Annals of Optimization with Applications

www.anowa.reapress.com

Ann. Optim. Appl. Vol. 1, No. 1 (2025) 1–11.

Paper Type: Original Article

Interior Points Algorithms in Data Envelopment Analysis (Case Study: An Iranian Bank)

Farhad Hosseinzade Lotfi[1,*], Amin Jabbari[1]

[1] Department of Mathematics, Science and Research Branch, Islamic Azad University, Tehran, Iran; farhad@hosseinzadeh.ir.

Citation:

Received: 13 September 2024 *Revised: 25 November 2024* *Accepted:12 Januray 2025*	Hosseinzade Lotfi, F., & Jabbari, A. (2025). Interior points algorithms in data envelopment analysis (Case study: An Iranian bank). *Annals of optimization with applications, 1*(1), 1-11.

Abstract

In this paper, interior point methods are discussed and analyzed. First, general aspects of Data Envelopment Analysis (DEA) are expressed, then the concept of efficiency, CCR and BCC, BCC-CCR, CCR-BCC models, and the reference sets in DEA models and super-efficiency in ranking are presented. Ellipsoid methods that theoretically prove that linear programming problems are efficiently solvable are also used. The result is significant since when a problem is solvable, practical and efficient algorithms can be developed. The ellipsoid method is not a practical algorithm for solving linear programming problems. Finally, a new category of algorithms, known as the interior point optimization method, is introduced, both valuable and efficient. The other advantage of the process above is that the complexity of the interior point algorithm is in polynomial order. In contrast, the simplex algorithm uses exponential order in the worst cases. This is a good reason to indicate the priority of the interior point optimization algorithm compared with the simplex algorithm. Therefore, this point justifies using the interior point methods in DEA.

Keywords: Interior points algorithms, Data envelopment analysis, Efficiency, Linear programming.

1 | Introduction

Managers or Decision-Making Units (DMUs) always decide to create appropriate strategies to maximize available resources. Limitations of some factors, such as capital and human resources, forced managers to think of ways to use these factors optimally. Assessment problems and helping to improve the performance of units under the manager's supervision are the most important characteristics of a manager related to

Corresponding Author: farhad@hosseinzadeh.ir

appropriate decisions to guide and promote different units. The complexity of problems, the high amount of information, and the effects of external factors on performance are the factors that managers and experts cannot be informed of regarding the functionality of units and cannot find an appropriate solution to improve efficiency and effectiveness. On the other hand, since efficiency is one of the main criteria, we use the Data Envelopment Analysis (DEA) model to evaluate units' performance. This technique was introduced through the CCR model by Charnes Cooper e in [1]. They aimed to measure the performance of units under the manager's control. Six years later, Banker et al. [2] introduced the BCC model. After that, the non-radial FDH models and were designed, and the target was to obtain an approximation of the production function to assess performance and calculate the relative efficiency. Efficiency or having good performance in one unit is a function of internal factors and indicators of organization. On the other hand, the effectiveness of a unit is a function of external organizational factors. However, managers' primary goal is to maximize available resources to achieve the best results and obtain the best results with maximum use of existing resources, which is called productivity.

2 | Literature Review

2.1 | Efficiency

Efficiency means doing things right to achieve the company's goals, while effectiveness means doing the right things where the actual output meets the plan. DMU means the unit that's been able to receive the input vector like.

Create output vector-like:

$$(y_1, y_2, \dots, y_s), (x_1, x_2, \dots, x_m).$$

If, for a decision maker unit, outputs include all prices while input be all expenses, then we can calculate efficiency through the following formula:

$$x_1, x_2, \dots, x_m,$$

$$y_1, y_2, \dots, y_s,$$

$$= \frac{u_1 y_1 + \dots + u_s y_s}{v_1 x_1 + \dots + v_m x_m},$$

where u_r is the price of r the $r = (1, \dots,)$ output, and v_i is the expense of ii,, $i = 1, \dots, m$ the input? Consider a set of n DMUs that the inputs and outputs vectors for $DMUj$ are respectively. Assume that all the input and output vectors for each DMU are nonnegative and are not equal to zero.

