An efficient algorithm for linear programming

V CH VENKAIAH*

SERC and Department of Applied Mathematics, Indian Institute of Science, Bangalore 560012, India

* Present address: Central Research Laboratory, Bharat Electronics Ltd., 25, M G Road, Bangalore 560001, India

MS received 22 September 1989; revised 17 May 1990

Abstract. A simple but efficient algorithm is presented for linear programming. The algorithm computes the projection matrix exactly once throughout the computation unlike that of Karmarkar's algorithm where in the projection matrix is computed at each and every iteration. The algorithm is best suitable to be implemented on a parallel architecture. Complexity of the algorithm is being studied.

Keywords. Direction vector; Karmarkar's algorithm; Moore-Penrose inverse; orthogonal projection; projection matrix; projective transformation.

1. Introduction

In 1979 Khachiyan gave an algorithm called ellipsoid algorithm for linear programming [1]. It has polynomial-time complexity. But it is found to be not superior to the simplex algorithm in practice. A new polynomial-time algorithm based on projective transformation technique was published by Karmarkar 1984 [2]. It is being said that this algorithm is superior to the simplex algorithm even in practice.

In this paper an algorithm is presented [3]. We feel that this algorithm is efficient than the existing algorithms because

1. it computes the projection matrix $P = I - A^+ A$, which requires $O(mn^2)$ operations, exactly once throughout the computation and needs only $O(n^2)$ operations per iteration.
2. it makes use of the information that is available from the recently computed point.

Note that Karmarkar's algorithm computes the projection matrix at each iteration and hence requires $O(n^{2.5})$ operations per iteration.

2. Main idea of the algorithm

Consider the following problem

$$\begin{aligned} &\text{minimize} && C'X \\ &\text{subject to} && X \geq 0. \end{aligned} \tag{P1}$$

This problem is trivial because the solution is obtained by setting those components

of $\mathbf{X}$ to zero that correspond to positive components of $\mathbf{C}$ and other components of $\mathbf{X}$ to 'infinity'. We defer from the usual practice to solve the above problem by an iterative method with an initial feasible solution $\mathbf{X}^0 > \mathbf{0}$. The method is described in the following algorithm.

Algorithm A1

Step 1. Compute an initial feasible solution $\mathbf{X}^0 > \mathbf{0}$.

Step 2. Set $\mathbf{Y} = \mathbf{C}$ and $K = 0$.

Step 3. Compute

$$\lambda = \min \left\{ \frac{X_i^k}{Y_i} : Y_i > 0 \right\}$$

Step 4. Compute

$$\mathbf{X}^{k+1} = \mathbf{X}^k - \varepsilon \lambda \mathbf{Y}$$

where $0 < \varepsilon < 1$.

Step 5. If the optimum is achieved then stop.

Step 6. Set $K = K + 1$, $D_k = \text{diag}(X_1^k, X_2^k, X_3^k, \dots, X_n^k)$

Step 7. $\mathbf{Y} = D_k \mathbf{C}$, go to step 3.

$D_k \mathbf{C}$ can be thought of angular projection of $\mathbf{C}$ onto the coordinate axis.

Now consider the original problem

$$\begin{aligned} &\text{minimize} && \mathbf{C}'\mathbf{X} \\ &\text{subject to} && \mathbf{A}\mathbf{X} = \mathbf{b}, \\ &&& \mathbf{X} \geq \mathbf{0}. \end{aligned} \tag{P2}$$

Let $A^+ = A'(AA')^{-}$, —denotes the generalized inverse. A^+ is called the Moore–Penrose inverse of A . Also, let $P = I - A^+ A$. It can be proved that the columns of P span the null space of A and are normals to the hyperplanes $X_i = 0$ in the null space of A . P is the projection operator that projects every vector in R^n orthogonally onto the null space of A .

To use Algorithm A1 to solve P2, we need to have the initial feasible solution $\mathbf{X}^0$ such that $\mathbf{A}\mathbf{X}^0 = \mathbf{b}$ and the direction vector $\mathbf{Y}$ not in R^n but in the null space of A . This can be achieved by operating P on this vector. With this explanation we now give an algorithm to solve P2.

3. The algorithm

3.1 Algorithm A2

Step 1. Compute an initial feasible solution

$$\mathbf{X}^0 = (X_1^0, X_2^0, X_3^0, \dots, X_n^0)$$

such that $X_i^0 > 0$ for $i = 1, 2, \dots, n$.

Step 2. Compute the projection operator

$$P = I - A^+ A$$

Step 3. Compute

$$\mathbf{Cp} = PC$$

$$\mathbf{Y} = \frac{\mathbf{Cp}}{\|\mathbf{Cp}\|}$$

Step 4. Set $K = 0$

Step 5. Compute

$$\lambda = \min \left\{ \frac{X_i^k}{Y_i} : Y_i > 0 \right\}$$

the problem has unbounded solution if all $Y_i \leq 0$ and at least one $Y_i < 0$. If all $Y_i = 0$ then $\mathbf{X}^k$ is a solution.

