P-ISSN 2586-9000
E-ISSN 2586-9027
Homepage : *https://tci-thaijo.org/index.php/SciTechAsia* Science & Technology Asia
Vol.29 No.2 April - June 2024 Page: [122-137]
Original research article

The Slope-Circuit Hybrid Method for Solving Degenerate Two-Dimensional Linear Programs

Panthira Jamrunroj, Aua-aree Boonperm*

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

Received 15 January 2024; Received in revised form 29 February 2024
Accepted 1 April 2024; Available online 25 June 2024

ABSTRACT

Traditional linear programming (LP) methods, like the simplex algorithm, often struggle with the efficiency of solving degenerate LP problems. This study introduces the slope-circuit hybrid method, an innovative interior search technique designed to overcome these challenges by strategically combining slope-based analysis and circuit direction search. This method accelerates convergence, yielding optimal solutions. Focusing on degenerate constraints, the algorithm intelligently selects an initial circuit direction using slope information. The circuit direction search adeptly navigates the next direction to improve a solution, resulting in a significant reduction in iterations. Rigorous termination at an optimal solution is guaranteed through the computation of associated dual variables. Empirical testing on degenerate 2D linear programs supports substantial performance enhancements over simplex, interior point, and slope algorithms, evident in reduced iterations and improved running time. The slope-circuit hybrid method emerges as a promising solution for optimizing resource allocation in industrial settings, especially those constrained by limited computational resources. Its potential extends to streamlining decision-making processes and enhancing efficiency across various real-world applications.

Keywords: Circuit direction; Degenerate linear programming problem; Interior search technique; Simplex algorithm

1. Introduction

Linear programming (LP) has a rich history dating back to World War II. Originally devised to minimize army expenses and maximize losses for adversaries, it has evolved into an optimization technique.

This method seeks the optimal outcome by maximizing or minimizing a linear objective function within defined equality or inequality constraints. In real-world applications, linear programming is widely utilized to address diverse industrial challenges, including the traveling salesman problem, production planning problem, and assignment problem, with the aim of achieving optimal results.

In 1947, Dantzig [1] introduced the simplex method, an efficient iterative approach for solving linear programming problems by searching for an optimal point within a feasible region. However, Klee and Minty [2] demonstrated the worst-case computational time of the simplex method using the Klee-Minty cube. In this scenario, the method traverses all vertices of the cube, resulting in an exponential number of iterations. Despite its practical utility, this worst-case complexity has led researchers to propose algorithms aimed at reducing the computational complexity of the simplex method. Significant advancements have been made, particularly in two-dimensional linear programming (2DLP).

In 1976, Shamos and Hoey [3] proposed an algorithm with a time complexity of $O(n \log n)$, vastly superior to the simplex method. Megiddo [4] further improved upon this, achieving linear-time complexity for 2DLP in 1983. Later, Dyer [5] also presented a novel algorithm with $O(n \log n)$ complexity.

Focusing on the simplex method, in 2014, Boonperm and Sinapiromsaran [6] introduced an innovative enhancement leveraging Megiddo's mapping technique, eliminating the need for artificial variables and improving its efficiency.

In 2018, Vitor and Easton [7] introduced the slope algorithm for solving a 2DLP, particularly applicable when the objective function's coefficient values are positive, and right-hand side values are non-negative. The algorithm, integrated with the simplex method in the double pivot simplex method, identifies two variables for updating a solution, resulting in reduced iterations compared to the simplex method. However, the slope algorithm may start with an infeasible point and take longer to identify two leaving variables in each iteration. Additionally, it is not effective when returning only one variable, akin to the simplex method.

Addressing the limitations of the slope algorithm, recent advancements have been proposed by Jamrunroj and Boonperm [8] in 2021. This novel approach relies solely on constraint coefficients to indicate the optimal solution for a special 2DLP. Despite this improvement, the slope algorithm continues to be utilized in cases involving double pivots. However, the slope algorithm can exhibit a weakness when dealing with problems featuring numerous redundant constraints. These constraints can cause the algorithm to get stuck in regions far from the optimal solution, leading to increased iterations for finding it. To overcome this limitation, the interior search technique can be employed. By ensuring the algorithm remains within the feasible region, the interior search technique can potentially reduce the number of iterations needed to reach the optimal solution.

The interior search technique, also known as the interior point method (IPM) or Karmarkar's algorithm, was initially proposed by Karmarkar [9] in 1984. It utilizes a direction of steepest descent in the projective transformed space to update a solution in each iteration. Despite its quick approach to a neighborhood of the optimal solution, Karmarkar's algorithm tends to be slow in reaching the optimal solution itself.

The interior point method has attracted considerable attention, with significant contributions from researchers such as Anstreicher [10] and Ye [11,12]. The exploration and consolidation of various search direction types have been documented [13], showcasing advancements in algorithmic techniques. A notable recent development is a novel interior search method introduced by Visuthirattanamanee et al. [14] in 2020. This innovative approach leverages objective and binding constraint gradients to enhance solution outcomes.

An emerging concept within IPM is the circuit, introduced by Rockafellar [15] in 1969 as a set of elementary vectors in a subspace. In 1975, Graver [16] expanded on circuit directions, providing an optimality certificate for any linear program. Algorithms generating circuit walks, enhancing paths iteratively along circuits, find applications in various mathematical programming scenarios [17–20]. Ongoing research, exemplified by the introduction of greedy circuit directions by Hemmecke et al. [21] in 2011 and the steepest-descent circuit proposed by Loera et al. [22] in 2015, indicates the potential for accelerated convergence in solving linear programming problems.

Recognizing the computational expense associated with obtaining the steepest-descent circuit, alternative approaches have been investigated. In 2018, Borgwardt et al. [23] suggested simpler strategies, such as utilizing edge directions of constraints for specific cases, offering practical solutions. This collective progress underscores a promising trajectory in optimizing the Interior Point Method for efficient linear programming problem-solving.