$$Y_j = (y_{1j}, y_2, \dots, y_{sj}), X_j = (x_{1j}, x_{2j}, \dots, x_{mj}).$$

Consider the points below for proper evaluation of efficiency under the existing DMUs:

I. All the inputs and outputs of DMUs must be congruent.

II. All inputs and outputs should be collected at a special time or independent of time.

III. The output of each unit should only be dependent on its defined inputs.

IV. $n \geq 3(m + s)$, we obtained experimentally.

By solving the following model:

$$\begin{aligned} &\min \theta \\ &\text{s.t.} \quad (\theta X_p Y_p) \in T_c. \end{aligned}$$

According to the membership conditions T_s, the following model can be obtained, which is called an Envelopment form of the CCR model in the input orientation model:

$$\begin{aligned} \min\ & \theta, \\ \text{s.t.}\ & \sum_{j=1}^{n} \lambda_j X_j \le \theta X_p, \\ & \sum_{j=1}^{n} \lambda_j Y_j \ge Y_p, \\ & \lambda_j \ge 0. \end{aligned} \tag{1}$$

If the optimum answer for the above model $\theta^* < 1$, DMU_P is inefficient, $\theta^* = 1$, DMU_P it is located on the efficient boundary.

In addition, the *Model (1)* is always feasible because $\lambda_j = 0, \lambda_p = 1, \theta = 1, (j = p, j = 1, \dots, n)$ are the feasible answer to this problem; therefore $\theta^* \le 1$, every feasible answer θ is always positive.

Dual of *Model (1)*, which is called the multiplier form of the CCR model in input orientation, is:

$$\begin{aligned} \max\ & u^t Y_p, \\ \text{s.t.}\ & v^t X_p = 1, \\ & u^t Y_j - v^t X_j \le 0, \qquad j = 1, \dots, n, \\ & v \ge 0, \\ & u \ge 0. \end{aligned} \tag{2}$$

Considering the point that envelopment form is always feasible and finite, then the *Model (2)* is always feasible and $u'Y_p \le 1$.

In the above models, if DMU_p something is inefficient, we put it on the border by reducing the input to make it efficient.

A new input means $\theta^* X_p$ some values that can be produced Y_p and $(1-\theta^*)X_p$ are some wasted inputs and called the amount of inefficiency.

According to the above definition, the *Model (3)* indicates the envelop form of CCR in the output orientation:

$$\begin{aligned} \max\ & \varphi, \\ \text{s.t.}\ & \sum_{j=1}^{n} \lambda_j X_j \le X_p, \\ & \sum_{j=1}^{n} \lambda_j Y_j \ge \varphi Y_p, \\ & \lambda_j \ge 0. \end{aligned} \tag{3}$$

Based on a similar discussion, the model above is feasible. The envelop form of CCR in the input orientation is

$$
\begin{aligned}
&\min \ \theta, \\
&\text{s.t.} \quad \sum_{j=1}^{n} \lambda_j X_j + s_i^- = X_{ip}\theta, i = 1,\dots,m, \\
&\qquad \sum_{j=1}^{n} \lambda_j Y_j - s_r^+ = Y_{rp}, r = 1,\dots,s, \\
&\qquad \lambda_j \geq 0, j = 1,\dots,n, \\
&\qquad s_i^- \geq 0, i = 1,\dots,m, \\
&\qquad s_r^+ \geq 0, r = 1,\dots,s.
\end{aligned}
$$

In the above model, if $\theta^* < 1$ then DMU_p is inefficient, but if $\theta^* = 1$, then there are two conditions as follows:

Definition 1. If in the enveloped form of CCR in input orientation $\theta^* = 1$ and all optimized solutions, the auxiliary variable values equal zero, then DMU_p it is strong efficient or Pareto.

Definition 2. If in envelop form of CCR in input orientation $\theta^* = 1$ and some optimized solutions, at least one of the auxiliary variable values is not equal to zero, then DMU_p it is weak efficient.

Definition 3. In the above models, when $\theta^* = 1$, DMU_p radial efficiency is present, this efficiency can be weak or strong.

Definition 4. The amount θ^* in the CCR envelope model used to evaluat DMU_p e the technical efficiency of decision-makers $(1-\theta^*)$ is called technical inefficiency.

