PERGAMON

Computers & Operations Research 29 (2002) 195–203

computers & operations research

www.elsevier.com/locate/dsw

An efficient search direction for linear programming problems

Hsing Luh[a,*], Ray Tsaih[b]

[a]Department of Mathematical Sciences, National Chengchi University, District Wen-Shan, Taipei 116, Taiwan, ROC
[b]Department of MIS, National Chengchi University, District Wen-Shan, Taipei 116, Taiwan, ROC

Received 1 January 1999; received in revised form 1 March 2000

Abstract

In this paper, we present an auxiliary algorithm, in terms of the speed of obtaining the optimal solution, that is effective in helping the simplex method for commencing a better initial basic feasible solution. The idea of choosing a direction towards an optimal point presented in this paper is new and easily implemented. From our experiments, the algorithm will release a corner point of the feasible region within few iterative steps, independent of the starting point. The computational results show that after the auxiliary algorithm is adopted as phase I process, the simplex method consistently reduce the number of required iterations by about 40%.

Scope and purpose

Recent progress in the implementations of simplex and interior point methods as well as advances in computer hardware has extended the capability of linear programming with today's computing technology. It is well known that the solution times for the interior point method improve with problem size. But, experimental evidence suggests that interior point methods dominate simplex-based methods only in the solution of very large scale linear programs. If the problem size is medium, how to combine the best features of these two methods to produce an effective algorithm for solving linear programming problems is still an interesting problem. In this research we present a new effective ε-optimality search direction based on the interior point method to start an initial basic feasible solution near the optimal point for the simplex method.

Keywords: Linear programming; Interior-point-based solution

* Corresponding author. Tel.: + 886-2-2938-7373; fax: + 886-2-2939-0005.
E-mail address: paul@math.nccu.edu.tw (H. Luh).

PII: S0305-0548(00)00069-1

1. Introduction and literature review

Currently there are two popular approaches in linear programming (LP): simplex method and the interior-point algorithm originated from Karmarkar's approach. In addition, many of their variants developed both in theory and applications are still in progress. The simplex method obtains the optimal solution via moving consecutively to a better adjacent corner-point at the feasible region, and its modifications try to improve the speed of attaining the optimality. In contrast, the algorithm of Karmarkar [1] is claimed as an interior-point approach, which goes from a feasible point to a feasible point through the interior of the feasible region. Roughly speaking, the main difference among them is that the simplex method is devoted to reach the exact optimal solution, while Karmarkar-based method is quite fast in approaching to the neighborhood of the optimal solution, but it is slowed while approaching the optimal point.

Each iteration of Karmarkar's algorithm is a steepest descent search with scaling and projection. This brought linear programming algorithm into the realm of interior point methods, using nonlinear programming techniques. Further other kinds of interior point algorithms have been developed subsequent to Karmarkar's algorithm. Examples of those algorithms such as affine scaling, path following and affine potential reduction methods can be found in Fang and Puthenpura [2].

On the computational experiments, many implementations have used simple affine scaling-type algorithms; see for example, Monma and Morton [3] and Adler et al. [4]. Such methods are particularly simple to implement and in practical work well. Anstreicher and Watteyne [5] considered a family of search directions for the standard form variant of the projective algorithm. This family includes the original projected gradient direction, and a direction first proposed in Todd [6]. Using different search directions from the family, their computational results demonstrate that a small number of monotonic steps on early iterations may considerably improve the performance of the algorithm.

The recent work of Anderson and Ye [7] indicated that combining interior point and pivoting algorithms are both theoretic guarantee and practical efficiency to an optimal basis. Their method achieves for a speed of convergence leading to a solution of the problem in $O(n^{0.5}L)$ iterations which preserves the algorithm's polynomial-time complexity. Other computational experiments which combined the simplex and interior point algorithm have been reported by Bixby et al. [8] to illustrate the power of such a method for solving large-scale linear programs.

In this research, we present an auxiliary algorithm that can release a better initial corner point of feasible solution for the simplex method. The speed of obtaining the optimal solution is a better viewpoint, starting from that feasible corner point. Empirically, the auxiliary algorithm acts as the phase I process of the simplex method. After phase I, the simplex method will start from a feasible corner point to search the optimal solution as conventionally called the phase II process.