Building upon these advancements, the Circuit Direction Search Algorithm (CDSA) was introduced by Jamrunroj and Boonperm [24] in 2023 specifically for solving two-dimensional linear programming problems (2DLP). This algorithm exhibits effectiveness when the problem satisfies specific conditions, including positive coefficients for the objective function and right-hand side values. While CDSA demonstrates promising results, a limitation exists in its handling of degenerate cases. Its reliance on the steepest-ascent circuit as the initial direction can lead to longer solution times in these scenarios, hindering its broader applicability.

These limitations motivate our research, which aims to address the issue of slow convergence in degenerate cases for CDSA by introducing a novel criterion for initial circuit selection. This research aims to address this limitation by introducing a novel criterion that leverages the slope of degenerate constraints to enhance CDSA's performance. We propose a modification to CDSA that incorporates this criterion to guide initial circuit selection, aiming to accelerate convergence in degenerate scenarios. The study assesses the efficiency of the modified algorithm through comparisons with the original CDSA on randomly generated linear programming problems. Our research contributes to the development of more robust and widely applicable linear programming methods, particularly in domains where degeneracy is prevalent.

This paper is organized as follows: Section 2 introduces the fundamental concept of a circuit, laying the groundwork for the subsequent discussions. In Section 3, we present our proposed algorithm, emphasizing its novel features and addressing degenerate cases in two-dimensional linear programming. Section 4 showcases computational results, offering insights into the algorithm's performance through the average number of iterations and running time. Finally, the results and their implications are discussed in the concluding section, providing a comprehensive summary and potential avenues for future research.

2. Preliminaries

Given that the proposed algorithm shares a theoretical basis with the circuit direction search algorithm, the definitions and identifications of circuit directions closely align with those introduced in [23]. This section begins by presenting the definition of a circuit, followed by the identification of circuit directions.

2.1 Definition of Circuit

Consider the polyhedron P defined by:

$$P = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{B}\mathbf{x} \leqslant \mathbf{d}\}, \qquad (2.1)$$

where $\mathbf{A} \in \mathbb{R}^{m_A \times n}$, $\mathbf{B} \in \mathbb{R}^{m_B \times n}$, $\mathbf{b} \in \mathbb{R}^{m_A}$, $\mathbf{d} \in \mathbb{R}^{m_B}$ and $rank \begin{pmatrix} \mathbf{A} \\ \mathbf{B} \end{pmatrix} = n$. The set of circuits $C(\mathbf{A},\mathbf{B})$ of P is defined as follows.

Definition 2.1 ([23])**.** The set of circuits $C(\mathbf{A},\mathbf{B})$ of P defined in (2.1) 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 $\{\mathbf{Bx} | \mathbf{x} \in ker(\mathbf{A}) \setminus \{\mathbf{0}\}\}$.

According to Graver [16], the set of circuits is made up of all potential edge directions for P, as long as the right-hand-side vectors $\mathbf{b}$ and $\mathbf{d}$ vary. When working with circuits, any normalization that yields a distinct positive and negative representation for each of these one-dimensional directions can be employed. For geometrical considerations, every positive scalar multiple of a circuit $\mathbf{g} \in C(\mathbf{A},\mathbf{B})$, where $C(\mathbf{A},\mathbf{B})$ comprises all possible edge directions of P, is referred to as a *circuit direction of* P. The two-dimensional circuits for each constraint are shown in Fig. 1.

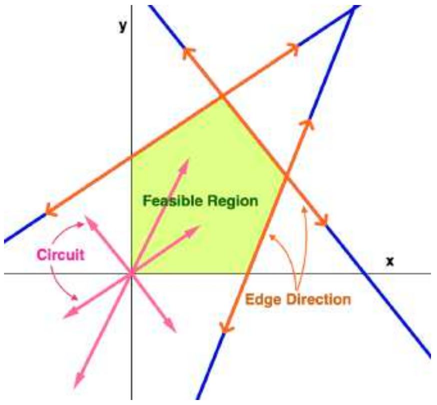

Fig. 1. Edge directions and circuits of each constraint in 2D.

Additionally, augmentation schemes for linear programming problems are developed using circuits. In these schemes, successively better, maximum steps are performed along circuit directions until an optimal solution is obtained or the problem is unbounded. Subsequently, De Loera et al. [22] proposed an augmentation scheme that updates the solution at each iteration by utilizing a steepest-descent circuit to solve the standard linear programming problem. According to the authors, a *strictly feasible circuit* that enhances a search direction at a feasible solution is referred to as the steepest-descent circuit. Therefore, the definition of a strictly feasible circuit is described first followed by the steepest-descent circuit definition.

Definition 2.2 (Strictly feasible direction)**.** Consider the following standard linear programming problem (SLP):

$$\begin{aligned} \min \quad & \mathbf{c}^T\mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} = \mathbf{b}, \\ & \mathbf{x} \geqslant \mathbf{0}, \end{aligned} \qquad (2.2)$$

where $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{x}, \mathbf{c} \in \mathbb{R}^n$, and $\mathbf{b} \in \mathbb{R}^m$. Let $\mathbf{x}_0$ be a feasible solution to SLP. A direction $\mathbf{d}$ is said to be strictly feasible at $\mathbf{x}_0$

if $\mathbf{x}_0 + \alpha\mathbf{d} \in \{\mathbf{x}|\mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geqslant \mathbf{0}\}$ for some $\alpha > 0$.

Definition 2.3 (Steepest-descent circuit). Let $\mathbf{x}_0$ be a feasible solution to SLP. 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.

The updated solution $\mathbf{x}_{k+1}$ can be the optimal solution when it is an extreme point that binds at least two linearly independent constraints. To verify its optimality, the Karush-Kuhn-Tucker conditions and complementary slackness are applied. Assume that $\mathbf{x}_{k+1}$ binds at constraints l and p.

