P-ISSN 2586-9000
E-ISSN 2586-9027
Homepage : https://tci-thaijo.org/index.php/SciTechAsia
Science & Technology Asia
Vol.28 No.1 January - March 2023
Page: [48-59]
*Original research article*

# The Circuit Direction Search Algorithm for Solving Two-Dimensional Linear Programming Problems

Panthira Jamrunroj, Aua-aree Boonperm*

*Department of Mathematics and Statistics, Faculty of Science and Technology, Thammasat University, Pathum Thani 12120, Thailand*

Received 21 April 2022; Received in revised form 31 October 2022
Accepted 31 October 2022; Available online 20 March 2023

**ABSTRACT**

The simplex method is the favored method that can solve a linear programming model. One of the important steps of the simplex method to speed up the algorithm is an effective pivot rule to exchange entering and leaving variables. The double-pivot rule that exchanges two entering and two leaving variables in each iteration is one of the interesting pivots. Two pivot variables can be obtained by solving a special two-dimensional linear programming problem. If an effective algorithm to solve a two-dimensional linear programming problem is established, it can speed up the simplex method. Therefore, in this paper, a new algorithm that is an interior search technique, called the *circuit direction search algorithm*, for solving a two-dimensional linear programming problem is proposed. It uses an appropriate circuit as a direction for updating a solution. Then, an associated dual variable is computed to check the optimality for terminating the algorithm. From the computational results, we found that the proposed algorithm could reduce the average number of iterations and the running time compared with the interior point method, the slope algorithm, and the simplex method.

**Keywords:** Circuit direction; Double pivot rule; Linear programming problem; Pivot rule; Search direction; Simplex method

## 1. Introduction

Linear programming is an optimization technique to investigate the best solution in which a linear objective function is minimized or maximized subject to linear inequality or equality constraints. In real-world problems, it is widely used to solve many industrial problems required to achieve the best outcome, such as a production planning problem [1–3], a travel-

***Corresponding author:** aua-aree@mathstat.sci.tu.ac.th
**doi:** 10.14456/scitechasia.2023.5

ing salesman problem [4, 5], an assignment problem [6, 7], or a transportation problem [8, 9]. The graphical method can easily solve a linear programming problem in two or three-dimensional problems with small constraints. However, if the problem has many constraints, it is not practical.

The first practical method used to solve a linear programming problem was presented in 1947 by Dantzig [10], namely the simplex method. The simplex method is an iterative method that exchanges only one variable to update a solution. It starts at a feasible solution and moves to a feasible adjacent solution by moving on the edge of the feasible region until the optimal solution is reached. However, in 1972, Klee and Minty [11] presented the Klee-Minty cube that showed the simplex method's worst-case computational time.

Later, in 1976, Shamos and Hoey [12] proposed a new algorithm used to solve a two-dimensional linear programming problem. This algorithm forms the intersection of geometric objects in the plane. Then, it starts by considering $n$ line segments in the plane and finding all intersecting pairs. After that, they use the $O(n \log n)$ algorithm to determine any two intersection points and determine whether two simple plane polygons intersect. They proved that a linear programming problem with two variables and $n$ constraints could be solved in $O(n \log n)$ time while the simplex method requires $O(n^2)$ time.

In 1983, Megiddo [13] introduced the linear-time algorithm to solve a linear program in $\mathbb{R}^2$. This method searches for the smallest circle enclosing $n$ given points in the plane. Moreover, it disproves a conjecture by Shamos and Hoey that their algorithm requires $O(n \log n)$ time. In addition, a 2-dimensional linear program that has a closely related problem called a separability problem in $\mathbb{R}^2$ is constructed and solved in $O(n)$ time. This algorithm corrects the error in Shamos and Hoey's paper that it can solve in the $O(n \log n)$ time.

In 1984, Dyer [14] also proposed the algorithm which requires $O(n)$ time to solve a 2-dimensional linear programming problem. The concept of this algorithm is the utilization of convexity, a well-known idea for creating the linear time algorithm. This algorithm starts at any feasible point, and the objective function coefficient values, $c_1$ and $c_2$, are considered. If $c_1 = c_2 = 0$, then this feasible point is optimal. Otherwise, the problem is transformed to provide a gradient direction of the objective function. After that, the optimal solution is gained by performing the pairing algorithm. All processes of Dyer's algorithm require $O(n)$ time for each process.

Recently, Vitor and Easton [15] proposed a new method to solve a two-dimensional linear programming problem called the slope algorithm. This algorithm can start when the problem satisfies the following conditions: the coefficient values of the objective function are positive, and all right-hand side values are nonnegative. Not only can it solve the two-dimensional problem, but also it can be applied to identify two variables into the basis called the double pivot simplex method. This pivot requires the optimal basis from the slope algorithm to indicate leaving variables. Then, it repeats steps until the optimal solution to the original problem is obtained. The computational result showed that this algorithm could reduce the iterations of the simplex method. However, since the slope algorithm may start from an exterior point before it moves to the feasible region, it may take more time to identify two leaving variables in each iteration.