During the phase I process, the auxiliary algorithm moves a point consecutively to a better solution on the other boundary of the polytope which is formed by the feasible region. The moving direction is a linear combination of the negative gradient direction of the objective function and a direction pointing towards the interior of the polytope. This movement, however, is taken only when the step size is large. If the step size becomes tiny, the point is moved to a better vertex on the same boundary. The process will be repeated till the point hits a corner point at the feasible region.

The paper is organized as follows. In Section 2, we first explain how an interior improving search direction is constructed for standard form linear programs and then discuss how this algorithm adapts to problems with the simplex method. Results of extensive computational tests comparing the use of the simplex method are reported in Section 3. Finally, the summary and future work are discussed in Section 4.

2. The proposed approach

Consider the following standard form of LP problem (1). In (1), $\mathbf{x}^t \equiv (x_1 \;\; x_2 \;\; \cdots \;\; x_n)$, $\tilde{\mathbf{A}}$ is an $m \times n$ matrix, $\tilde{\mathbf{b}}' \equiv (\tilde{b}_1 \;\; \tilde{b}_2 \;\; \cdots \;\; \tilde{b}_m)$, $\tilde{\mathbf{c}}' \equiv (\tilde{c}_1 \;\; \tilde{c}_2 \;\; \cdots \;\; \tilde{c}_n)$, and $\mathbf{0}$ is denoted a column vector, of appropriate dimension, with each component equals zero.

$$\begin{aligned}&\text{Minimize} \quad z = \tilde{\mathbf{c}}^t\mathbf{x}\\ &\text{Subject to} \quad \tilde{\mathbf{A}}\mathbf{x} \geqslant \tilde{\mathbf{b}}, \quad \mathbf{x} \geqslant 0.\end{aligned} \tag{1}$$

Let LP problem (2) be a normalization version of LP (1). That is, $\mathbf{c} \equiv \tilde{\mathbf{c}}/\|\tilde{\mathbf{c}}\|$, $\mathbf{A}_{i\cdot} \equiv \tilde{\mathbf{A}}_{i\cdot}/\|\tilde{\mathbf{A}}_{i\cdot}\|$, and $b_i \equiv \tilde{b}_i/\|\tilde{\mathbf{A}}_{i\cdot}$, where $\tilde{\mathbf{A}}_{i\cdot}$ is the ith row vector of $\tilde{\mathbf{A}}$ and the Euclidean norm is adopted for $\|\cdot\|$. It is easy to check the sets of feasible solutions as well as the optimal solutions of both LP (1) and LP (2) are equivalent.

$$\begin{aligned}&\text{Minimize} \quad z = \mathbf{c}^t\mathbf{x}\\ &\text{Subject to} \quad \mathbf{A}\mathbf{x} \geqslant \mathbf{b}, \quad \mathbf{x} \geqslant \mathbf{0}.\end{aligned} \tag{2}$$

Below, we take LP (2) as our standard form of input to illustrate the idea of the improving feasible direction in details.

Definition. Given $\mathbf{P} \equiv \{\mathbf{x} \mid \mathbf{A}\mathbf{x} \geqslant \mathbf{b},\, \mathbf{x} \geqslant \mathbf{0}\}$, the feasible region, and $\mathbf{x} \in \mathbf{P}$, we say $\mathbf{x}$ is an interior point of $\mathbf{P}$, if there exists a scalar $\varepsilon > 0$ such that the open ball $\mathbf{B} = \{\mathbf{y} \mid \|\mathbf{x} - \mathbf{y}\| < \varepsilon\}$ is contained in $\mathbf{P}$. Otherwise $\mathbf{x}$ is a boundary point.

If $\mathbf{x}^{(k)}$ is a boundary point of $\mathbf{P}$, let $\Omega_1^k \equiv \{i \mid \mathbf{A}_{i\cdot}\mathbf{x}^{(k)} = b_i\}$ and $\Omega_2^k \equiv \{j \mid \mathbf{x}_j^{(k)} = 0\}$ be the index set corresponding to the binding condition of functional and non-negativity constraints at $\mathbf{x}^{(k)}$, respectively. Thus, the binding constraints corresponding to $\mathbf{P}$ at $\mathbf{x}^{(k)}$ are $\mathbf{A}_{i\cdot}\mathbf{x} = b_i$, $\forall i \in \Omega_1^k$, and $x_j = 0$, $\forall j \in \Omega_2^k$.

