > 0, \ k = 1, 2, \ldots, n; \ \lambda_{k0} = 0, \ k = n+1, \ldots, m \) is a basis solution of the dual problem of (2.1):

\[
\min \ \langle b, \lambda \rangle \\
A^t \lambda \geq c^t \\
\lambda \geq 0,
\]

(3.1)

where \( \lambda \in R^m \). The assumption \( \lambda_{k0} > 0, \ k = 1, 2, \ldots, n \) means that the dual problem (3.1) is nondegenerated.

**Remark 1.** The vertex of the station cone is a basic solution of the dual problem.

4. **Pendulum principle and entering variable**

We find that, if we connect the vertices of the cones to the center of a circle, the vertices will oscillate around the optimal point according to the pendulum principle. Then finally stop at the optimal point. That is one of the main ideas of the station cone algorithm. In other words, the pendulum principle is one of the spinal ideas, from which the station cone algorithm is formed.

Let us approximate the equator of the earth by a polygon with the edge of 1 meter long. Then this polygon has 40 millions edges and 40 millions vertices. Suppose we have to find the maximum of a linear function \( cx_1 + cx_2 \) over this polygon.

On figure 1, let \( A \) denote an optimal point, \( B^1 \) denote the starting point. Suppose the distance between \( B^1 \) and \( A \) is 5 million meters. Then the simplex method will produce an optimal solution after 5 million iterations.

Let \( M^1 \) be a station cone defined by 2 constraints containing points \( B^1 \) and \( D^1 \), where \( D^1 \) is on the other side of \( A \) with a distance, for examples, 4 million meters to \( A \) (see figure 2).

![Figure 2. Pendulum principle](image)

We denote by \( x^1 \) the vertex of \( M^1 \). Since \( M^1 \) is a station cone, it is clear that \( cx_1 \geq cx, \forall x \in M^1 \). The station cone \( M^1 \) will be our starting cone. Starting our algorithm with the operation of connecting \( x^1 \) with \( O \), where \( O \) is the center of the equator. The segment \([x^1, 0] \) will intersect with the boundary of \( P \) at \( B^2 \). Replacing the constraint containing \( B^1 \) by the constraint containing \( B^2 \) we have a new cone \( M^2 \). Repeat the above procedure with \( M^2 \) and we have \( M^3 \), etc. (see figure 1). The replacement of one constraint by another has to follow the restriction that the new generating cone is a station cone. We note that at each iteration, the distance between two points \( B^k \) and \( D^k \) defined by two edges of the station cone \( M^k \) is reduced by approximately 2 times in comparison with the previous iteration. Therefore the number of the iterations \( T \) can be estimated by the following bound

\[
T \approx \log_2 \frac{m}{2}
\]

(4.1)
For our example with \( m = 40 \) million the formula (4.1) gives
\[
T \approx \log_2 \frac{m}{2} = \log_2 2.10^{25} < 25.
\]

The above example shows that our algorithm can produce an optimal solution after around 25 iterations.

**Initial station cone**

We now proceed to find an initial station cone. We can find an initial station cone \( M \) by solving the following system
\[
\begin{align*}
A^T \lambda &= c^T, \\
\lambda &\geq 0,
\end{align*}
\]
(4.2)

where \( \lambda \in \mathbb{R}^m \). We can suppose \( c^T \geq 0 \) because, if some coefficient of \( c^T \) is negative then we multiply both sides of the corresponding equation with \(-1\). To find a solution of (4.2), we solve the following big - \( M \) problem
\[
\begin{align*}
\min \{ M_1 y_1 + M_2 y_2 + \ldots + M_n y_n \} \\
A^T \lambda + E y &= c^T, \\
\lambda &\geq 0, \quad y \geq 0,
\end{align*}
\]
(4.3)

Where, \( \lambda \in \mathbb{R}^m \), \( y \in \mathbb{R}^n \) and \( E \) is the unit matrix of \((n \times n)\) and \( M_1, M_2, \ldots, M_n \) are significantly large positive numbers. The problem (4.3) has an optimal solution \( \lambda^* \geq 0 \), \( y^* = 0 \). and \( \lambda^* \) is a solution of (4.2).

We also assume that a strict interior feasible solution \( O \) of (2.1) is available. If such an initial point is not available then we modify the problem using the usual big - \( M \) augmentation [11] as follows:
\[
\begin{align*}
\max \{ \langle c, x \rangle - M x_{n+1} \} \\
A x - e x_{n+1} &\leq b, \\
x, x_{n+1} &\geq 0,
\end{align*}
\]
(4.4)

Where \( e = (1, 1, \ldots, 1)^T \in \mathbb{R}^m \) and \( M \) is a significantly large positive number.