Moreover, in some cases, the slope

algorithm exchanges only one variable into the basis, similar to the simplex method. So, it is useless in this case. Later, in 2021, Jamrunroj and Boonperm [16] published a new technique to improve the slope algorithm in the case of only one variable indicated as a leaving variable. This technique considers only the coefficients of constraints to identify the optimal solution to the special two-dimensional linear programming problem. Nevertheless, for the case of double pivots, the slope algorithm is also performed to identify the leaving variables to the general linear programming problem.

Since the slope algorithm starts by considering the slope of constraints, if the problem has many redundant constraints, it may start from an infeasible point, and it will take more iterations to move to the optimal solution. Therefore, if we have an algorithm that starts from a feasible point and moves to another feasible point, then it may reduce the iterations required to reach the optimal solution. This technique is called the interior search technique.

The interior search technique was first presented by Karmarkar [17] in 1984, called *Karmarkar's algorithm* or the *interior point algorithm*. The fundamental characteristic of the interior point algorithm is a *search direction*. Karmarkar's algorithm uses the steepest descent search with scaling and projection for each iteration to update a solution. This algorithm is quite fast in approaching the neighborhood of the optimal solution but slows while approaching the optimal solution. Karmarkar's algorithm has gained interest in theoretical analysis, practical implementations, and its many variants. One of these variants is the projective method proposed by Karmarkar [17] and studied by many researchers such as Yamashita [18], Anstrecher [19, 20], and Ye [21, 22]. A search direction is not only performed in the primal problem but also applied to the dual problem. The search direction for the dual problem was first presented by Yamashita [18] in 1986. Furthermore, many researchers proposed the search directions in various forms, as summarized by Hertog and Roos [23].

Recently, Rujira et al. [24] proposed a new interior search algorithm to solve a linear programming problem. This algorithm starts by relaxing all non-acute constraints to find an initial feasible point. Then, it jumps to the improved objective feasible point along with the improving direction, a composition of the objective vector, and its constraint. After that, the rest of the non-acute constraints are restored, and the dual simplex method is performed to find the optimal solution.

The investigation of a search direction to improve a solution is important. One of the interesting directions that researchers have been interested in is a circuit. It was first proposed by Rockafellar [25], in 1969, as the form of the *elementary vectors* of a subspace. Furthermore, the circuits have improved the augmentation schemes for solving linear and integer-linear programming problems.

In 2015, Loera et al. [26] proposed the steepest-descent augmentation scheme for solving a linear programming problem in the standard form that uses the *steepest-descent circuit* as a direction at each step. The steepest-descent circuit is defined as a *strictly feasible circuit* $\mathbf{g}$ at a feasible solution that minimizes $\frac{\mathbf{c}^T\mathbf{g}}{\|\mathbf{g}\|_1}$ over all such circuits where $\mathbf{c}$ is an objective function vector. A limitation of the steepest-descent circuit is that it is obtained by solving the characteristic linear programming problem: it spends more time at each iteration.

However, in a two-dimensional problem, Borgwardt et al. [27] showed that the circuits could be obtained easily by identifying the edge directions of all constraints. So, the step to searching for the circuit in a two-dimensional linear programming problem is relatively easy. Therefore, the optimal solution may be rapidly obtained if a potential circuit is used as a search direction in a two-dimensional problem.

Due to the limitations mentioned above, in this paper, we propose a new interior search approach to solve a special two-dimensional linear programming problem by applying a circuit as a direction. This algorithm is called the *circuit direction search algorithm*. It starts from the origin point, and the steepest-descent circuit is used to improve the solution, which is obtained easily without solving the characteristic linear programming problem. To show the efficiency of the algorithm, the randomly generated linear programming problems were tested, and the average number of iterations and the running time were compared with the interior point method, the slope algorithm, and the simplex method.

This paper is organized as follows. In Section 2, the definition of a circuit is described, followed by the proposed algorithm in Section 3. Then, the computational results are presented in Section 4. Finally, the results and findings are described and concluded.

## 2. Preliminaries

Since this paper presents a new interior search technique that uses a circuit as a direction, the definition of a circuit is described first.

### 2.1 Definition of Circuit

Consider the polyhedron P:

$$P = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{Ax} = \mathbf{b}, \mathbf{Bx} \leqslant \mathbf{d}\},$$

where $\mathbf{A} \in \mathbb{R}^{m_A \times n}$, $\mathbf{B} \in \mathbb{R}^{m_B \times n}$, and $rank \begin{pmatrix} \mathbf{A} \\ \mathbf{B} \end{pmatrix} = n$. Recall from [27,28], the definition of the set of circuits $C(\mathbf{A},\mathbf{B})$ of $P$ is as follows.

**Definition 2.1.** The set of circuits $C(\mathbf{A},\mathbf{B})$ of a polyhedron $P = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{Ax} = \mathbf{b}, \mathbf{Bx} \leqslant \mathbf{d}\}$ consists of all $\mathbf{g} \in ker(\mathbf{A}) \setminus \{\mathbf{0}\}$ normalized to coprime integer components for which $\mathbf{Bg}$ is support-minimal over $\{\mathbf{Bx} | \mathbf{x} \in ker(\mathbf{A}) \setminus \{\mathbf{0}\}\}$.