Let the polytope $\mathbf{P}^{(k)} \equiv \{\mathbf{x} \mid \mathbf{c}^t\mathbf{x} \leqslant \mathbf{c}^t\mathbf{x}^{(k)}\} \cap \mathbf{P}$. Thus, $\mathbf{x}^{(k)}$ is a boundary point of $\mathbf{P}^{(k)}$. At point $\mathbf{x}^{(k)}$, the binding constraints corresponding to $\mathbf{P}^{(k)}$ are $\mathbf{c}^t\mathbf{x} = \mathbf{c}^t\mathbf{x}^{(k)}$, $\mathbf{A}_{i\cdot}\mathbf{x} = b_i$, $\forall i \in \Omega_1^k$ and $x_j = 0$, $\forall j \in \Omega_2^k$.

Lemma 1. If $\mathbf{x}^{(k)}$ is a boundary point of $\mathbf{P}$, then $\sum_{i \in \Omega_1^k} \mathbf{A}_{i\cdot}^t + \sum_{j \in \Omega_2^k} \mathbf{e}_j$ will not be a zero vector, where $\mathbf{e}_j$ is a n-dimensional unit vector with 1 for its jth component and 0 for the rest.

Proof. Let $\Omega_1^k = \{i \mid \mathbf{A}_{i\cdot}\mathbf{x}^{(k)} = b_i\}$ and $\Omega_2^k = \{j \mid x_j^{(k)} = 0\}$ and $\mathbf{Q} = \{\mathbf{x} \mid \mathbf{A}_{i\cdot}\mathbf{x} = b_i, \forall i \in \Omega_1^k; \mathbf{A}_{i\cdot}\mathbf{x} \geqslant b_i,\, \forall i \notin \Omega_1^k;\, x_j = 0, \forall j \in \Omega_2^k;\, x_j \geqslant 0,\, \forall j \notin \Omega_2^k\}$. Obviously, we have $\mathbf{Q} \subseteq \mathbf{P}$. Although Ω_1^k and Ω_2^k be index sets defined by $\mathbf{x}^{(k)}$ as it is a point at the binding constraints of $\mathbf{P}$, $\mathbf{x}^{(k)}$ does not correspond to the binding constraints of $\mathbf{Q}$. Now, assume

$$\sum_{i\in\Omega_1^k} \mathbf{A}_{i\cdot}^t + \sum_{j\in\Omega_2^k} \mathbf{e}_j = \mathbf{0}. \tag{3}$$

Then, $\Omega_1^k \cup \Omega_2^k$ is nonempty for it has at least two elements, and $\sum_{i\in\Omega_i^k} b_i = 0$ since

$$\mathbf{A}_{i\cdot}\mathbf{x}^{(k)} = b_i \quad \forall i\in\Omega_1^k,$$
$$\mathbf{e}_j^t\mathbf{x}^{(k)} = 0 \quad \forall j\in\Omega_2^k.$$

Therefore, these linear equations, $\mathbf{A}_{i\cdot}\mathbf{x} = b_i$, $\forall i\in\Omega_1^k$ and $\mathbf{e}_j^t\mathbf{x} = 0$, $\forall j\in\Omega_2^k$, are linearly dependent. Under condition (3), we have $\mathbf{P} \subseteq \mathbf{Q}$. This implies that $\mathbf{P}$ is equivalent to $\mathbf{Q}$, and results in that $\mathbf{x}^{(k)}$ is not at the binding constraints of $\mathbf{P}$. It contradicts to the assumption of $\mathbf{x}^{(k)}$, a boundary point of $\mathbf{P}$. □

From Lemma 1, when $\mathbf{x}^{(k)}$ is a boundary point of $\mathbf{P}$, there exists

$$\mathbf{h}^{(k)} \equiv \frac{\sum_{i\in\Omega_1^k} \mathbf{A}_{i\cdot}^t + \sum_{j\in\Omega_2^k} \mathbf{e}_j}{\|\sum_{i\in\Omega_1^k} \mathbf{A}_{i\cdot}^t + \sum_{j\in\Omega_2^k} \mathbf{e}_j\|} \tag{4}$$

and $\mathbf{h}^{(k)}$ is not a zero vector.