Let \( x_{n+1}^0 = \max \{0, -b_1, -b_2, \ldots, -b_n\} \). Then \( (0, \ldots, 0, x_{n+1}^0)^T \) is a strict interior feasible solution of (4.4) which is in the same form as (2.1).

**Initial interior point**

Let \( O \) be a strict interior point of \( P \). Denoted by \( O', i = 1, 2, \ldots, n \) the projections of \( O \) onto \( n \) facets of the station cone \( M^k \). Let \( H_i, i = 1, 2, \ldots, n \) be the intersection points of the boundary of \( P \) and the segments \( O, O', i = 1, 2, \ldots, n \). Then the new point \( O^* \) will be calculated by the following formula
\[
O^* = \frac{1}{n+1} \left( \sum_{i=1}^{n} H_i + O \right)
\]
(4.5)

5. **Station Cone Algorithm**

1. **Initialization**

Determine the starting station cone \( M \). Calculate the point \( O^* \) by formula (4.5).

Let \( M^k = M; O = O^* \).

2. **Step** \((k = 1, 2, \ldots)\)

If the vertex \( x_k \) of the station cone \( M^k \) is a feasible point of \( P \), then \( x_k \) is an optimal solution. In the contrary case, select the inequality \( A_j x \leq b_j \) for entering the station cone and define the inequality \( A_j x \leq b_j \) for leaving the station cone. Determine the new station cone \( M^{[k+1]} \) with the vertex \( x^{[k+1]} \). Go to next step \( k = k + 1 \).

**Remark 2.** Except for the calculation for finding the entering variable, each step of algorithm 1 is a simplex pivot.

With the assumption that the dual problem (3.1) of (2.1) is nondegenerated, we hence have the following
**Theorem 2.6.**
The above algorithm produces an optimal solution after a finite number of iterations.
*Proof. Follows from the theorems 2.3,2.4,2.5.*

6. **Computational experiences**
The above proposed station cone algorithm has been tested, using MatLab, on a set of randomly generated linear problems [13] of the form
\[
\begin{align*}
\max \{c^T x\} \\
A x \leq b,
\end{align*}
\]  
(6.1)
where \(c=(1,1,...,1)\in R^n\), \(A\) is the full matrix of \((n \times m)\) with \(a_j\) is randomly generated from the interval \([0,1]\), the vector \(b\) has been chosen such that the hyperplanes \(A_i x = b_i\), \(i=1,...,m\) are tangent to the sphere \((0,1)\) with center at origin and radius \(r=1\). To ensure that (6.1) has a finite optimal solution we add the constraints
\[
x_i \leq 1, \ i=1,2,...,n.
\]  
(6.2)

The optimal solution and objective function value of ((6.1)-(6.2)) have been retested by simplex algorithm from MatLab.

Function **Data01. m** randomly generates the input data for the problems and stores the matrix \(A\) and, vector \(b\) in the data base form **Dat01. mat**. Function **Alg01. m** solves the problem by a new proposed algorithm and function **Simplex01. m** itself is the simplex algorithm from the optimization toolbox of MatLab.

Test results are shown in the tables below (SCA: Station Cone Algorithm).

| \(n\) | \(m\) | Problem | Pivots | Ratio \((\text{SIMPLEX}/\text{SCA})\) |
|------|------|---------|--------|-----------------|
|      |      |         | SIMPLEX | SCA              |
| 2    | 500  | 1       | 257     | 9               | 28.5     |
|      | 1000 | 1       | 518     | 8               | 64.8     |
|      | 2000 | 1       | 1000    | 10              | 100      |
|      | 3000 | 1       | 1540    | 11              | 140      |
|      | 5000 | 1       | 2505    | 13              | 192.6    |
|      | 10000| 1       | 4955    | 14              | 353.9    |
|      | 20000| 1       | 9967    | 14              | 711.9    |
|      | 50000| 1       | 25043   | 15              | 1669.5   |
|      | 100000| 1       | 50314   | 16              | 3144.6   |
| 3    | 500  | 1       | 44      | 12              | 3.6      |
|      | 1000 | 1       | 60      | 15              | 4        |
|      | 2000 | 1       | 98      | 13              | 7.5      |
|      | 3000 | 1       | 104     | 18              | 5.7      |
|      | 5000 | 1       | 149     | 18              | 8.2      |
|      | 10000| 1       | 174     | 18              | 9.7      |
|      | 20000| 1       | 284     | 17              | 16.7     |
|      | 50000| 1       | 423     | 21              | 20.1     |
|      | 100000| 1       | 626     | 22              | 28.5     |
|      | 150000| 1       | 779     | 18              | 43.2     |
|      | 200000| 1       | 912     | 23              | 39.7     |
Table 2. $150 \leq n \leq 300$, $200 \leq m \leq 700$