Nevertheless, finding a steepest-descent circuit for SLP proves challenging and is obtained through the linear programming problem detailed in [24]. Borgwardt et al. [23] affirm that the set of circuits for a given variation of **b** consists precisely of all edge directions of the polyhedron $\{\mathbf{x}|\mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geqslant \mathbf{0}\}$ for vary **b**. Consequently, deriving the set of circuits for a general two-dimensional linear programming problem is straightforward. However, our focus in this paper is on the maximization problem, necessitating the identification of the steepest-ascent circuit.

2.2 A Steepest-Ascent Circuit

Consider the following general two-dimensional linear programming problem:

$$\begin{array}{lrl} \max & z = c_1x_1 + c_2x_2 & \\ \text{s.t.} & a_{11}x_1 + a_{12}x_2 \leqslant b_1, & \\ & a_{21}x_1 + a_{22}x_2 \leqslant b_2, & \\ & \vdots & \\ & a_{m1}x_1 + a_{m2}x_2 \leqslant b_m, & \\ & x_1, x_2 \geqslant 0, & \end{array} \tag{2.3}$$

where $c_1, c_2, b_i \in \mathbb{R}$ for $i = 1, 2, \ldots, m$, and $a_{i1}, a_{i2} \in \mathbb{R}$ are not both zero simultaneously for all $i = 1, 2, \ldots, m$. Then, the acquisition of the set of the potential circuits is described as follows.

Let G be the non-zero vectors set containing $\delta_i = \begin{bmatrix} -a_{i2} \\ a_{i1} \end{bmatrix}$ and $-\delta_i$ for $i = 1, ..., m$. Thus, G is the set of all circuits for the problem (2.3). Since, in this study, we are focusing on the maximization problem, the potential circuits $\mathbf{g} \in G$ that improve the objective value z must satisfy $\mathbf{c}^T\mathbf{g} > 0$, For $i = 1, 2, \ldots, m$, if we define

$$\mathbf{g}_i = \begin{cases} \delta_i, & \mathbf{c}^T\delta_i > 0 \\ -\delta_i, & \mathbf{c}^T\delta_i < 0, \end{cases} \tag{2.4}$$

then the potential circuits set is defined by

$$G^+ = \{\mathbf{g}_i | i = 1, 2, \ldots, m\}. \tag{2.5}$$

The steepest-ascent circuit $\mathbf{g}_r$ for the problem (2.3) can be found, where r is computed as follows:

$$r = \operatorname{argmax}\left\{\frac{\mathbf{c}^T\mathbf{g}_i}{\|\mathbf{c}\|\|\mathbf{g}_i\|} \mid \mathbf{g}_i \in G^+\right\}. \tag{2.6}$$

In the interior search technique, an initial direction is typically required. The steepest-ascent circuit can be employed as an initial circuit direction in certain cases.

3. The Proposed Method

In this section, we present a new criterion for determining the initial circuit direction in the degenerate case of a special 2DLP. We apply this criterion in conjunction with the CDSA in [24] to address the increased time required for finding an optimal solution in solving a two-dimensional linear programming problem. Subsequently, we describe a new criterion for establishing the initial circuit direction in the degenerate case of a special 2DLP with the slope information, followed by the slope-circuit hybrid method.

3.1 The Circuit Direction Search Algorithm

Recall CDSA in [24], consider the following special non-degenerate two-dimensional linear programming problem:

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

where $c_1,\ c_2 > 0$, $b_i > 0$ for all $i \in \{1, 2, \ldots, m\}$ and $a_{i1}, a_{i2} \in \mathbb{R}$ are not both zero simultaneously for all $i = 1, 2, \ldots, m$. Since the potential circuits set is

$$G^+ = \{\mathbf{g}_i | i = 1, 2, \ldots, m\}, \tag{3.2}$$

any feasible solution $\mathbf{x}_k$ is updated with the steepest-ascent circuit $\mathbf{g}_r$ where

$$r = \text{argmax}\left\{\frac{\mathbf{c}^T\mathbf{g}_i}{\|\mathbf{c}\|\|\mathbf{g}_i\|} \mid \mathbf{g}_i \in G^+\right\}, \tag{3.3}$$

and a step size can be obtained by determining the following set:

$$Ap = \{\alpha_i > 0 \mid i \in \{1, 2, \ldots, m+2\} \setminus \{r\}\}, \tag{3.4}$$

where $\alpha_i = \frac{b_i - A_{i:}\mathbf{x}_k}{A_{i:}\mathbf{g}_r}$, $A_{i:}\mathbf{g}_r \neq 0$, and $A_{i:}$ is the coefficient vector of the constraint i for all $i \in \{1, ..., m+2\}$ and $i \neq r$. If $Ap = \emptyset$, meaning $\alpha_i \leqslant 0$ for all i, then the algorithm is terminated and the current solution $\mathbf{x}_k$ is the optimal solution. Otherwise, $p = \arg\min\{Ap\}$ and the current solution $\mathbf{x}_k$ is updated by

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

The updated solution $\mathbf{x}_{k+1}$ can be the optimal solution when it is an extreme point that binds at least two linearly independent constraints. To verify its optimality, the Karush-Kuhn-Tucker conditions and complementary slackness are applied. Assume that $\mathbf{x}_{k+1}$ binds at constraints l and p. By the complementary slackness, w_i can be set to zero for all $i \in \{1, ..., m+2\} \setminus \{l, p\}$, and w_l, w_p can be obtained by computing $\begin{bmatrix} A_{l:}^T & A_{p:}^T \end{bmatrix}^{-1} \mathbf{c}$, where $A_{l:}$ and $A_{p:}$ are the gradient vectors of constraints l and p. If $w_l, w_p \geqslant 0$, then $\mathbf{w} \geqslant \mathbf{0}$ becomes the dual feasible solution. Thus, $\mathbf{x}_{k+1}$ is the optimal solution to the problem (3.1).

Given that a potential circuit with the closest angle to the objective function can enhance a solution or achieve optimality, the criterion for choosing an initial circuit is presented first as Sub-Algorithm 1 (in Table 1), followed by the circuit direction search algorithm as Algorithm 1 (in Table 2).