At the kth iteration with the current boundary point $\mathbf{x}^{(k)}$, the following improving direction $\mathbf{g}^{(k)}$ is adopted as

$$\mathbf{g}^{(k)} = \begin{cases} \mathbf{0} & \text{if } \mathbf{h}^{(k)} = \mathbf{c}, \\ \dfrac{\mathbf{h}^{(k)} - \mathbf{c}}{\|\mathbf{h}^{(k)} - \mathbf{c}\|} & \text{if } \mathbf{h}^{(k)} \neq \mathbf{c}. \end{cases} \tag{5}$$

The vector $\mathbf{g}^{(k)}$ is an unit vector, started from $\mathbf{x}^{(k)}$ and emitted towards the interior of $\mathbf{P}^{(k)}$. Thus, we have the following lemma.

Lemma 2. *If* $\mathbf{x}^{(k)}$ *is not an optimal solution of LP* (2), $\mathbf{g}^{(k)}$ *will not be a zero vector.*

Proof. By contraposition, we prove that if $\mathbf{g}^{(k)}$ is a zero vector, then $\mathbf{x}^{(k)}$ is not an optimal solution of LP (2). From (5), if $\mathbf{g}^{(k)} = \mathbf{0}$, then $\mathbf{h}^{(k)} = \mathbf{c}$. Let

$$\alpha = \left\| \sum_{i\in\Omega_1^k} \mathbf{A}_i^t + \sum_{j\in\Omega_2^k} \mathbf{e}_j \right\| > 0.$$

Then for any $\mathbf{x}\in\mathbf{P}$,

$$\mathbf{c}^t(\mathbf{x} - \mathbf{x}^{(k)}) = \frac{1}{\alpha}\left[\sum_{i\in\Omega_1^k} (\mathbf{A}_{i\cdot}\mathbf{x} - \mathbf{A}_{i\cdot}\mathbf{x}^{(k)}) + \sum_{j\in\Omega_2^k} (\mathbf{x}_j - \mathbf{x}_j^{(k)}) \right] \geqslant 0.$$

Hence $\mathbf{c}^t\mathbf{x}^{(k)} \leqslant \mathbf{c}^t\mathbf{x}$ for any $\mathbf{x}\in\mathbf{P}$. The fact $\mathbf{h}^{(k)} = \mathbf{c}$ implies that, regarding $\mathbf{P}$, $\{\mathbf{x}|\mathbf{A}_{i\cdot}\mathbf{x} - b_i, \forall i\in\Omega_1^k, x_j = 0, \forall j\in\Omega_2^k\}$ is on a supporting hyperplane passing $\mathbf{x}^{(k)}$. Thus $\mathbf{x}^{(k)}$ is an optimal solution of LP (2). □

The following lemma shows that the value of the objective function is decreased as $\mathbf{x}$ is being adjusted with $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \eta^{(k)}\mathbf{g}^{(k)}$, where

$$\eta^{(k)} = \min\left\{\frac{-x_j^{(k)}}{g_j^{(k)}},\ g_j^{(k)} < 0,\ 1 \leqslant j \leqslant n;\ \frac{b_i - \mathbf{A}_i.\mathbf{x}^{(k)}}{\mathbf{A}_i.\mathbf{g}^{(k)}},\ \mathbf{A}_i.\mathbf{g}^{(k)} < 0,\ 1 \leqslant i \leqslant m\right\}. \tag{6}$$

Lemma 3. *If* $\mathbf{g}^{(k)} \neq \mathbf{0}$ *and* $\eta^{(k)} > 0$, *then* $\mathbf{c}^t\mathbf{x}^{(k+1)} < \mathbf{c}^t\mathbf{x}^{(k)}$.

Proof. Since both $\mathbf{h}^{(k)}$ and $\mathbf{c}$ are unit vectors, we have $\mathbf{c}^t\mathbf{g}^{(k)} \leqslant 0$. If $\mathbf{g}^{(k)} \neq \mathbf{0}$, then $\mathbf{c}^t\mathbf{g}^{(k)} < 0$. Thus, if $\mathbf{g}^{(k)} \neq \mathbf{0}$ and $\eta^{(k)} > 0$, we have $\mathbf{c}^t\mathbf{x}^{(k+1)} = \mathbf{c}^t\mathbf{x}^{(k)} + \eta^{(k)}\mathbf{c}^t\mathbf{g}^{(k)} < \mathbf{c}^t\mathbf{x}^{(k)}$. □

Thus, we have $\mathbf{c}^t\mathbf{x}^{(k+1)} \leqslant \mathbf{c}^t\mathbf{x}^{(k)}$ for all k. If $\mathbf{x}^{(k+1)} \neq \mathbf{x}^{(k)}$ then $\mathbf{c}^t\mathbf{x}^{(k+1)} < \mathbf{c}^t\mathbf{x}^{(k)}$. Furthermore, it is easy to visualize $\mathbf{P}^{(k+1)} \subset \mathbf{P}^{(k)} \subset \mathbf{P}$, and if LP (2) is bounded, then $\mathbf{P}^{(k)}$ converges to the optimal set. From (6), $\mathbf{x}^{(k+1)}$ is also a boundary point of $\mathbf{P}$.