Step 6. Compute

$$\mathbf{X}^{k+1} = \mathbf{X}^k - \varepsilon \lambda \mathbf{Y}$$

where $0 < \varepsilon < 1$.

Step 7. If

$$\|\mathbf{X}^{k+1} - \mathbf{X}^k\| < 2^{-L},$$

where L is defined as in [2], then stop.

Step 8. Set

$$K = K + 1, D_k = \text{diag}(d_{ii}),$$

where $d_{ii} = X_i^k / \sqrt{P_{ii}}$ and is the distance between the point $\mathbf{X}^k$ and the hyperplane $X_i = 0$. P_{ii} is positive because $P^2 = P$.

Step 9. Compute

$$\mathbf{Y} = \frac{PD_k \mathbf{Cp}}{\|PD_k \mathbf{Cp}\|}$$

go to step 5.

Distance δ between the point $\mathbf{X}^k$ and the hyperplane $X_i = 0$ is calculated as follows. Consider

$$\mathbf{X}^k - \delta \frac{P_i}{\|P_i\|} \Rightarrow X_i^k - \delta \sqrt{P_{ii}} = 0$$

Since $P_i / \|P_i\|$ is the unit normal to the hyperplane $X_i = 0$ and $\|P_i\| = \sqrt{P_{ii}}$. This implies

$$\delta = \frac{X_i^k}{\sqrt{P_{ii}}}.$$

Step 3 computes the orthogonal projection of $\mathbf{C}$ onto the null space of A and the initial direction vector, step 5 computes the step length, step 6 a new feasible solution at which the objective function value is less than that of the previous value and step 9 the new direction vector and its projection on to the null space of A . It can be easily seen that the computation of the algorithm is dominated by the computation of the new direction vector which requires only $O(n^2)$ operations. Whereas in Karmarkar's algorithm it requires $O(n^{2.5})$ operations.

3.2 Correctness of algorithm A2

Theorem 1. *Let P2 have a bounded solution. Then algorithm A2 converges to a solution of P2.*

Proof.

(i) *Nonnegativity conditions*

Since $\mathbf{X}^0 > 0$,

$$\lambda = \min \left\{ \frac{X_i^k}{Y_i} : Y_i > 0 \right\},$$

and $\varepsilon < 1$ it follows that $\mathbf{X}^{k+1} > 0$ for each k and hence

$$\lim_{k \rightarrow \infty} X_i^k \geq 0$$

for each i .

(ii) *Equality constraints*

Since $A\mathbf{X}^0 = \mathbf{b}$ and for $k = 0$

$$A\mathbf{Y} = \frac{A\mathbf{Cp}}{\|\mathbf{Cp}\|} = \frac{APC}{\|\mathbf{Cp}\|} = \mathbf{0} \quad \text{since } \mathbf{Cp} = PC$$

it follows that $A\mathbf{X}^1 = \mathbf{b}$.

For $k \geq 1$,

$$A\mathbf{Y} = \frac{1}{\|PD_k\mathbf{Cp}\|} APD_k\mathbf{Cp} = \mathbf{0}$$

and hence $A\mathbf{X}^k = \mathbf{b}$.

(iii) *Optimality of the objective function*

For $k = 0$, we have

$$\begin{aligned} \mathbf{C}'\mathbf{X}^k - \mathbf{C}'\mathbf{X}^{k+1} &= \varepsilon\lambda\mathbf{C}' \frac{\mathbf{Cp}}{\|\mathbf{Cp}\|} \\ &= \frac{1}{\|\mathbf{Cp}\|} \varepsilon\lambda\mathbf{C}'\mathbf{Cp} \\ &= \frac{1}{\|\mathbf{Cp}\|} \varepsilon\lambda\mathbf{C}'PCp \quad \text{since } PCp = Cp \\ &= \frac{1}{\|\mathbf{Cp}\|} \varepsilon\lambda\mathbf{C}'P'\mathbf{Cp} \quad \text{since } P = P' \end{aligned}$$

$$\begin{aligned}
&= \frac{1}{\|\mathbf{Cp}\|} \varepsilon \lambda (PC)' \mathbf{Cp} \\
&= \frac{1}{\|\mathbf{Cp}\|} \varepsilon \lambda \mathbf{Cp}' \mathbf{Cp} \text{ since } PC = \mathbf{Cp} \\
&= \varepsilon \lambda \|\mathbf{Cp}\| > 0.
\end{aligned}$$

For $k \geq 1$,

$$\begin{aligned}
\mathbf{C}'\mathbf{X}^k - \mathbf{C}'\mathbf{X}^{k+1} &= \varepsilon \lambda \mathbf{C}'\mathbf{Y} \\
&= \frac{1}{\|PD_k \mathbf{Cp}\|} \varepsilon \lambda \mathbf{C}' PD_k \mathbf{Cp} \\
&= \frac{\varepsilon \lambda}{\|PD_k \mathbf{Cp}\|} \mathbf{Cp}' D_k \mathbf{Cp} \\
&= \frac{\varepsilon \lambda}{\|PD_k \mathbf{Cp}\|} (\sqrt{D_k} \mathbf{Cp})' (\sqrt{D_k} \mathbf{Cp})
\end{aligned}$$

where $\sqrt{D_k} = \text{diag}(\sqrt{d_{ii}})$. $\sqrt{d_{ii}}$ is real because $d_{ii} > 0$. Hence

$$\mathbf{C}'\mathbf{X}^k - \mathbf{C}'\mathbf{X}^{k+1} = \frac{\varepsilon \lambda}{\|PD_k \mathbf{Cp}\|} \|\sqrt{D_k} \mathbf{Cp}\|^2 > 0.$$