Table 1. The criterion for choosing an initial circuit for a non-degenerate problem.

```
Sub-Algorithm 1: The criterion for choosing an initial circuit of non-degenerate
1:  begin
2:      Find the set of all potential circuits
        $G^+ = \{\mathbf{g}_i | i = 1, 2, \ldots, m\}$;
3:      If $\mathbf{g}_i \ngeqslant \mathbf{0}$ for all $\mathbf{g}_i \in G^+$ then
4:          return $\mathbf{g}_r = \mathbf{c}$;
5:      Else
6:          If $\mathbf{g}_i \geqslant \mathbf{0}$ for $\mathbf{g}_i \in G^+$ then
7:              $r = \text{argmax}\left\{\frac{\mathbf{c}^T\mathbf{g}_i}{\|\mathbf{c}\|\|\mathbf{g}_i\|} \mid \mathbf{g}_i \in G^+\right\}$;
8:              return $\mathbf{g}_r$;
9:  End
```

CDSA demonstrates effectiveness in addressing non-degenerate linear programming problems. However, its suitability diminishes in cases of degeneracy, where CDSA's initial circuit selection criterion may result in choosing an unsuitable direction, leading to solution updates that might converge to an infeasible point (Fig. 2). To overcome this challenge, we present a new criterion specifically designed for selecting

Table 2. The steps of the circuit direction search algorithm.

Algorithm 1: CDSA.

```
1:  begin
2:      Set $k, l, p = 0$, $\alpha_p = 0$ and $\mathbf{x}_k = (0, 0)$;
3:      Perform Sub-Algorithm 1;
4:      While $\mathbf{g}_r \neq \mathbf{0}$ do
5:          $Ap = \{\alpha_i > 0 \mid i \in \{1, 2, \dots, m\} \setminus \{r\}\}$;
            where $\alpha_i = \frac{b_i - A_{i:}\mathbf{x}_k}{A_{i:}\mathbf{g}_r}$, $A_{i:}\mathbf{g}_r \neq 0$;
6:          If $Ap = \emptyset$ then return $\mathbf{x}_k^* = \mathbf{x}_k$,$l, p$;
7:          Else
8:              $p = \arg\min\{Ap\}$;
9:              $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_p \mathbf{g}_r$;
10:             $k \leftarrow k + 1$;
11:             For $i = 1, \dots, m$;
12:                 If $\mathbf{A}_{i:}\mathbf{x}_{k+1} = b_i$ and $i \neq r$ then
13:                     $l \leftarrow i$;
14:                     $i \leftarrow i + 1$;
15:                 Else $\mathbf{g}_r = \mathbf{g}_p$;
16:             Construct a sub-matrix $A' = \begin{bmatrix} A_{l:} \\ A_{p:} \end{bmatrix}^T$;
17:             Compute $\mathbf{w} = (A')^{-1}\mathbf{c}$;
18:             If $\mathbf{w} \geqslant \mathbf{0}$ then return $\mathbf{x}_k^* = \mathbf{x}_k$,$l, p$;
19:             Else $\mathbf{g}_r = \mathbf{g}_l$;
20: End
```

initial circuits in degenerate scenarios. The details of this novel criterion are outlined in the subsequent subsection 3.2.

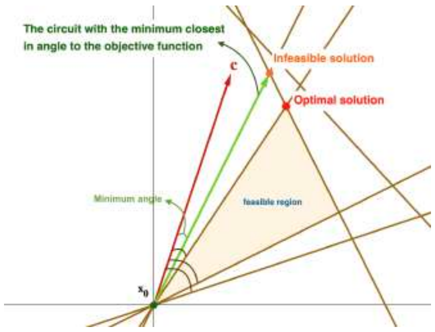
The circuit with the minimum closest in angle to the objective function
c
Infeasible solution
Optimal solution
Minimum angle
feasible region
x0

Fig. 2. An initial circuit chosen by CDSA in the degenerate case.

3.2 A Slope-Circuit Criterion for Selecting an Initial Circuit in Degenerate Case

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

$$
\begin{aligned}
\max \quad & z = c_1x_1 + c_2x_2 \\
\text{s.t.} \quad & a_{11}x_1 + a_{12}x_2 \leqslant b_1, \\
& a_{21}x_1 + a_{22}x_2 \leqslant b_2, \\
& \vdots \\
& a_{m1}x_1 + a_{m2}x_2 \leqslant b_m, \\
& x_1, x_2 \geqslant 0,
\end{aligned}
\tag{3.6}
$$

where $c_1, c_2 > 0$, $b_i \geqslant 0$ for all $i = 1, 2, \dots, m$ and $a_{i1}, a_{i2} \in \mathbb{R}$ are not both zero simultaneously for all $i = 1, 2, \dots, m$, and there exists at least one $b_i = 0$ for some i.

Due to the potential inadequacy of CDSA's initial circuit selection criterion, we have modified the process of choosing the initial direction for the problem (3.6). Please consult the following figure for further clarification.

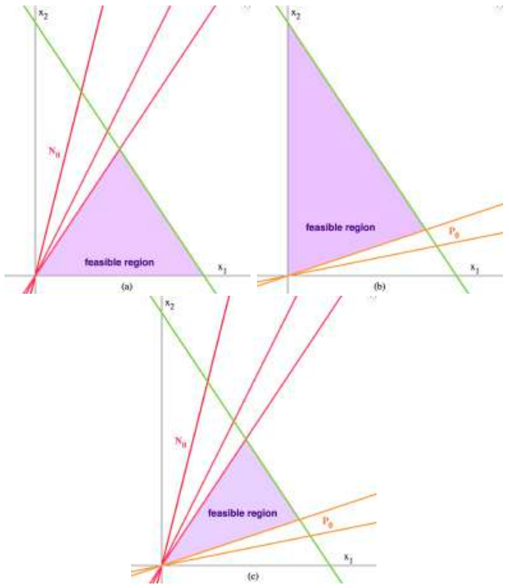

Fig. 3. Examples for degenerate cases of 2DLP.