Theorem 1. $\mathbf{x}^{(k)}$ *is an optimal solution if and only if* $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)}$.

Proof. (I) We prove if $\mathbf{x}^{(k)}$ is an optimal solution, then $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)}$.
(a) If $\mathbf{c}^t\mathbf{g}^{(k)} = 0$, then $\mathbf{g}^{(k)} = 0$ because $\mathbf{g}^{(k)}$, $\mathbf{h}^{(k)}$ and $\mathbf{c}$ are unit vectors. Thus $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)}$.
(b) If $\mathbf{c}^t\mathbf{g}^{(k)} < 0$, then $\mathbf{g}^{(k)} \neq \mathbf{0}$.
(b1)If $\eta^{(k)} = 0$, then $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)}$.
(b2)If $\eta^{(k)} > 0$, then, $\mathbf{x}^{(k+1)}$ is a feasible solution. From Lemma 3, we have $\mathbf{c}^t\mathbf{x}^{(k+1)} < \mathbf{c}^t\mathbf{x}^{(k)}$.
This implies that $\mathbf{x}^{(k)}$ is not an optimal solution. It contradicts to our assumption.
Therefore, if $\mathbf{x}^{(k)}$ is an optimal solution, then $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)}$.
(II) We prove if $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)}$, then $\mathbf{x}^{(k)}$ is an optimal solution.
By contraposition, we prove that if $\mathbf{x}^{(k)}$ is not an optimal solution, then $\mathbf{x}^{(k+1)}$ does not equal $\mathbf{x}^{(k)}$. Lemma 2 has proved that if $\mathbf{x}^{(k)}$ is not an optimal solution, $\mathbf{g}^{(k)}$ will not be a zero vector. Thus there remains to prove that, if $\mathbf{x}^{(k)}$ is not an optimal solution, there exists an $\eta^{(k)} > 0$. From (5), we have $(\mathbf{h}^{(k)})^t\mathbf{g}^{(k)} > 0$. If $\mathbf{A}_i.\mathbf{g}^{(k)} \geqslant 0$, for all $i \in \Omega_1^k$ and $\mathbf{g}_j^{(k)} \geqslant 0$, for all $j \in \Omega_2^k$, then there exists an $\eta^{(k)} > 0$, such that $\mathbf{x}^{(k+1)}$ is a feasible solution and $\mathbf{x}^{(k+1)}$ does not equal $\mathbf{x}^{(k)}$. Otherwise, let $\mathbf{M}$ be a projection matrix defined on the null space of equations in Ω_1^k and Ω_2^k. Define $\mathbf{M} \equiv -[\mathbf{I} - \mathbf{A}_q^t(\mathbf{A}_q\mathbf{A}_q^t)^{-1}\mathbf{A}_q]$, where $\mathbf{A}_q$ is defined to be composed of the row vectors of $\mathbf{A}_i.\ \forall i \in \Omega_1^k$ and $\mathbf{e}_j^t\ \forall j \in \Omega_2^k$. Thus, it will make $\mathbf{M}\mathbf{g}^{(k)}$ a feasible direction with $\eta^{(k)} > 0$ and $\mathbf{c}^t\mathbf{M}\mathbf{g}^{(k)} < 0$ such that $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \eta^{(k)}\mathbf{M}\mathbf{g}^{(k)}$ is a feasible solution and $\mathbf{x}^{(k+1)} \neq \mathbf{x}^{(k)}$. □

Table 1 displays the proposed algorithm. From Lemma 2, the criterion of $\mathbf{h}^{(k)} = \mathbf{c}$ in Step 3 is a sufficient stopping criterion of the proposed algorithm. Step 6 guarantees $\mathbf{x}^{(k+1)}$ is always a boundary point of $\mathbf{P}$. Theorem 1 says that the criterion in Step 7 is also a sufficient stopping criterion of the proposed algorithm. As to find an initial feasible solution of LP (2) in Step 0, we can apply the same algorithm to LP (2) with artificial variables.