DEA provides a theoretical framework for performance analysis and efficiency measurement [3]. This model includes sets of linear programming techniques that create the efficient boundary by using the observed data and then evaluating and measuring the decision-maker unit [4], [5].

Unlike many conventional models in microeconomic theory, the DEA model can have multiple inputs and outputs. Moreover, it doesn't need information about the prices of goods and services.

2.2 | Interior Point Methods

Interior point methods (also referred to as barrier methods) are a class of algorithms for solving linear and nonlinear convex optimization problems [6], [7]. They were inspired by Karmarkar's algorithm, developed by Karmarkar [8] in 1984 for linear programming.

Contrary to the simplex method, it reaches an optimal solution by traversing the interior of the feasible region [9]. Any convex optimization problem can be transformed into minimizing (or maximizing) a linear function over a convex set. The idea of encoding the feasible set using a barrier and designing barrier methods was studied in the early 1960s by, amongst others, and Fiacco et al. [10]. These ideas were mainly developed for general nonlinear programming [11]. Still, they were later abandoned due to the presence of more competitive methods for this class of problems (e.g., sequential quadratic programming) [12].

To understand the key concepts of the interior point approaches, they are divided into three major categories, and the geometry of each is discussed:

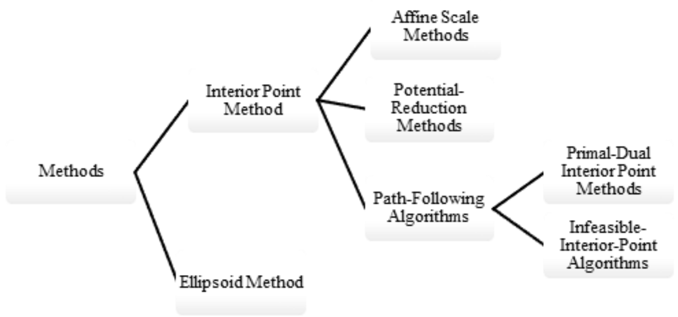

Fig. 1. The key concepts of the interior point approaches.

Affine scale methods

This is the simplest interior point algorithm. It is simple and also has exemplary practical implementation. In some ways, it is closely related to the simplex method. If this algorithm starts from the nearest vertex point, it moves approximately along the edges of the feasible set.

Potential-reduction methods

This algorithm is in the second category of interior point methods [13]. Instead of improving the optimality by decreasing the objective function value, we do that by reducing the value of a nonlinear potential function. The potential function follows these two objectives:

I. Reducing the amount of target.

II. To stay away from the border of feasible set.

This algorithm has polynomial complexity.

Path-following algorithms

The path-following algorithm contains three key ingredients in its loop: The predictor, the corrector, and the step size control. For the predictor step, there are three commonly used predictors: The tangent (or Euler) predictor, the secant predictor, and the Cubic (or Hermite) predictor, each of which has its own merit.

Infeasible primal-dual interior point methods

An example of the algorithm path-following is available, which has been proven in practice and is very successful. This method starts a starting point $P^{\circ}, s^{\circ} > 0, x^{\circ} > 0$ that is not required to be feasible for primal or dual, i.e $Ax^{\circ} \neq b$ $A'P^{\circ} + s^{\circ} \neq c$ [14].

This algorithm is also standard, like Primal-Dual path-following. The direction of Newton $d = (d_x^k, d_p^k, d_s^k)$ is precisely similar to the previous condition and can be calculated by solving the following system:

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A' & I \\ S_k & 0 & X_k \end{bmatrix} \begin{bmatrix} d_x^k \\ d_p^k \\ d_s^k \end{bmatrix} = - \begin{bmatrix} Ax^k = b \\ A'P^k + s^k = c \\ X_k S_k e - \mu^k e \end{bmatrix}.$$

It has been proven that the above method is convergent to an optimized answer and has exemplary practical implementation.