Examining Fig. 3 reveals that one of the degenerate constraints aligns with the edge of the feasible region, suggesting potential improvement in the solution along this direction. Consequently, we will focus on the set of constraints where the right-hand side value is 0, and its slope will be taken into consideration.

Consider the set of all degeneracy constraints for the problem, defined as follows:

$$B_0 = \{i \in \{1, \ldots, m\} | b_i = 0\}. \quad (3.7)$$

If $a_{i1} \leqslant 0$, and $a_{i2} \leqslant 0$, then the constraint i is a redundant constraint. Moreover, if there exists $a_{i1} > 0$ and $a_{i2} > 0$ for some $i \in B_0$, then we can conclude that the optimal solution is the origin point without solving.

After the redundant constraints are eliminated, we partition B_0 into two sets, N_0 and P_0, by considering the coefficient value a_{i1} for $i \in B_0$. If $a_{i1} < 0$ for $i \in B_0$ (indicating that constraint i has a feasible region beneath itself, as depicted in Fig. 3 (a)), and its slope is positive, we set $i \in N_0$ and observe that the constraint with the minimum slope always offers a feasible direction. Conversely, if $a_{i1} > 0$ for $i \in B_0$ (indicating that constraint i has a feasible region above itself, as shown in Fig. 3 (b)), and its slope is positive, we set $i \in P_0$ and find that the constraint with the maximum slope always provides a feasible direction.

Therefore, N_0 and P_0 can be expressed as the following sets:

$$N_0 = \{i \mid a_{i1} < 0, a_{i2} > 0, \text{ and } i \in B_0\}, \quad (3.8)$$

$$P_0 = \{i \mid a_{i1} > 0, a_{i2} < 0, \text{ and } i \in B_0\}. \quad (3.9)$$

Thus, for $i \in N_0$, we set $\beta_i = -\frac{a_{i1}}{a_{i2}}$, which is computed to identify an initial circuit for this set, and for $i \in P_0$, we set $\gamma_i = -\frac{a_{i2}}{a_{i1}}$, which is the slope with respect to the y-axis.

Then, the initial circuit, $\mathbf{g}_r$, is selected by taking into account the sets N_0 and P_0, which may occur in three different scenarios. Let

$$ns = \arg\min \left\{ \beta_i = -\frac{a_{i1}}{a_{i2}} \mid i \in N_0 \right\}, \quad (3.10)$$

$$ps = \arg\min \left\{ \gamma_i = -\frac{a_{i2}}{a_{i1}} \mid i \in P_0 \right\}. \quad (3.11)$$

1. If $N_0 = \emptyset$ and $P_0 \neq \emptyset$ (Fig. 3 (b)), then the constraint that has the maximum slope always provides a feasible direction. So, the initial circuit is selected as $\mathbf{g}_r = \mathbf{g}_{ps}$.
2. If $N_0 \neq \emptyset$ and $P_0 = \emptyset$ (Fig. 3 (a)), then the constraint that has the minimum slope always provides a feasible direction. So, the initial circuit is selected as $\mathbf{g}_r = \mathbf{g}_{ns}$.
3. If $N_0 \neq \emptyset$ and $P_0 \neq \emptyset$ (Fig. 3 (c)), then let

$$G^+_{PN} = \{\mathbf{g}_i \in G^+ \mid b_i > 0, \quad (3.12)$$
$$i \in \{1, \ldots, m\} \cup \{ns, ps\}\}$$

An initial circuit can be found by computing

$$nd = \text{argmax} \left\{ \frac{\mathbf{c}^T \mathbf{g}_i}{\|\mathbf{c}\| \|\mathbf{g}_i\|} \mid \mathbf{g}_i \in G^+_{PN} \right\}, \quad (3.13)$$

and set $\mathbf{g}_r = \mathbf{g}_{nd}$.

Next, a step size can be obtained by determining the following set:

$$Ap = \{\alpha_i > 0 \mid i \in \{1, 2, \ldots, m+2\} \setminus \{r\}\}, \quad (3.14)$$

where $\alpha_i = \frac{b_i - A_{i:}\mathbf{x}_k}{A_{i:}\mathbf{g}_r}$, $A_{i:}\mathbf{g}_r \neq 0$, and $A_{i:}$ is the coefficient vector of the constraint i for all $i \in \{1, ..., m+2\}$ and $i \neq r$. If $Ap = \emptyset$, meaning $\alpha_i \leqslant 0$ for all i, then the algorithm is terminated and the current solution $\mathbf{x}_k$ is the optimal solution. Otherwise, $p = \arg\min\{Ap\}$ and the current solution

Table 3. The criterion for selecting an initial circuit in a degenerate problem.

```
Sub-Algorithm 2: The selection of an initial circuit of degenerate case.
1:  begin
2:      Let $N_0$ and $P_0$ be the empty sets;
3:      Find the set of all potential circuits
            $G^+ = \{\mathbf{g}_i | i = 1, 2, \ldots, m\}$,
        and the set of all degenerate constraints
            $B_0 = \{i \in \{1, \ldots, m\} | b_i = 0\}$;
4:      Set $N_0 = \{i \mid a_{i1} < 0, a_{i2} > 0, i \in B_0\}$,
        and $P_0 = \{i \mid a_{i1} > 0, a_{i12} < 0, i \in B_0\}$;
5:      If $N_0 = \emptyset$ and $P_0 \neq \emptyset$ then
6:          $ps = \arg\min \left\{\gamma_i = -\frac{a_{i2}}{a_{i1}} \mid i \in P_0\right\}$;
7:          return $\mathbf{g}_r = \mathbf{g}_{ps}$;
8:      Else If $N_0 \neq \emptyset$ and $P_0 = \emptyset$ then
9:          $ns = \arg\min \left\{\beta_i = -\frac{a_{i1}}{a_{i2}} \mid i \in N_0\right\}$;
10:         return $\mathbf{g}_r = \mathbf{g}_{ns}$;
11:     Else If $N_0 \neq \emptyset$ and $P_0 \neq \emptyset$ then
12:         Let $G^+_{NP} = \{\mathbf{g}_i | i \in \{1, \ldots, m\} \cup \{ns, ps\}\}$;
13:         $nd = \text{argmax} \left\{\frac{\mathbf{c}^T \mathbf{g}_i}{\|\mathbf{c}\| \|\mathbf{g}_i\|} \mid \mathbf{g}_i \in G^+_{PN}\right\}$;
14:         return $\mathbf{g}_r = \mathbf{g}_{nd}$;
15:     Else return $\mathbf{g}_r = \mathbf{c}$;
16: End
```

$\mathbf{x}_k$ is updated by $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_p \mathbf{g}_r$. To verify the optimality of $\mathbf{x}_{k+1}$, it can be determined similar to the circuit direction search algorithm.