In order to avoid the zig-zag situation during the process, the loop of Steps 11–16 is set for determining a basic feasible solution to start the simplex method. Generally, Steps 11–16

Table 1
The proposed algorithm. $\mathbf{e}_j$ is a n-dimensional unit vector with 1 for its jth component and 0 for the rest, and $\mathbf{A}_{i\cdot}$ is the ith row of $\mathbf{A}$. $\varepsilon < 10^{-6}$, and L is a large number

Step 0.	Start with a feasible solution of LP (2), $\mathbf{x}^{(0)}$. Let $k = 0$.
Step 1.	Let $\Omega_1^k = \{i \vert \mathbf{A}_{i\cdot}\mathbf{x}^{(k)} = b_i\}$ and $\Omega_2^k = \{j \vert x_j^{(k)} = 0\}$.
Step 2.	Calculate $\mathbf{h}^{(k)} = \dfrac{\sum_{i\in\Omega_1^k}\mathbf{A}_{i\cdot}^t + \sum_{j\in\Omega_2^k}\mathbf{e}_j}{\Vert\sum_{i\in\Omega_1^k}\mathbf{A}_{i\cdot}^t + \sum_{j\in\Omega_2^k}\mathbf{e}_j\Vert}$.
Step 3.	If $\mathbf{h}^{(k)} = \mathbf{c}$, then STOP and claim $\mathbf{x}^{(k)}$ is an optimal solution.
Step 4.	Calculate $\mathbf{g}^{(k)} = \dfrac{\mathbf{h}^{(k)} - \mathbf{c}}{\Vert\mathbf{h}^{(k)} - \mathbf{c}\Vert}$.
Step 5.	If $\forall i \in \Omega_1^k$ $\mathbf{A}_{i\cdot}\mathbf{g}^{(k)} > 0$ and $\forall j \in \Omega_2^k \mathbf{g}_j^{(k)} > 0$, then $\mathbf{v} = \mathbf{g}^{(k)}$; Otherwise, let $\mathbf{v} = -[\mathbf{I} - \mathbf{A}_q^t(\mathbf{A}_q\mathbf{A}_q^t)^{-1}\mathbf{A}_q]\mathbf{g}^{(k)}$, $\mathbf{A}_q$ is composed of the row vectors of $\mathbf{A}_{i\cdot}\,\forall i \in \Omega_1^k$ and $\mathbf{e}_j^t$ $\forall j \in \Omega_2^k$.
Step 6.	Let $\eta^{(k)} = \min\{\dfrac{-x_j^{(k)}}{v_j}, v_j < 0, j \notin \Omega_2^k; \dfrac{b_i - \mathbf{A}_{i\cdot}\mathbf{x}^{(k)}}{\mathbf{A}_{i\cdot}\mathbf{v}}, \mathbf{A}_{i\cdot}\mathbf{v} < 0, i \notin \Omega_1^k\}$
Step 7.	If $\eta^{(k)} = 0$, then STOP and claim $\mathbf{x}^{(k)}$ is an optimal solution.
Step 8.	Let $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \eta^{(k)}\mathbf{v}$, $k + 1 \to k$.
Step 9.	Let $\Omega_1^k = \{i \vert \mathbf{A}_{i\cdot}\mathbf{x}^{(k)} = b_i\}$ and $\Omega_2^k = \{j \vert \mathbf{x}_j^{(k)} = 0\}$.
Step 10.	If $\varepsilon < \eta^{(k-1)} < L$, then go to Step 2.
Step 11.	If $\Vert\Omega_1^k + \Omega_2^k\Vert \geqslant n$, then STOP and claim $\mathbf{x}^{(k)}$ is a corner point feasible solution.
Step 12.	Let $\mathbf{A}_q$ be defined as composed of the row vectors of $\mathbf{A}_{i\cdot}\,\forall i \in \Omega_1^k$ and $\mathbf{e}_j^t$ $\forall j \in \Omega_2^k$.
Step 13.	Calculate $\mathbf{d} = -[\mathbf{I} - \mathbf{A}_q^t(\mathbf{A}_q\mathbf{A}_q^t)^{-1}\mathbf{A}_q]\mathbf{c}$.
Step 14.	Let $\eta = \min\{\dfrac{-x_j^{(k)}}{d_j}, d_j < 0, j \notin \Omega_2^k; \dfrac{b_i - \mathbf{A}_{i\cdot}\mathbf{x}^{(k)}}{\mathbf{A}_{i\cdot}\mathbf{d}}, \mathbf{A}_{i\cdot}\mathbf{d} < 0, i \notin \Omega_1^k\}$
Step 15.	Let $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \eta\mathbf{d}$, $k + 1 \to k$.
Step 16.	Let $\Omega_1^k = \{i \vert \mathbf{A}_{i\cdot}\mathbf{x}^{(k)} = b_i\}$ and $\Omega_2^k = \{j \vert x_j^{(k)} = 0\}$. Go to Step 10.