2.3 | Comparison of Interior-Point Methods Versus Simplex Algorithms

The current opinion is that the efficiency of exemplary implementations of simplex-based and interior point methods is similar to routine linear programming applications [15–18]. However, for specific LP problems, one kind of solver is better than another (sometimes much better) [19]. Later, the experimental result will show that the interior point method for large-scale linear programming problems is more accurate than the simplex method [20]. In contrast, if we want to compare the simplex method with Karmarkar's (interior point method) in the runtime aspect, we can observe that neither is faster than the other in all problems [21], [22].

The practical efficiency of both methods depends strongly on the details of their implementation. On the other hand, the number of iterations required by Karmarkar's method is typically between 10 and 100, while the Simplex method needs 2n − 3n iterations, where n is the number of primal variables.

Thus, we can discuss later that the IPMs (interior point methods) are generally better for large-scale problems.

Algorithm

Kojima et al. [13] constructed the first primal-dual interior-point method for linear programming based on the work of Megiddo [23], [24]. The following general framework captures the essence of this algorithm and contains the majority of the primal-dual interior-point methods that can be found in the literature.

Algorithm 1. Generic Kojima et al. [24] primal-dual algorithm.

Given a strictly feasible point $(x^\circ, y^\circ, z^\circ)$. For k = 0,1,2,..., do:

I. Choose $\sigma^k \in (0,1]$ and set $\mu^k = \sigma^k (x^k)^T z^k / n$.

II. Solve the following system $(\Delta x^k, \Delta y^k, \Delta z^k)$:

$$F'(x^k, y^k, z^k)\begin{pmatrix}\Delta x\\ \Delta y\\ \Delta z\end{pmatrix} = -F(x^k, y^k, z^k) + \mu^k \begin{pmatrix}0\\0\\0\end{pmatrix}.$$

III. Choose $T^k \in (0,1)$ and compute the step length $\sigma^k = \mathrm{Min}\left(1, T^k \hat{\alpha}^k\right)$, where:

$$\hat{\alpha}^k = \frac{-1}{\mathrm{Min}\left((X^k)^{-1}\Delta x^k, (Z^k)^{-1}\Delta z^k\right)}.$$

IV. From the new iterate:

$$(x^{k+1}, y^{k+1}, z^{k+1}) = \left(x^k, y^k, z^k\right) + \alpha^k (\Delta x^k, \Delta y^k, \Delta z^k).$$

2.4 | Numerical Experience

This section discusses the numerical results obtained using the information mentioned algorithm for DEA problems.

Our performed experiment used a code implemented in MATLAB for large-scale data. The case study is one of the branches of MELAT bank in Iran. Using the interior point algorithm, 1928 MELAT bank branches have been assessed for the performance of DMUs. These DMUs contain 3 inputs and 9 outposts in which the inputs are:

I. I_1 personnel score.

II. I_2 margin.

III. I_3 and, claims respectively.

The outputs are:

I. O_1: Facilities.

II. O_2: The total amount of four deposits.

III. O_3: Long-term deposit.

IV. O_4: Current deposit.

V. O_5: Loan deposit.

VI. O_6: Short-term deposit.

VII. O_7: Received interest.

VIII. O_8: Received commission.

IX. O_9: Other resources.

The starting point is not necessarily feasible. The code generates a sequence of iterates that approach feasibility and drive the gap to zero. We say that a problem is solved to an accuracy of 10^m for some positive integer m if the algorithm is terminated when:

$$\max\left(\frac{|c^t x^k - b^T y^k|}{1+|b^T y^k|}, \frac{\|Ax^k - b\|_1}{1+\|x\|_1}, \frac{\|A^T y^k + z^k - c\|_1}{1+\|y^k\|_1+\|z^k\|_1}, \frac{\|X^k z^k - e\|_2}{x^T z / n}\right) \le 10^{-m}.$$

In this study, all the problems were solved to an accuracy of 10^8. The algorithm stops when the problem is solved to the given accuracy or when the number of iterations equals 200.

We used SeDuMi 1.05, an add-on for MATLAB that lets you solve optimization problems with linear, quadratic, and semi-definiteness constraints [25]. The reason for using Sedumi is that this toolbox can efficiently solve large-scale optimization problems by exploiting sparsity.