Therefore, the steps for selecting an initial circuit in a degenerate case are summarized in Table 3.

Utilizing slope information in conjunction with the circuit direction, the new criterion for selecting an initial circuit in degenerate cases seeks to enhance CDSA's performance. This amalgamation of approaches is termed the slope-circuit hybrid method (SCHM). The ensuing subsection provides a detailed explanation of SCHM.

3.3 The Slope-Circuit Hybrid Method (SCHM)

Consider the following special two-dimensional linear programming problem:

$$\begin{aligned}
\max \quad & z = c_1 x_1 + c_2 x_2 \\
\text{s.t.} \quad & a_{11} x_1 + a_{12} x_2 \leqslant b_1, \\
& a_{21} x_1 + a_{22} x_2 \leqslant b_2, \\
& \vdots \\
& a_{m1} x_1 + a_{m2} x_2 \leqslant b_m, \\
& x_1, x_2 \geqslant 0,
\end{aligned} \tag{3.15}$$

where $c_1, c_2 > 0$, $b_i \geqslant 0$ for all $i = 1, 2, \ldots, m$ and $a_{ij} \in \mathbb{R}$ are not both zero simultaneously for all $i = 1, 2, \ldots, m$ and for all $j = 1, 2$..

The circuit direction search algorithm was initially introduced in [24] to solve a specific non-degenerate two-dimensional linear programming problem. It can also be applied to solve a special degenerate two-dimensional linear programming problem, although it requires more time to achieve the optimal solution. In some cases, an initial solution may be updated to an infeasible point, leading to an incorrect optimal solution to the problem.

Therefore, the new criterion presented in Subsection 3.2 is applied to the circuit direction search algorithm to enhance its efficiency in handling degenerate cases. The steps of the slope-circuit hybrid method for performing a special two-dimensional linear programming problem (3.15) are summarized in Table 4.

Then, an example demonstrating the application of the slope-circuit hybrid method will be illustrated in Subsection 3.4.

Table 4. The steps of the slope-circuit hybrid method.

```
Algorithm 2: The slope-circuit hybrid method.
 1: begin
 2:     Set $k, l, p = 0$, $\alpha_p = 0$ and $\mathbf{x}_k = (0,0)$;
 3:     If $b_i > 0$ for all $i = 1, \dots, m$ then
            Perform Sub-Algorithm 1;
 4:     Else Perform Sub-Algorithm 2;
 5:     While $\mathbf{g}_r \neq \mathbf{0}$ do
 6:         $Ap = \{\alpha_i > 0 \mid i \in \{1,2,\dots,m\} \setminus \{r\}\}$;
            where $\alpha_i = \frac{b_i - A_{i:}\mathbf{x}_k}{A_{i:}\mathbf{g}_r}$, $A_{i:}\mathbf{g}_r \neq 0$;
 7:         If $Ap = \emptyset$ then return $\mathbf{x}_k^* = \mathbf{x}_k, l, p$;
 8:         Else
 9:             $p = \arg\min\{Ap\}$;
10:             $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_p \mathbf{g}_r$;
11:             $k \leftarrow k+1$;
12:             For $i = 1, \dots, m$;
13:                 If $\mathbf{A}_{i:}\mathbf{x}_{k+1} = b_i$ and $i \neq r$ then
14:                     $l \leftarrow i$;
15:                     $i \leftarrow i+1$;
16:                 Else $\mathbf{g}_r = \mathbf{g}_p$;
17:             Construct a sub-matrix $A' = \begin{bmatrix} A_{l:} \\ A_{p:} \end{bmatrix}^T$;
18:             Compute $\mathbf{w} = (A')^{-1}\mathbf{c}$;
19:             If $\mathbf{w} \geqslant \mathbf{0}$ then return $\mathbf{x}_k^* = \mathbf{x}_k, l, p$;
20:             Else $\mathbf{g}_r = \mathbf{g}_l$;
21: End
```

3.4 An Illustrative Example

The following example shows how to use the slope-circuit hybrid method for solving a special two-dimensional linear programming problem.

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

$$\begin{array}{lrcll} \max & x_1 + 3x_2 & & & \\ \text{s.t.} & -2x_1 + x_2 & \leqslant & 0, & \cdots(1) \\ & -3x_1 + 2x_2 & \leqslant & 0, & \cdots(2) \\ & x_1 - 2x_2 & \leqslant & 0, & \cdots(3) \\ & 2x_1 + x_2 & \leqslant & 6, & \cdots(4) \\ & 5x_1 + 8x_2 & \leqslant & 40, & \cdots(5) \\ & 3x_1 + 8x_2 & \leqslant & 36, & \cdots(6) \\ & x_1 + 3x_2 & \leqslant & 12, & \cdots(7) \\ & x_1 & \geqslant & 0, & \cdots(8) \\ & x_2 & \geqslant & 0. & \cdots(9) \end{array} \tag{3.16}$$

Let $\mathbf{x}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ be an initial solution. Since there exists the right-hand side value $b_i = 0$ for $i \in \{1,2,3\}$, Sub-Algorithm 2 is performed to select an initial circuit. The set of all potential circuits contains the following circuits:

Table 5. All potential circuits of problem 3.16.