Graver [29] stated that the set of circuits consists of all potential edge directions of $P$ as the right-hand side vectors $\mathbf{b}$ and $\mathbf{d}$ vary. Then, any normalization which results in a unique positive and negative representative for each of these one-dimensional directions can be used when working with circuits. So, any positive scalar multiple of a circuit $\mathbf{g} \in C(\mathbf{A},\mathbf{B})$, where $C(\mathbf{A},\mathbf{B})$ consists of all potential edge direction of $P$, is called a *circuit direction* of $P$ by geometrical reasons. Fig. 1 illustrates the circuits of each constraint in two dimensions.

Moreover, circuits are used to develop augmentation schemes for solving linear programming problems in which successive, improving, maximal steps are taken along circuit directions until an optimal solution is reached or the problem is unbounded. Then, De Loera et al. [26] presented an augmentation scheme for solving the standard linear programming problem by using a steepest-descent circuit as a direction to update a solution at each iteration. The authors define a steepest-descent circuit as a strictly feasible circuit that improves a search direction at a feasible solution. Thus, a strictly feasible circuit is first defined as follows.

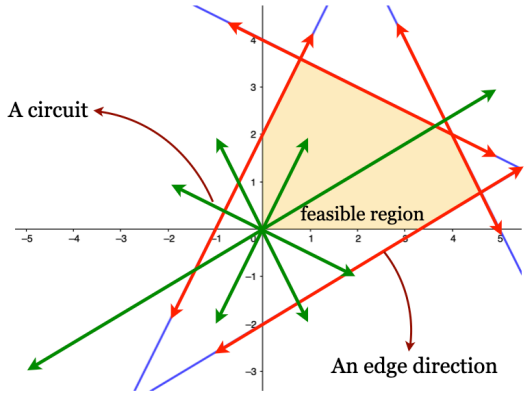



**Fig. 1.** The circuits of each constraint.

**Definition 2.2.** Consider the standard linear programming problem:

$$SLP = \min\{\mathbf{c}^T\mathbf{x}|\mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geqslant \mathbf{0}\},$$

with a feasible solution $\mathbf{x}_0$. A direction $\mathbf{d}$ is said to be strictly feasible at $\mathbf{x}_0$ if $\mathbf{x}_0 + \alpha\mathbf{d} \in SLP$ for some $\alpha > 0$.

Next, the definition of a steepest-descent circuit of the standard linear programming problem is recalled as the following definition.

**Definition 2.3** ([26])**.** Consider the standard linear programming problem:

$$SLP = \min\{\mathbf{c}^T\mathbf{x}|\mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geqslant \mathbf{0}\},$$

with a feasible solution $\mathbf{x}_0$. A steepest-descent circuit is a strictly feasible circuit $\mathbf{g} \in C(\mathbf{A})$ at $\mathbf{x}_0$ that minimizes $\frac{\mathbf{c}^T\mathbf{g}}{\|\mathbf{g}\|_1}$ over all such circuits.

However, the steepest-descent circuit for SLP is difficult to obtain. In each iteration, it is gained by solving the following linear programming problem over a polyhedral set $P = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{B}\mathbf{x} \leqslant \mathbf{d}\}$:

$$\begin{aligned} \min \quad & \mathbf{c}^T\mathbf{x} \\ \text{s.t.} \quad & A\mathbf{x} = \mathbf{0}, \\ & B\mathbf{x} = \mathbf{y}^+ - \mathbf{y}^-, \\ & \mathbf{y}_i^+ = 0, \ \forall i : (B\mathbf{x}_0)_i = \mathbf{d}_i, \\ & \sum_{i=1}^{m_B} \mathbf{y}_i^+ + \sum_{i=1}^{m_B} \mathbf{y}_i^- = 1, \\ & \mathbf{y}^+, \mathbf{y}^- \geqslant \mathbf{0}. \end{aligned} \tag{2.1}$$

In 2016, Borgwardt et al. [27] mentioned that the set of circuits consists exactly of all edge directions of the polyhedron $\{\mathbf{x}|\mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geqslant \mathbf{0}\}$ for vary $\mathbf{b}$. So, for a general two-dimensional linear programming problem, the set of circuits can be obtained easily.

## 2.2 A Steepest-Ascent Circuit in 2D

Consider the following general two-dimensional linear programming problem:

$$\begin{aligned} \max \quad & z = c_1x_1 + c_2x_2 \\ \text{s.t.} \quad & a_{11}x_1 + a_{12}x_2 \leqslant b_1, \\ & a_{21}x_1 + a_{22}x_2 \leqslant b_2, \\ & \quad \vdots \\ & a_{m1}x_1 + a_{m2}x_2 \leqslant b_m, \\ & x_1, \qquad x_2 \geqslant 0, \end{aligned} \tag{2.2}$$

where $a_{j1}$ and $a_{j2}$ are not all simultaneously equal to zero for all $j = 1, 2, \ldots, m$.

Since we focus on a maximization problem, a potential circuit should be the steepest-ascent circuit. In this subsection, we describe the acquisition of the set of the potential circuits.