Our model is a CCR Input Oriented model:

$$\begin{aligned}
\min\ & \theta, \\
\text{s.t.}\ & \sum_{j=1}^{n} \lambda_j X_j \le \theta X_p, \\
& \sum_{j=1}^{n} \lambda_j Y_j \ge Y_p, \\
& \lambda_j \ge 0.
\end{aligned}$$

We can standard the model by adding Slack variable:

$$\begin{aligned}
\min\ & \theta, \\
\text{s.t.}\ & \sum_{j-1}^{n} \lambda_j X_j + s_i^- = X_{ip}\theta, i=1,\dots,m, \\
& \sum_{j-1}^{n} \lambda_j Y_j - s_r^+ = Y_{rp}, r=1,\dots,s, \\
& \lambda_j \ge 0, j=1,\dots,n, \\
& s_i^- \ge 0, i=1,\dots,m, \\
& s_r^+ \ge 0, r=1,\dots,s.
\end{aligned}$$

Therefore, our model is: LP:

$$12\,\text{Constarins}:\left(\begin{array}{l}
9.55\lambda_1+13.76\lambda_2+....+8.25\lambda_{1923}+ \\
4.45\lambda_{1924}+6.24\lambda_{1926}+4.51\lambda_{1927}\leq\theta x_{1p}, \\
1441049759\lambda_1+1223216971\lambda_2+....+ \\
842550647\lambda_{1926}+469974998\lambda_{1927}\leq\theta x_{2p}, \\
242470622\lambda_1+321087157\lambda_2+....+ \\
170694174\lambda_{1926}+9838700\lambda_{1927}\leq\theta x_{3p}, \\
51607712527\lambda_1+1.0658E+11\lambda_2+....+ \\
16708693165\lambda_{1926}+20827421921\lambda_{1927}\geq Y_{1p}, \\
\quad . \qquad\qquad . \qquad\qquad . \\
\quad . \qquad\qquad . \qquad\qquad . \\
\quad . \qquad\qquad . \qquad\qquad . \\
189820238\lambda_1+187178954\lambda_2+....+ \\
32752670\lambda_{1926}+28477102\lambda_{1927}\geq Y_{9p}.
\end{array}\right)$$

The result of assessing for DMU= 1 is gathered in *Table 1*:

Table 1. The result of assessing for DMU= 1.

Rows	λ^*
λ 630	3.0538
λ 831	1.3000
λ 836	0.5040
λ 1547	1.7611
λ 1720	0.0361
λ 1929	0.5792
λ 1934	0.0521
λ 1936	0.0507
λ 1937	0.0014
λ 1940	0.0001

For other rows, the $\lambda^*=0$.

The other information is gathered in *Table 2*.

Table 2. No caption.

Input Right-Hand Side	Output Right-Hand Side	Primal Solution	Dual Solution
0	0.0107	0.0750	3.5614e-008
0	0.0067		
0	0.0013		
	0.0020		
	0.0013		
	0.0021		
	0.0003		
	0.0000		
	0.0001		

Iter seconds digits c*x, b*y
25, 0.8, Inf 5.7917206034e-001, 5.7982122630e-001.
|Ax-b|= 1.1e-005, [Ay-c] _+ = 2.9E-017, |x|= 3.8e+000, |y|= 7.9e+003.

Table 3 shows the result of the assessment for DMU= 100, 500, 1900.

Table 3. For DMU= 100, 500, 1900.

Rows	DMU=500	DMU=100	DMU=1900
λ 364	7.9749	0.2669	0
λ 630	0	0.363	2.0557
λ 831	0	0	1.2244
λ 836	5.8765	0.5668	0
λ 983	0.2353	0.245	0
λ 1455	0	0	0.9312
λ 1547	0	0	2.7475
λ 1720	0.0098	0	0
λ 1929	0.3598	0.3633	0.6511
λ 1932	0.0003	0.0002	0
λ 1934	0.0047	0.006	0.04
λ 1936	0.0004	0.0058	0.0376
λ 1937	0.0042	0.0002	0.0024

For other rows, the $\lambda^* = 0$. The other information is gathered in *Table 4*. For DMU= 100.

Table 4. For DMU= 100.