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

The non-negative constraints are identified in the set of degenerate constraints. Therefore, we get $B_0 = \{1,2,3,8,9\}$, and separate it into two sets $N_0 = \{1,2,8\}$ and $P_0 = \{3\}$. Since N_0 and P_0 are non-empty sets, the indices ns and ps are computed by

$$ns = \arg\min\left\{\beta_i = -\frac{a_{i1}}{a_{i2}} \mid i \in N_0\right\} = 2,$$

and

$$ps = \arg\min\left\{\gamma_i = -\frac{a_{i2}}{a_{i1}} \mid i \in P_0\right\} = 3.$$

Let $G_{NP}^+ = \{\mathbf{g}_i | i \in \{4,5,6,7\} \cup \{2,3\}\}$ and compute

$$nd = \text{argmax}\left\{\frac{\mathbf{c}^T\mathbf{g}_i}{\|\mathbf{c}\|\|\mathbf{g}_i\|} \mid \mathbf{g}_i \in G_{PN}^+\right\} = 2.$$

Thus, $\mathbf{g}_2$ is selected as an initial circuit. Next, the step size α_p is obtained by computing

$$p = \text{argmin}\{\alpha_i | \alpha_i > 0, i \in \{1,...,9\} \setminus \{2\}\} = 4,$$

where $\alpha_i = \frac{b_i - A_{i:}\mathbf{x}_0}{A_{i:}\mathbf{g}_2}$ and $A_{i:}\mathbf{g}_r \neq 0$, Therefore, the updated solution is

$$\mathbf{x}_1 = \mathbf{x}_0 + \alpha_4 \mathbf{g}_2 = \begin{bmatrix} 1.71 \\ 2.57 \end{bmatrix}.$$

Since $\mathbf{x}_1$ binds at the constraints 2 and 4, we get, we get

$$A' = \begin{bmatrix} -3 & 2 \\ 2 & 1 \end{bmatrix}^T \text{ and } \mathbf{w} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \geqslant \mathbf{0}.$$

Therefore, $\mathbf{x}_1 = \begin{bmatrix} 1.71 \\ 2.57 \end{bmatrix}$ is the optimal solution with $z^* = 6$.

For Example 3.1, the circuit direction search algorithm selects $\mathbf{g}_1$ as an initial circuit, and updates $\mathbf{x}_0$ to an infeasible point, which binds at the conditions 1 and 4. After testing the optimality, this infeasible point is the optimal solution to the problem, which is incorrect. Thus, the new criterion will enable the circuit direction search algorithm to select a suitable circuit direction for the degenerate case of a special two-dimensional linear programming problem. Additionally, the slope algorithm [7] requires three iterations to solve this example, while the proposed algorithm only requires one iteration. Figs. 4-6 illustrate the movement of the feasible solution in each iteration for the proposed algorithm (Fig. 4), the circuit direction search algorithm (Fig. 5), and the slope algorithm (Fig. 6).

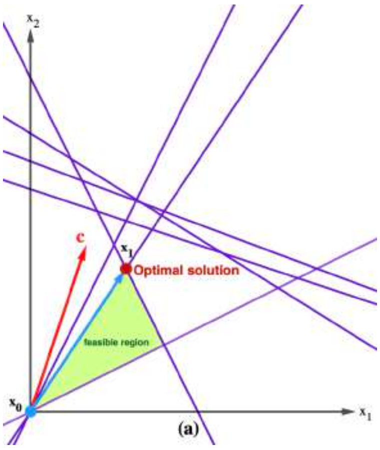

Fig. 4. Illustration of Example 3.1 solved by the proposed algorithm.

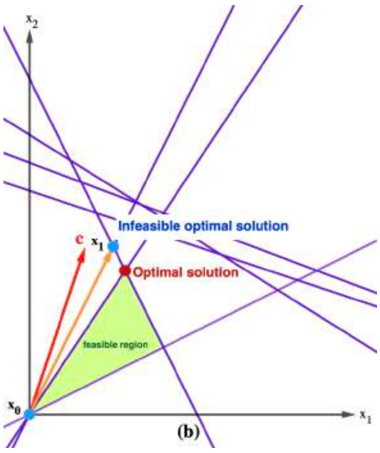

Fig. 5. Illustration of Example 3.1 solved by the circuit direction search algorithm.

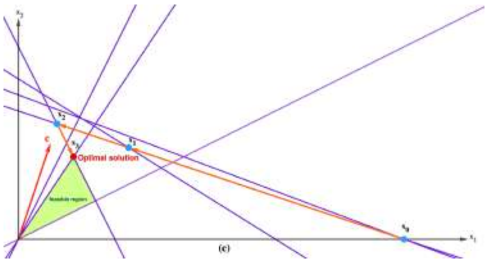

Fig. 6. Illustration of Example 3.1 solved by the slope algorithm.

4. Experimental Results

In this section, we present the comparative results for the average number of iterations and running time of the Slope-Circuit Hybrid Method (SCHM), Slope Algorithm (SA), Simplex Method (SM), and Interior Point Method (IPM). These algorithms, SCHM, SA, SM, and IPM, were implemented using Python on Google Colab. We then generated random instances of general degeneracy two-dimensional linear programming problems and 25%, 50%, 75% degeneracy problems, each with 50, 100, 200, 300, 400, 500, 600, 700, 800, 900, and 1,000 constraints, to assess the efficiency of the proposed algorithm. The comprehensive results, including average iterations and running time, are detailed in Tables 6-7. Additionally, Figs. 7-10 provide graphical representations of the average number of iterations and running time presented in Tables 6-7. Furthermore, Fig. 7 illustrates the percentage improvement in the average number of iterations and running time of the proposed algorithm compared to the simplex method. The percentage improvement can be computed by

$$\begin{aligned}&\%\text{Improvement iteration}\\&=\frac{\text{Avg. \#iter. of SM}-\text{Avg. \#iter. of SCHM}}{\text{Avg. \#iter. of SM}},\end{aligned}\tag{4.1}$$

$$\begin{aligned}&\%\text{Improvement running time}\\&=\frac{\text{Avg. Time of SM}-\text{Avg. Time of SCHM}}{\text{Avg. Time of SM}},\end{aligned}\tag{4.2}$$

From the comprehensive testing of the proposed algorithms' efficiency, it is evident that SCHM consistently demonstrates the minimum average number of iterations and running time among all the algorithms considered. Subsequently, we calculate the percentage improvement of the average number of iterations and running time of the proposed algorithm compared to the simplex method using Eqs. (4.1)-(4.2). The results of these percentage improvements are depicted in Fig. 11.