Let $G$ be the set of vectors consisting of

$$\delta_i = \begin{bmatrix} -a_{i2} \\ a_{i1} \end{bmatrix} \text{ and } -\delta_i \text{ for } i = 1, ..., m.$$

Therefore, $G$ is the set of all circuits of problem (2.2). Since the potential circuit $\mathbf{g} \in G$ that can improve the objective value must satisfy $\mathbf{c}^T\mathbf{g} > 0$, if for each $i = 1, 2, \ldots, m$, we define

$$\mathbf{g}_i = \begin{cases} \delta_i, & \mathbf{c}^T\delta_i > 0 \\ -\delta_i, & \mathbf{c}^T\delta_i < 0, \end{cases} \tag{2.3}$$

then the set of all potential circuits is given by

$$G^+ = \{\mathbf{g}_i | i = 1, 2, 3, \ldots, m\}. \tag{2.4}$$

Without solving the specific problem (2.1), the steepest-ascent circuit of problem (2.2) can be easy computed by

$$\mathbf{g}_r = \text{argmax}\left\{\frac{\mathbf{c}^T\mathbf{g}_i}{\|\mathbf{c}\|\|\mathbf{g}_i\|} | \mathbf{g}_i \in G^+\right\}. \tag{2.5}$$

The steepest-ascent circuit as Eq. (2.5) will be used to update a solution in the proposed algorithm.

## 3. The Circuit Direction Search Algorithm (CDSA)

Consider the following special non-degenerate two-dimensional linear programming problem:

$$\begin{aligned} \max \quad & z = c_1x_1 + c_2x_2 \\ \text{s.t.} \quad & a_{11}x_1 + a_{12}x_2 \leqslant b_1, \\ & a_{21}x_1 + a_{22}x_2 \leqslant b_2, \\ & \qquad \vdots \\ & a_{m1}x_1 + a_{m2}x_2 \leqslant b_m, \\ & x_1, \qquad x_2 \geqslant 0, \end{aligned} \tag{3.1}$$

where $c_1, c_2 > 0$ and $b_j > 0$ for all $j \in \{1, 2, \ldots, m\}$.

Since a new interior search technique to solve a two-dimensional linear programming problem is proposed, for any feasible solution $\mathbf{x}_k$, a direction, a step size, and a criterion to stop the algorithm are required.

In this paper, the steepest-ascent circuit is used as a direction in which $\mathbf{g}_r$ of problem (3.1) is as follows:

$$\mathbf{g}_r = \text{argmax}\left\{\frac{\mathbf{c}^T\mathbf{g}_i}{\|\mathbf{c}\|\|\mathbf{g}_i\|} | \mathbf{g}_i \in G^+\right\}, \tag{3.2}$$

and a step size can be calculated by letting

$$\alpha_i = \frac{b_i - A_{i:}\mathbf{x}_k}{A_{i:}\mathbf{g}_r}, \tag{3.3}$$

where $A_{i:}$ is a coefficient vector of constraint $i$ for all $i \in \{1, ..., m+2\}$ and $i \neq r$. Then,

$$\alpha_p = \min\{\alpha_i | \alpha_i > 0, i \in \{1, ..., m+2\}\backslash\{r\}\}. \tag{3.4}$$

So, the updated solution is

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_p\mathbf{g}_r. \tag{3.5}$$

Additionally, the important step of the interior search technique is the stopping criterion. In the proposed algorithm, we consider the dual solution to stop the algorithm.

Consider the following dual problem associated with the problem (3.1):

$$\begin{aligned} \min \quad & z = b_1w_1 + b_2w_2 + \cdots + b_mw_m \\ \text{s.t.} \quad & a_{11}w_1 + a_{21}w_2 + \cdots + a_{m1}w_m \geqslant c_1, \\ & a_{12}w_1 + a_{22}w_2 + \cdots + a_{m2}w_m \geqslant c_2, \\ & w_1, w_2, \ldots, w_m \geqslant 0. \end{aligned} \tag{3.6}$$

To check the optimality of $\mathbf{x}_{k+1}$, we will use the complementary slackness and the KKT conditions to verify it. Theoretically, $\mathbf{x}_{k+1}$ will be the optimal solution to

the problem (3.1) when it is an extreme point that binds at least two constraints; assumed that $l$ and $p$. By complementary slackness, we can set $w_i = 0$ for $i = \{1, ..., m\} \setminus \{l, p\}$ and compute $w_l, w_p$ from $\begin{bmatrix} A_{l:}^T & A_{p:}^T \end{bmatrix}^{-1} \mathbf{c}$, where $A_{l:}$ and $A_{p:}$ are the gradient vectors of constraints $l$ and $p$. If $w_l, w_p \geqslant 0$, then $\mathbf{w} \geqslant \mathbf{0}$ is the dual feasible solution. Therefore, $\mathbf{x}_{k+1}$ is the optimal solution to the problem (3.1), and the algorithm is terminated.