implement the gradient projection method to obtain a corner point of feasible solution. The matrix $\mathbf{M} \equiv \mathbf{I} - \mathbf{A}_q^t(\mathbf{A}_q\mathbf{A}_q^t)^{-1}\mathbf{A}_q$ is the projection matrix corresponding to the boundary. Action by it on any vector yields the projection of that vector onto the boundary. Since $\mathbf{d} = -\mathbf{Mc}$ in Step 13, $\mathbf{d}$ is a feasible direction of descent on the boundary if $\mathbf{d} \neq \mathbf{0}$. That is, we have $\mathbf{c}^t\mathbf{x}^{(k-1)} < \mathbf{c}^t\mathbf{x}^{(k)}$, if $\mathbf{x}^{(k-1)}(= \mathbf{x}^{(k)} + \eta\mathbf{d})$ does not equal $\mathbf{x}^{(k)}$. Step 14 guarantees $\mathbf{x}^{(k+1)}$ is always a boundary point. Before obtaining a basic feasible solution, it is easy to check there are at most n iterations in the loop of Steps 11–16.

3. Computational results

Before we present our computational results, some remarks about the test problems are in order. All test problems essentially were randomly generated, but were first screened by a preprocessor to remove any possible inconsistent problems. The problems were then scaled by normalizing each row of $\mathbf{A}$. For each test problem, there are ten different initial points, which are picked up

Table 2
Comparison of the average numbers of iterations required in each pair of (n, m) test problems[a]

m	*n*	Non_Z (%)	Auxiliary + simplex			Simplex iterations
			Dist	Auxiliary iterations	Simplex iterations	
10	10	90	0.14 (0.11)	1.9 (0.57)	0.6 (0.97)	6
10	10	80	0.64 (0.18)	2.0 (0.0)	1.5 (0.53)	12
10	10	70	0.71 (0.36)	2.0 (0.0)	1.3 (1.06)	6
10	10	60	0.49 (0.30)	2.1 (0.32)	0.6 (0.97)	4
10	10	50	0.65 (0.25)	2.0 (0.0)	1.8 (1.40)	3
10	10	40	0.64 (0.18)	2.0 (0.0)	0.5 (0.97)	6
10	10	30	0.66 (0.16)	1.8 (0.42)	1.0 (0.00)	5
20	20	90	1.25 (0.29)	2.0 (0.0)	10.6 (5.46)	22
20	20	80	1.09 (0.18)	2.0 (0.0)	9.1 (3.63)	12
20	20	70	0.88 (0.25)	2.0 (0.0)	6.6 (4.67)	11
20	20	60	1.58 (0.17)	2.0 (0.0)	12.4 (4.03)	14
20	20	50	1.13 (0.25)	2.0 (0.0)	10.0 (3.77)	25
20	20	40	0.89 (0.19)	2.0 (0.0)	4.4 (2.07)	21
20	20	30	0.78 (0.25)	2.0 (0.0)	5.8 (2.94)	18
30	30	90	1.24 (0.47)	2.0 (0.0)	19.0 (7.86)	28
30	30	80	1.46 (0.34)	2.0 (0.0)	20.7 (7.99)	28
30	30	70	1.51 (0.37)	2.0 (0.0)	16.7 (7.63)	26
30	30	60	1.46 (0.32)	2.0 (0.0)	14.2 (5.59)	29
30	30	50	1.77 (0.29)	2.0 (0.0)	25.9 (7.36)	37
30	30	40	1.63 (0.15)	2.0 (0.0)	15.0 (5.34)	32
30	30	30	1.30 (0.12)	2.0 (0.0)	13.0 (4.85)	36
30	30	20	0.58 (0.29)	2.0 (0.0)	13.7 (3.40)	37

[a]The numbers shown in the parenthesis is the associated standard deviation.

randomly, for the auxiliary algorithm. According to the problem size in terms of m and n of matrix **A**, we report the required iterations versus that required by the simplex method.