Input Right-Hand Side	Output Right-Hand Side	Primal Solution	Dual Solution
0	0.0049	0.0750	2.5060e-007
0	0.0040		
0	0.0008		
	0.0006		
	0.0012		
	0.0015		
	0.0000		
	0.0000		
	0.0000		

Iter seconds digits c*x, b*y
25, 0.8. Inf 3.6325708088e-001, 3.6325708120e-001
|Ax-b|= 6.2e-010, [Ay-c]_+ = 0.0E+000, |x|= 8.5e-001, |y|= 1.3e+004

Table 5. For DMU=500.

Input Right-Hand Side	Output Right-Hand Side	Primal Solution	Dual Solution
0	0.0357	0.0357	0.0357
0	0.0165		
0	0.0074		
	0.0036		
	0.0010		
	0.0045		
	0.0002		
	0.0000		
	0.0000		

Iter seconds digits c*x, b*y.
26, 0.8, Inf 3.5976135603e-001, 3.5976135620e-001
|Ax-b| = 3.4e-010, [Ay-c]_+ = 8.0E-012, |x|= 9.9e+000, |y|= 2.2e+003

Table 6. For DMU=1900.

Input Right-Hand Side	Output Right-Hand Side	Primal Solution	Dual Solution
0	0.0086	0.0750	2.5176e-008
0	0.0084		
0	0.0029		
	0.0015		
	0.0002		
	0.0038		
	0.0002		
	0.0000		
	0.0000		

Iter seconds digits c*x, b*y.
24, 0.7. Inf 6.5113532930e-001, 6.5113533031e-001.
|Ax-b| = 6.7e-009, [Ay-c]_+ = 0.0E+000, |x|= 3.8e+000, |y|= 5.3e+003

3 | Conclusion

Interior point methods have a significant advantage in practically implementing linear programming. However, better detection performance of simplex or interior point methods depends on problems and samples. But some factors are:

I. The simplex method is not good on significant problems, especially problems with degenerate solutions, while the interior point method is different. Therefore, it is expected that when the problem has a degenerate solution, the interior point method will act better than the simplex algorithm.

II. The most important step in calculating an interior point method is to solve the following system:

$$(AX_k^2A')d = f.$$

III. Since the degenerate is automatically too high in DEA, we expect the interior point method to act better than the simplex method.

IV. The following path algorithm is the category of interior point algorithm, which has the best time from the point of complexity and has interesting geometric properties. Some of the issues that will be discussed in the future are:

- *Reducing the solving time of heavy problems in DEA using interior point optimization algorithm.*
- *Analyzing DEA models in GAMS, MATLAB, and SEDUMI (plugging in MATLAB) and expressing the advantages and disadvantages of this software.*

Conflict of Interest

The authors declare no conflict of interest.

Data Availability

All data are included in the text.

Funding

This research received no specific grant from funding agencies in the public, commercial, or not-for-profit sectors

References

[1] Charnes, A., Cooper, W. W., & Rhodes, E. (1978). Measuring the efficiency of decision making units. *European journal of operational research*, 2(6), 429–444. https://doi.org/10.1016/0377-2217(78)90138-8

[2] Banker, R. D., Charnes, A., & Cooper, W. W. (1984). Some models for estimating technical and scale inefficiencies in data envelopment analysis. *Management science, 30*(9), 1031-1142. https://doi.org/10.1287/mnsc.30.9.1078

[3] Cooper, W. W., Seiford, L. M., Tone, K. (2007). *Data envelopment analysis: A comprehensive text with models, applications, references and DEA-solver software* (Vol. 2). Springer. https://doi.org/10.1007/b109347

[4] Bertsimas, D., & Tsitsiklis, J. N. (1997). *Introduction to linear optimization* (Vol. 6). Athena Scientific Belmont, MA. https://B2n.ir/tb1508

[5] Cooper, A. C. W. W., & Charnes, W. (1962). Programming with linear fractional functionals. *Naval research logistics quarterly*, *9*(3), 181–186. https://doi.org/10.1002/nav.3800090303

[6] Alizadeh, F. (1995). Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM journal on optimization*, *5*(1), 13–51. https://doi.org/10.1137/0805002