Thus, the steps of the circuit direction algorithm for solving problem (3.1) are summarized as Algorithm 1, and it can be illustrated as the following example 3.1:

**Example 3.1.** Consider the following non-degenerate special two-dimensional linear programming problem:

$$\begin{array}{lrcll} \max & 6x_1 + 4x_2 & & & \\ \text{s.t.} & -x_1 + x_2 & \leqslant & 3, & \\ & x_1 + 5x_2 & \leqslant & 13, & \\ & 4x_1 + 7x_2 & \leqslant & 28, & \\ & x_1 & \leqslant & 10, & \\ & 6x_1 - x_2 & \leqslant & 30, & (3.7) \\ & 2x_1 - x_2 & \leqslant & 4, & \\ & x_1 - x_2 & \leqslant & 1, & \\ & 4x_1 - x_2 & \leqslant & 8, & \\ & 3x_1 - 5x_2 & \leqslant & 2, & \\ & x_1, x_2 & \geqslant & 0. & \end{array}$$

Since $\mathbf{x}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ is a feasible point, we will choose it as the initial point. From Eqs. (2.3)-(2.4), the set of all potential circuits $G^+$ is in Table 1:

**Table 1.** All potential circuits $\mathbf{g}_i$ for $i = \{1, ..., m\}$.

| $i$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| $\mathbf{g}_i$ | $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ | $\begin{bmatrix} 5 \\ -1 \end{bmatrix}$ | $\begin{bmatrix} 7 \\ -4 \end{bmatrix}$ | $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ | $\begin{bmatrix} 1 \\ 6 \end{bmatrix}$ | $\begin{bmatrix} 1 \\ 2 \end{bmatrix}$ | $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ | $\begin{bmatrix} 1 \\ 4 \end{bmatrix}$ | $\begin{bmatrix} 5 \\ 3 \end{bmatrix}$ |

Since

$$r = \text{argmax}\left\{ \frac{\mathbf{c}^T \mathbf{g}_i}{\|\mathbf{c}\| \|\mathbf{g}_i\|} | \mathbf{g}_i \in G^+ \right\} = 9,$$

and

$$p = \text{argmin}\{\alpha_i | \alpha_i > 0, i \in \{1, ..., 8\}\} = 8,$$

where $\alpha_i = \frac{b_i - A_{i:}\mathbf{x}_k}{A_{i:}\mathbf{g}_r}$, we get the initial circuit is $\mathbf{g}_9$ and $\alpha_8 = \frac{8}{17}$.

So, the solution $\mathbf{x}_k$, the index of circuit direction $r$, the index of step size $p$ and the associated dual solution $\mathbf{w}$ for each iteration are presented as follows (Table 2):

**Table 2.** Summary of the computation in each iteration.

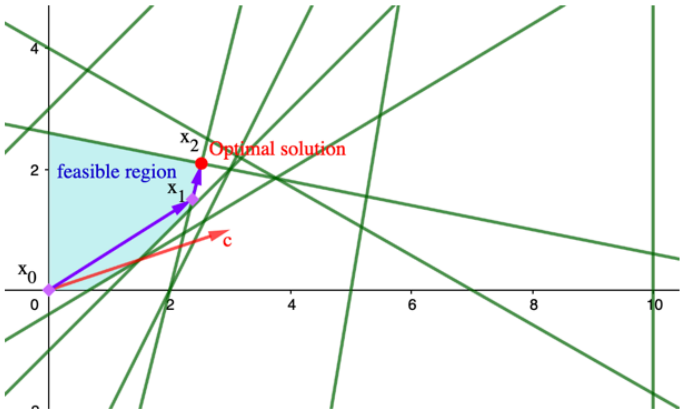
| $k$ | $\mathbf{x}_k$ | $r$ | $p$ | No. of binding constraints | $\mathbf{w}$ |
|---|---|---|---|---|---|
| 0 | $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$ | - | - | 0 | - |
| 1 | $\begin{bmatrix} 2.35 \\ 1.14 \end{bmatrix}$ | 9 | 8 | 1 | - |
| 2 | $\begin{bmatrix} 2.52 \\ 2.09 \end{bmatrix}$ | 8 | 2 | 2 | $\begin{bmatrix} 1.23 \\ 1.05 \end{bmatrix}$ |

Since $\mathbf{w} \geqslant \mathbf{0}$, the optimal solution is obtained, say $\mathbf{x}_2 = \begin{bmatrix} 2.52 \\ 2.09 \end{bmatrix}$, with $z^* = 9.67$.

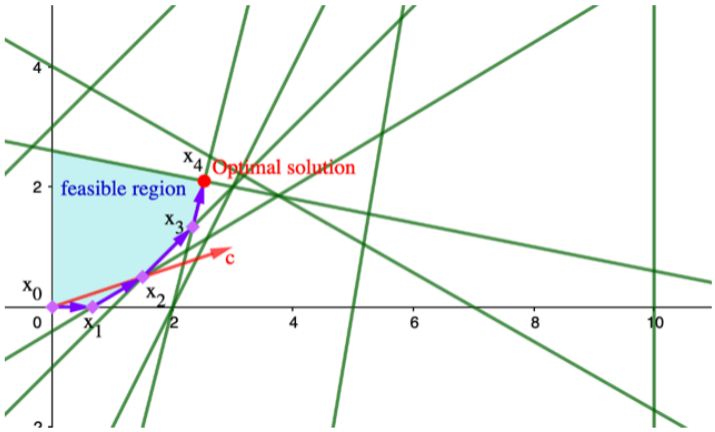
Fig. 2 shows the graph of this example solved by the CSDA. For solving the above example by the slope algorithm and the simplex method, they require 3 and 4 iterations to update a solution to gain the optimal solution while the circuit direction algorithm does it in only two iterations.