| $n$ | $m$ | Problem | Pivots | Ratio (SIMPLEX/SCA) |
|-----|-----|---------|--------|---------------------|
| 150 | 200 | 1       | 13282  | 1710               |
|     |     | 2       | 13714  | 1950               |
|     |     | 3       | 12672  | 1720               |
|     |     | Average | 11720  | 1424               | 8.230 |
| 150 | 250 | 1       | 12834  | 1710               |
|     |     | 2       | 13714  | 1950               |
|     |     | 3       | 12672  | 1720               |
|     |     | Average | 13073  | 1793               | 7.291 |
| 200 | 300 | 1       | 26367  | 2628               |
|     |     | 2       | 24800  | 2941               |
|     |     | 3       | 27010  | 2813               |
|     |     | Average | 26059  | 2794               | 9.326 |
| 250 | 300 | 1       | 35942  | 3387               |
|     |     | 2       | 36978  | 3434               |
|     |     | 3       | 40686  | 3473               |
|     |     | Average | 37869  | 3473               | 11.047 |
| 250 | 500 | 1       | 66942  | 5751               |
|     |     | 2       | 65302  | 5608               |
|     |     | 3       | 68747  | 5422               |
|     |     | Average | 66003  | 5593               | 11.801 |
| 300 | 600 | 1       | 108448 | 7964               |
|     |     | 2       | 115799 | 11007              |
|     |     | Average | 115047 | 101047             | 11.801 |

Table 3. $n = 300, 400, 500$; $m = 1000$, 2000

| $n$ | $m$ | Problem | Pivots | Ratio (DUAL SIMPLEX/SCA) |
|-----|-----|---------|--------|--------------------------|
| 300 | 1000| 1       | 227 215 | 8952               | 26.44 |
| 400 | 1000| 1       | 388 676 | 13 266             | 29.29 |
| 500 | 1000| 1       | 583 464 | 21 033             | 27.74 |
| 100 | 2000| 1       | 997 853 | 21 807             | 45.75 |

7. Conclusions

7.1. The above tested examples show that the number of pivots of the station cone algorithm is significantly smaller than the simplex and dual methods.

7.2. The test has confirmed the trend that as the number of variables and constraints increases, the number of pivots of the simplex algorithm increases more rapidly than the number of pivots of the station cone algorithm. Therefore, it is necessary to carry out calculations with larger examples.

References

[1] M.L. Balinski Mathematical programming: journal, society, recollections History of Mathematical Programming: a Collection of Personal Reminiscences, ed J.-K. Lenstra, A. H. G. Rinnooy Kan et A. Schrijver (CWI et North-Holland Publishing Company, Amsterdam) pp. 5-18

[2] R.E. Bixby 1992 Implementing the simplex method: The initial basis ORSA Journal on Computing 4 pp 267-284
[3] R.E. Bixby 1994 Progress in linear programming ORSA Journal on Computing 6(1) pp 15-22
[4] S.N. Chernikov 1968 Linear Inequalities (Nauka, Moskva)
[5] G.B. Dantzig 1951 Maximization of a linear function of variables subject to linear inequalities Activity Analysis of Production and Allocation, ed Koopmans, T.C. (Wiley, New York) pp. 339-347
[6] G.B. Dantzig 1963 Progress in linear programming ORSA Journal on Computing 6(1) pp 15-22
[7] J.J. Forrest, D. Goldfarb 1992 Steepest-edge simplex algorithms for linear programming Mathematical Programming 57 pp 341-374
[8] N.K. Karmarkar 1984 A new polynomial-time algorithm for linear programming Combinatorica 4 pp 373-395
[9] L.G. Khachiyan 1979 A polynomial algorithm in linear programming (in Russian), Doklady Akademiia Nauk SSSR 224 pp 1093-1096. English translation: Soviet Mathematics Doklady 20 pp 191-194
[10] V. Klee, G. J. Minty 1972 How good is the simplex algorithm? Inequalities III ed Shisha O. (Academic Press) pp 159-175
[11] K. G. Murty 2005 A Gravitational Interior Point Method for LP Opsearch 42(1) pp 28–36
[12] K. G. Murty 2006 A New Practically Efficient Interior Point Method for LP Algorithmic Operations Research 1 pp 3–19
[13] K. Paparrizos *, N. Samaras, G. Stephanides 2003 An efficient simplex type algorithm for sparse and dense linear programs European Journal of Operational Research 148 pp 323–334
[14] S. Smale 1983 On the average number of the simplex method of linear programming Mathematical Programming 27 pp 241-262
[15] M.J. Todd 2002 The many facets of linear programming Mathematical Programming 91 pp 417-436
[16] D.G. Luenberger, Y. Ye 2008 Linear and Nonlinear Programming (Springer Berlin)
[17] L.V. Kantorovich 1939 Mathematical methods of organizing and planning production