[7] Alizadeh, F., Haeberly, J. P. A., & Overton, M. L. (1998). Primal-dual interior-point methods for semidefinite programming: Convergence rates, stability and numerical results. *SIAM journal on optimization*, *8*(3), 746–768. https://doi.org/10.1137/S1052623496304700

[8] Karmarkar, N. (1984). A new polynomial-time algorithm for linear programming. *Proceedings of the Sixteenth annual acm symposium on theory of computing* (pp. 302–311). Association for Computing Machinery. https://doi.org/10.1145/800057.808695

[9] Borgwardt, K. H. (2012). *The simplex method: A probabilistic analysis* (Vol. 1). Springer Science & Business Media. https://doi.org/10.1007/978-3-642-61578-8

[10] Fiacco, A. V, & McCormick, G. P. (1964). Computational algorithm for the sequential unconstrained minimization technique for nonlinear programming. *Management science*, *10*(4), 601–617. https://doi.org/10.1287/mnsc.10.4.601

[11] Andersen, E. D., & Andersen, K. D. (1995). Presolving in linear programming. *Mathematical programming*, *71*, 221–245. https://doi.org/10.1007/BF01586000

[12] Andersen, K. D. (1996). A modified Schur-complement method for handling dense columns in interior-point methods for linear programming. *ACM transactions on mathematical software (TOMS)*, *22*(3), 348–356. https://doi.org/10.1145/232826.232937

[13] Kojima, M., Shindoh, S., & Hara, S. (1997). Interior-point methods for the monotone semidefinite linear complementarity problem in symmetric matrices. *SIAM journal on optimization*, *7*(1), 86–125. https://doi.org/10.1137/S1052623494269035

[14] Jansen, B., Roos, C., & Terlaky, T. (1996). A polynomial primal-dual Dikin-type algorithm for linear programming. *Mathematics of operations research*, *21*(2), 341–353. https://doi.org/10.1287/moor.21.2.341

[15] Chvatal, V. (1983). *Linear programming."* W. H. Freeman and Company, New York. https://B2n.ir/pb2293

[16] Freund, R. M. (1991). Polynomial-time algorithms for linear programming based only on primal scaling and projected gradients of a potential function. *Mathematical programming*, *51*(1), 203–222. https://doi.org/10.1007/BF01586933

[17] Freund, R. W., & Jarre, F. (1997). A QMR-based interior-point algorithm for solving linear programs. *Mathematical programming*, *76*, 183–210. https://doi.org/10.1007/BF02614383

[18] Gonzaga, C. C. (1990). Polynomial affine algorithms for linear programming. *Mathematical programming*, *49*, 7–21. https://doi.org/10.1007/BF01588776

[19] Ahuja, R. K., Magnanti, T. L., Orlin, J. B. (1993). *Network flows: Theory, algorithms, and applications* (Vol. 1). Prentice Hall Englewood Cliffs, NJ. https://B2n.ir/jg9167

[20] Adler, I., & Monteiro, R. D. C. (1991). Limiting behavior of the affine scaling continuous trajectories for linear programming problems. *Mathematical programming*, *50*, 29–51. https://doi.org/10.1007/BF01594923

[21] Jarre, F., & Saunders, M. A. (1995). A practical interior-point method for convex programming. *SIAM journal on optimization*, *5*(1), 149–171. https://doi.org/10.1137/0805008

[22] Karmarkar, N. K., Lagarias, J. C., Slutsman, L., & Wang, P. (1989). Power series variants of Karmarkar-type algorithms. *AT & T technical journal*, *68*(3), 20–36. https://doi.org/10.1002/j.1538-7305.1989.tb00316.x

[23] Fourer, R., & Mehrotra, S. (1993). Solving symmetric indefinite systems in an interior-point method for linear programming. *Mathematical programming*, *62*(1), 15–39. https://doi.org/10.1007/BF01585158

[24] Kojima, M., Mizuno, S., & Yoshise, A. (1989). *A primal-dual interior point algorithm for linear programming*. Springer. https://doi.org/10.1007/BF01582151

[25] Den Hertog, D. (2012). *Interior point approach to linear, quadratic and convex programming: Algorithms and complexity* (Vol. 277). Springer Science & Business Media. https://doi.org/10.1007/978-94-011-1134-8