## 4. The Computational Results

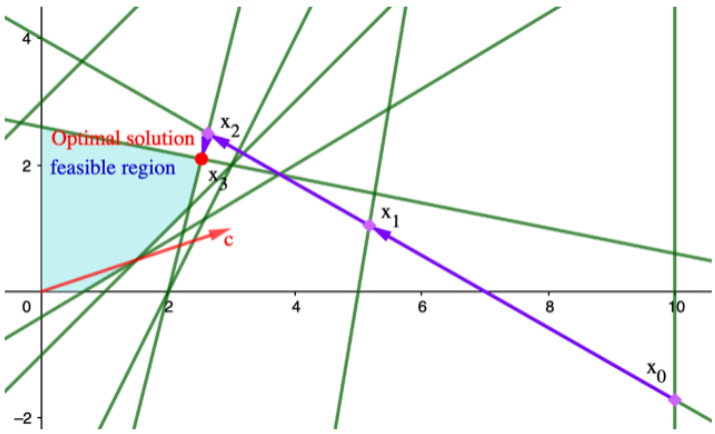
In this section, we present the comparisons of the average number of iterations and the running time of the circuit direction search algorithm (CDSA), the slope algorithm (SA), the simplex method (SM), and the interior point method (IPM). Then, we




The movement of the circuit direction search algorithm to solve the example




The movement of the simplex method to solve the example




The movement of the slope algorithm to solve the example

**Fig. 2.** Graph of the Example 3.1.

**Table 3.** The algorithm of circuit direction search algorithm.

**Algorithm 1:** The circuit direction search algorithm for solving the problem (3.1).

**begin**
  Set $k = 0$ and $\alpha_p = 0$;
  $\mathbf{x}_k = (0, 0)$;
  Find the set of all potential circuits $G^+ = \{\mathbf{g}_i \in G | \mathbf{c}^T \mathbf{g}_i > 0,\ i = 1, ..., m\}$;
  **if** $\mathbf{g}_i \not\geqslant \mathbf{0}$ for all $\mathbf{g}_i \in G^+$ **then**
    $\mathbf{g}_r = \mathbf{c}$;
    $p = \text{argmin}\left\{\alpha_i = \frac{b_i - A_{i:}\mathbf{x}_k}{A_{i:}\mathbf{g}_r} \geqslant 0 \mid i \in \{1, ..., m+2\} \setminus \{r\}\right\}$;
  **else**
    **do**
      Choose a circuit $\mathbf{g}_r$ where $r = \text{argmax}\left\{\frac{\mathbf{c}^T \mathbf{g}_i}{||\mathbf{c}||||\mathbf{g}_i||} \mid \mathbf{g}_i \in G^+\right\}$;
      $G^+ = G^+ \setminus \{\mathbf{g}_r\}$;
      $p = \text{argmin}\left\{\alpha_i = \frac{b_i - A_{i:}\mathbf{x}_k}{A_{i:}\mathbf{g}_r} \geqslant 0 \mid i \in \{1, ..., m+2\} \setminus \{r\}\right\}$;
    **while** $(\alpha_p = 0)$;
  $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_p \mathbf{g}_r$;
  $k \leftarrow k + 1$;
  **for** $i = 1, ..., m$;
    **if** $A_{i:}\mathbf{x}_{k+1} = b_i$ and $i \neq r$ **then**
      $l \leftarrow i$;
      $i = i + 1$;
     **else** $\mathbf{g}_r = \mathbf{g}_p$;
  Construct a sub-matrix $A' = \begin{bmatrix} A_{l:} \\ A_{p:} \end{bmatrix}^T$;
  Compute $\mathbf{w} = (A')^{-1}\mathbf{c}$;
  **if** $\mathbf{w} \geqslant \mathbf{0}$ **then return** $\mathbf{x}^* = \mathbf{x}_{k+1},\ l,\ p$;
  **else** $\mathbf{g}_r = \mathbf{g}_l$;
**end**

implemented CDSA, SA, SM, and IPM using Python code on Colaboratory, a product from Google Research. To test the efficiency of the proposed approach, we randomly generated special nondegenerate two-dimensional linear programming problems for the various number of constraints which are 50, 100, 200, 300, 400, 500, 600, 700, 800, 900, and 1,000. Thus, the results are presented as Table 4, and Figs. 3-4 show the graphs of the average number of iterations and the running time of the randomly generated nondegenerate problems.