Therefore $\mathbf{C}'\mathbf{X}^k$ is a decreasing sequence. Since P2 is assumed to have a bounded solution it follows that $\mathbf{C}'\mathbf{X}^k$ is bounded below. Hence $\mathbf{C}'\mathbf{X}^k$ converges.

3.3 Complexity of algorithm A1

Let L be such that 2^L is numerical infinity and 2^{-L} is numerical zero.

Theorem 2. Algorithm A1 converges to the solution of P1 in $O(L)$ steps.

Proof. Without loss of generality, we assume $\mathbf{X}^0 = (1 \ 1 \ 1 \dots 1)'$. Then

$$\begin{aligned}
\mathbf{X}^1 &= \mathbf{X}^0 - \varepsilon \lambda_1 \mathbf{C}, \\
\lambda_1 &= \min \left\{ \frac{1}{C_i} : C_i > 0 \right\}.
\end{aligned}$$

Let $\lambda_1 = 1/C_j$. Then

$$X_i^1 = 1 - \varepsilon \frac{C_i}{C_j}.$$

Consider

$$\begin{aligned}
X_i^2 &= X_i^1 - \varepsilon \lambda_2 X_i^1 C_i, \\
\lambda_2 &= \min \left\{ \frac{X_i^1}{X_i^1 C_i} : C_i > 0 \right\} \text{ since } X_i^1 > 0 \\
&= \frac{1}{C_j}.
\end{aligned}$$

Therefore

$$\begin{aligned} X_i^2 &= X_i^1(1 - \varepsilon \lambda_2 C_i) \\ &= X_i^1 \left(1 - \varepsilon \frac{C_i}{C_j}\right) \\ &= \left(1 - \varepsilon \frac{C_i}{C_j}\right)^2. \end{aligned}$$

In general,

$$X_i^k = \left(1 - \varepsilon \frac{C_i}{C_j}\right)^k$$

If $C_i < 0$ then X_i^k converges to 2^L for

$$k = \frac{L}{\log_2 \left(1 - \varepsilon \frac{C_i}{C_j}\right)}$$

because

$$\begin{aligned} \left(1 - \varepsilon \frac{C_i}{C_j}\right)^k &\geq 2^L \Rightarrow k \log_2 \left(1 - \varepsilon \frac{C_i}{C_j}\right) \geq L \\ &\Rightarrow k \geq \frac{L}{\log_2 \left(1 - \varepsilon \frac{C_i}{C_j}\right)}. \end{aligned}$$

Similarly, if $C_i > 0$ then X_i^k converges to 2^{-L} for

$$k = \frac{-L}{\log_2 \left(1 - \varepsilon \frac{C_i}{C_j}\right)}$$

Therefore, Algorithm A1 converges to the solution in $O(L)$ steps.

4. Conclusions

Complexity of Algorithm A2 is being studied. Observe that $\lambda_1 = \lambda_2 = \dots$ in Algorithm A1. This observation may be useful in establishing the complexity of Algorithm A2. We feel that the path followed by our algorithm is the same as that of Karmarkar's algorithm. Therefore, Algorithm A2 takes at the most $O(n)$ steps to reach the optimum and hence the complexity of Algorithm A2 is $O(n^3)$. In practice the calculation need to be done in high precision.

Modifying the direction vector $\mathbf{Y} = (PD_k \mathbf{Cp}) / \|PD_k \mathbf{Cp}\|$ in Step 9 of Algorithm A2 as $\mathbf{Y} = (PD_k^{\gamma_k} \mathbf{Cp}) / \|PD_k^{\gamma_k} \mathbf{Cp}\|$ by introducing a real parameter γ_k , another algorithm will be obtained. The resulting algorithm will be better in performance and can handle practical problems with the existing precision if the optimal values can be computed for γ_k . Further details on computing γ_k will be discussed in a future correspondence.

Acknowledgements

The author thanks Dr Eric A Lord for the discussions and Dr S K Sen for getting him research associateship in SERC for pursuing this research work.

References

- [1] Khachiyan L G, *A polynomial algorithm in linear programming*, Doklady Akademii Nauk SSSR 244:S (1979), pp. 1093–1096, translated in Soviet Mathematics Doklady 20:1 (1979), pp. 191–194
- [2] Karmarkar N, *A new polynomial – time algorithm for linear programming*, Tech. Rep., AT & T Bell Labs., New Jersey, (1984)
- [3] Venkaiah V Ch, *An efficient algorithm for linear programming*, Tech. Rep., TR\SERC\KBCS\89 – 003, SERC, IISc, Bangalore 560 012, (1989)