The detailed computational results are reported in Table 2. Each row represents the outcome of using the proposed algorithm and the simplex method on the same test problem. The number of constraints and variables in each test problem are denoted by m and n, while the number of nonzeros in the constraint matrix denoted as Non_Z is shown for characterizing its sparsity. The process of the auxiliary + simplex includes the solution improvement before reaching a feasible corner point denoted as auxiliary iterations and the work of attaining the optimal point denoted as simplex iterations. For each test problem, there are ten different initial points, which are picked up randomly, for the proposed method. Because an initial point may not be feasible in LP (2), we describe it by the Euclidean distance between the feasible region and the point, which is denoted by dist.

Table 3
The summary of the average numbers of total iterations required for the simplex method[a]

m	n	Auxiliary + simplex	Simplex
10	10	3.04 (1.01)	6 (2.89)
20	20	11.41 (4.62)	17.57 (5.38)
30	30	19.28 (7.43)	31.63 (4.50)

[a]The numbers shown in the parenthesis is the associated standard deviation.

Table 2 shows that the iteration required in the auxiliary procedure is independent of the position of the initial starting point. That is clearly reflected by the improving search direction proposed in the procedure. Moreover, the proposed improving search direction has also shown that it is more effective in solving sparse matrices while working with the simplex method. Table 3 shows the summary of the computational results. It shows that the auxiliary algorithm help the simplex method consistently reducing the number of required iterations by about 40%.

4. Conclusion and future work

Here we propose an auxiliary algorithm for the simplex method. The auxiliary algorithm has the following advantages: there is no need to know the value of an optimal solution, and no special format for the problem is needed. It is a practical algorithm because its computation complexity is $O(m+n)$. Furthermore, the simplex method with this help can posses a better worst-case complexity bound than those without this help. Although there lacks a proof for the better global complexity for our method, we have explored its computational performance on some random test problems. The computational results show that the auxiliary algorithm does help the simplex method consistently reducing the number of required iterations by about 40%. For future studies, we shall test it with more larger-scale problems as well as take the same test problems used in NETLIB for further comparisons.

Acknowledgements

The authors would like to thank the National Science Council of Republic of China for supporting this work under grant No. NSC 89-2213-E-004-004. Thanks are also due to Alex Hsueh and Wen-Hwa Chuang for their help in the computing work.

References

[1] Karmarkar N. A new polynomial-time algorithm for linear programming. Combinatorica 1984;4:373–95.
[2] Fang SC, Puthenpura S. Linear optimization and extensions. Englewood Cliffs, NJ, USA: Prentice-Hall, 1993.

[3] Monma CL, Morton AJ. Computational experience with a dual affine variant of Karmarkar's method for linear programming. Operations Research Letters 1987;6:261–7.
[4] Adler I, Karmarkar N, Resende MGC, Veiga G. An implementation of Karmarkar's algorithm for linear programming. Mathematical Programming 1989;44:297–335.
[5] Anstreicher KM, Watteyne P. A family of search directions for Karmarkar's algorithm. Operations Research 1993;41(4):759–67.
[6] Todd MJ. A Dantzig–Wolfe-like variant of Karmarkar's interior-point linear programming algorithm. Operations Research 1990;38(6):1006–18.
[7] Anderson ED, Yinyu Ye. Combining interior-point and pivoting algorithms for linear programming. Management Science 1996;42(12):1719–31.
[8] Bixby RE, Gregory JW, Lustig IJ, Marsten RE, Shanno DF. Very large-scale linear programming: a case study in combining interior point and simplex methods. Operations Research 1992;40(5):885–97.

Hsing Luh is a Professor in the Department of Mathematical Sciences at the National Chengchi University in Taiwan, ROC. He received his Ph.D. from North Carolina State University. His interests include queueing theory and queueing networks, Markov decision processes, linear and nonlinear programming, simulation, and research on managerial statistical solutions to business problems.

Ray Tsaih is a Professor in the Department of Management Information Systems at the National Chengchi University in Taiwan, ROC. He serves as department head since 1999. He received his Ph.D. from University of California at Berkeley where he specialized in Operations Research. His interests include neural networks, linear and nonlinear programming, and Finance.