In Table 4, we can see that the simplex method has the minimum average number of iterations for the problem sizes 50 and 100, while CDSA has the second minimum average of iterations. However, CDSA has the minimum average number of running for these sizes. For other problem sizes, the CDSA has the minimum average number of both iterations and the running time. We can see that the circuit direction search algorithm can reduce the num-

**Table 4.** The results of the randomly nondegenerate tested problems.

| | The average number of iterations | | | | The average number of the running time | | | |
|---|---|---|---|---|---|---|---|---|
| Sizes | IPM | CDSA | SA | SM | IPM | CDSA | SA | SM |
| 50 | 5.95 | 2.10 | 4.30 | **1.70** | 0.023 | **0.003** | 0.004 | 0.011 |
| 100 | 5.40 | 2.10 | 6.25 | **1.75** | 0.042 | **0.004** | 0.005 | 0.033 |
| 200 | 6.25 | **2.05** | 8.15 | 2.15 | 0.078 | **0.006** | 0.008 | 0.145 |
| 300 | 5.95 | **2.10** | 7.65 | 2.25 | 0.141 | **0.010** | 0.014 | 0.287 |
| 400 | 6.10 | **2.10** | 8.65 | 2.45 | 0.237 | **0.013** | 0.017 | 0.579 |
| 500 | 6.50 | **2.00** | 9.55 | 2.25 | 0.393 | **0.014** | 0.016 | 0.772 |
| 600 | 6.10 | **2.05** | 10.80 | 2.25 | 0.554 | **0.018** | 0.041 | 1.291 |
| 700 | 6.15 | **2.00** | 9.55 | 2.35 | 0.767 | **0.020** | 0.026 | 1.753 |
| 800 | 6.05 | **2.00** | 9.90 | 2.25 | 1.065 | **0.023** | 0.031 | 2.276 |
| 900 | 6.25 | **2.00** | 11.20 | 2.20 | 1.403 | **0.025** | 0.030 | 2.837 |
| 1000 | 6.30 | **2.05** | 10.40 | 2.30 | 1.789 | **0.028** | 0.029 | 3.454 |
| **Total** | 67.00 | **22.55** | 96.40 | 23.90 | 6.493 | **0.192** | 0.220 | 13.438 |

**Note:** The minimum averages number of iterations and running time for each size are bolded.

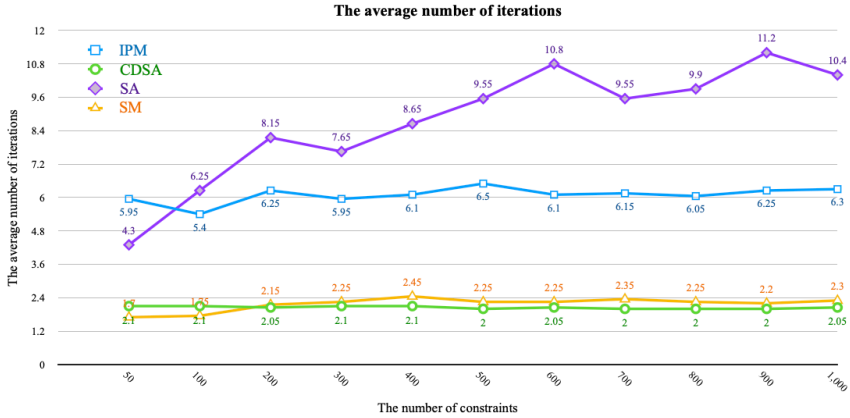



**Fig. 3.** The average number of iterations of the randomly tested problems.

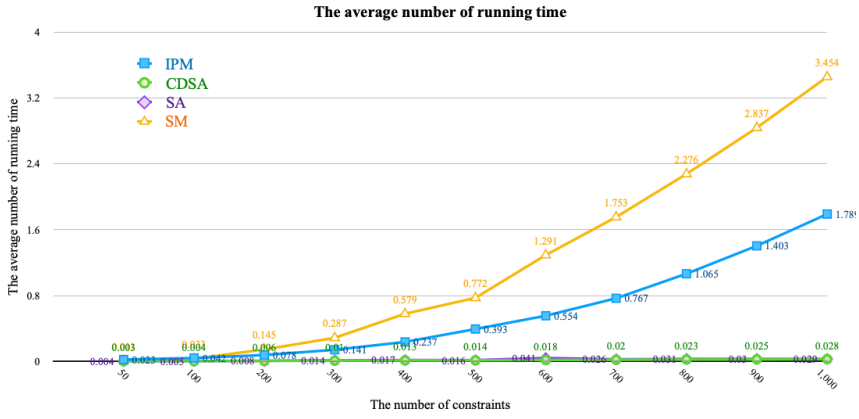



**Fig. 4.** The average number of the running time of the randomly tested problems.

ber of iterations and the running time compared with the slope algorithm, the simplex method, and the interior point method for the randomly special nondegenerate generated two-dimensional linear programming problems. If the problem has many redundant constraints, the CDSA performs quite well, while the slope algorithm performs more iterations to reach the optimal solution. Moreover, the CDSA can reduce the number of iterations since it starts at the origin point and crosses through the feasible region along with the appropriate circuit to a feasible point while the simplex method moves to a feasible adjacent point. It is the reason that the CDSA can reduce the number of iterations.

## 5. Conclusions

In this paper, a new algorithm called the *circuit direction search algorithm* (CDSA), which is the interior search technique to solve the special nondegenerate two-dimensional linear programming problem, is proposed. It uses the suitable circuit of the problem as a direction for updating a solution. Then, the complementary slackness and the KKT conditions are used to check the optimality for terminating the algorithm. The results show that the proposed algorithm could reduce the average number of iterations and the running time compared with the interior point, slope, and simplex methods.

Although the proposed algorithm can solve only a two-dimensional linear programming problem, it can be used to exchange two entering and leaving variables for a general nondegenerate linear programming problem solved by the simplex method. Consequently, in future work, we will apply the CDSA to identify double pivots at each iteration for the simplex method to solve a general nondegenerate linear programming problem. Moreover, we will improve this algorithm by finding the appropriate direction to update a solution to a degenerate two-dimensional linear programming problem. Finally, it will be used to identify double pivots for a general linear programming problem solved by the simplex method.

## References

[1] Hung Y.F, Leachman R.C. A production planning methodology for semiconductor manufacturing based on iterative simulation and linear programming calculations. IEEE Transactions on Semiconductor Manufacturing. 1996;9(2):257-69.

[2] Stefan V, David L.W. Introduction to computational optimization models for production planning in a supply chain. Vol.240 Springer Science & Business Media; 2006.

[3] Missbauer H, Reha Uz. Optimization models of production planning problems. Planning production and inventories in the extended enterprise. Springer. 2011; 437-507.

[4] Flood M.M., The traveling-salesman problem, Operations research. 1956:4(1);61-75.

[5] Reinelt G. TSPLIB-A traveling salesman problem library. ORSA journal on computing. 1991:3(4);376-84.

[6] Kuhn H.W. The Hungarian method for the assignment problem. Naval research logistics quarterly. 1955:2(1-2);83-97.

[7] Pardalos P.M, Leonidas S.P. The quadratic assignment problem. Handbook of combinatorial optimization 4; 1998. p.241.

[8] Ford Jr L.R, Fulkerson D.R. Solving the transportation problem. Management Science. 1956:3(1);24-32.

[9] Appa G.M. The transportation problem and its variants. Journal of the Operational Research Society. 1973:24(1);79-99.

[10] Dantzig G.B. Linear programming and extensions. RAND Corporation: Santa Monica; 1963.

[11] Klee V, Minty G. How good is the simplex algorithm. In Equalities. Academic Press: New York; 1972.

[12] Shamos M, Dan Hoey. Geometric intersection problems. In: 17th Annual Symposium on Foundations of Computer Science; 1976. p.208-15.

[13] Megiddo N. Linear-time algorithms for linear programming in R3 and related problems. SIAM Journal on Computing. 1983:12(4);759-76.

[14] Dyer M. Linear time algorithms for two- and three-variable linear programs, SIAM Journal on Computing. 1984:13(1);31-45.

[15] Vitor F, Easton T. The double pivot simplex method. Mathematical Methods of Operations Research. 2018:87;109-37.

[16] Jamrunroj P, Boonperm A. A New technique for solving a 2-dimensional linear program by considering the coefficient of constraints. Advances in Intelligent Systems and Computing. 2021:1324;276-86.

[17] Karmarkar N. A new polynomial-time algorithm for linear programming. Combinatorica. 1984:(4);373-95.

[18] Yamashita H. A polynomially and quadratically convergent method for linear programming. Mathematical Systems Institute; 1986.

[19] Anstreicher K.M. A standard form variant and safeguarded line search for the modified Karmarkar algorithm. Mathematical Programming. 1990:47;337-51.

[20] Anstreicher K. A monotonic projective algorithm for fractional linear programming. Algorithmica. 2005:1;483-98.

[21] Ye Y. A class of projective transformations for linear programming. SIAM Journal on Computing. 1990:19;457-66.

[22] Ye Y. Recovering optimal basic variables in Karmarkar's polynomial algorithm for linear programming. Mathematics of Operations Research. 1990:15(3):564-72.

[23] den Hertog D, Roos C. A survey of search directions in interior point methods for linear programming. Mathematical Programming. 1991:52;481-509.

[24] Visuthirattanamanee R, Sinapiromsaran K, Boonperm A. Self-Regulating Artificial-Free Linear Programming Solver Using a Jump and Simplex Method. Mathematics. 2020:8(3);356-71.

[25] Rockafellar R.T. The elementary vectors of a subspace of RN. In Combinatorial Mathematics and its Applications, University of North Carolina Press; 1969. p.104-27.

[26] De Loera J.A, Hemmecke R, Lee J. On augmentation algorithms for linear and integer-linear programming: from Edmonds-Karp to Bland and beyond. SIAM Journal on Optimization. 2015:25(4);2494-511.

[27] Borgwardt S, Finhold E, Hemmecke R. On the circuit diameter of dual transportation polyhedra. SIAM Journal on Discrete Mathematics. 2016:29(1);113-21.

[28] Borgwardt S, Finhold E, Hemmecke R. Quadratic diameter bounds for dual network flow polyhedra. Mathematical Programming (1-2), Serie A. 2016:159;237-51.

[29] Graver J.E. On the foundation of linear and integer programming I. Mathematical Programming. 1975:9;207-26.