Based on the data presented in Fig. 11, we computed the average percentage improvement for each degeneracy scenario. SCHM demonstrates significant improvements of 24.35%, 65.39%, 61.73%, and 67.52% in the average percentage improvement of the number of iterations for general degeneracy, 25%, 50%, and 75% degeneracy tested problems, respectively. Moreover, SCHM exhibits substantial improvements of 93.97%, 95.97%, 94.26%, and 96.97% in the average percentage improvement of running time for general degeneracy, 25%, 50%, and 75% degeneracy tested problems, respectively. These findings affirm that SCHM effectively enhances both the average number of iterations and running time compared to the simplex method.

5. Conclusions

In this study, we introduce the slope-circuit hybrid method (SCHM), a novel approach that significantly enhances the efficiency and versatility of the circuit direction search algorithm for solving linear programming problems. Specifically, SCHM addresses the critical challenge of selecting appropriate initial circuits in degenerate cases, a common obstacle in LP optimization. Through a carefully designed criterion that considers potential circuits derived from degenerate constraints, SCHM ensures robust convergence towards optimal solutions. Empirical results demonstrate that SCHM consistently outperforms existing methods, including the interior point method, slope algorithm, and simplex method, in terms of both the average number of iterations and overall running time.

Moreover, SCHM extends the capability of the circuit direction search algorithm beyond its original scope. While previously limited to identifying two entering and leaving variables in non-degenerate cases, SCHM successfully operates in both degenerate and non-degenerate scenarios, enabling its application as a double pivot within the simplex method. This advancement unlocks the potential to solve a broader range of general linear programming problems with enhanced efficiency.

The promising results of this study pave the way for future research exploring further extensions of SCHM to higher-dimensional LP problems and its integration into advanced optimization frame-

Table 6. The average number of iterations and running time of randomly generated general degeneracy and 25% degeneracy two-dimensional linear programming problems.

#Constraints	General Problem								25% Degeneracy Problem							
	Average #iterations				Average running time				Average #iterations				Average running time			
	IPM	SA	SM	SCHM	IPM	SA	SM	SCHM	IPM	SA	SM	SCHM	IPM	SA	SM	SCHM
50	7.6	3.5	2.1	**1.5**	0.010	0.004	0.011	**0.003**	7.0	2.2	3.1	**1.3**	0.022	0.004	0.010	**0.002**
100	8.5	6.3	2.1	**1.7**	0.032	0.007	0.035	**0.006**	7.9	2.7	3.1	**1.6**	0.056	0.006	0.042	**0.004**
200	9.0	5.7	2.1	**1.8**	0.056	0.022	0.153	**0.010**	7.8	4.1	4.1	**1.5**	0.098	0.012	0.152	**0.008**
300	9.2	8.5	2.2	**1.5**	0.102	0.018	0.284	**0.010**	8.3	3.5	3.8	**1.1**	0.202	0.016	0.368	**0.008**
400	10.1	9.5	2.0	**1.5**	0.231	0.039	0.589	**0.014**	8.6	4.4	3.6	**1.5**	0.334	0.026	0.652	**0.012**
500	9.2	8.6	2.5	**2.0**	0.335	0.032	0.917	**0.021**	8.3	3.5	4.7	**1.3**	0.504	0.030	1.110	**0.014**
600	9.3	8.2	2.7	**1.8**	0.533	0.068	1.540	**0.027**	9.0	4.0	5.0	**1.4**	0.848	0.068	1.594	**0.016**
700	9.5	9.8	2.9	**2.0**	0.747	0.060	1.777	**0.030**	8.5	2.9	4.4	**1.3**	0.832	0.048	2.292	**0.022**
800	8.3	9.2	2.4	**1.7**	0.902	0.054	2.430	**0.029**	9.0	3.3	5.2	**1.5**	1.286	0.050	3.250	**0.026**
900	8.8	8.3	2.3	**2.1**	1.126	0.048	2.973	**0.039**	9.9	2.4	4.3	**1.4**	1.946	0.066	3.822	**0.028**
1000	9.7	9.7	2.6	**1.9**	1.698	0.071	3.695	**0.044**	9.8	3.1	4.8	**1.6**	2.484	0.068	4.828	**0.038**
Average Total	9.02	7.94	2.35	**1.77**	0.525	0.038	1.309	**0.021**	8.55	3.28	4.19	**1.41**	0.783	0.0358	1.647	**0.016**

Table 7. The average number of iterations and running time of randomly generated 50% and 75% degeneracy two-dimensional linear programming problems.

#Constraints	50% Degeneracy Problem								75% Degeneracy Problem							
	Average #iterations				Average running time				Average #iterations				Average running time			
	IPM	SA	SM	SCHM	IPM	SA	SM	SCHM	IPM	SA	SM	SCHM	IPM	SA	SM	SCHM
50	7.7	2.5	2.8	**1.3**	0.012	**0.004**	0.010	**0.004**	7.9	2.1	2.4	**1.3**	0.022	0.004	0.012	**0.002**
100	8.2	2.6	2.4	**1.6**	0.037	0.008	0.069	**0.006**	7.7	1.5	4.0	**1.1**	0.032	0.006	0.044	**0.002**
200	8.3	1.7	3.2	**1.3**	0.062	0.012	0.160	**0.006**	8.5	1.7	4.4	**1.1**	0.066	0.012	0.178	**0.006**
300	8.5	2.4	3.2	**1.4**	0.112	0.016	0.346	**0.008**	8.6	1.5	4.4	**1.4**	0.122	0.016	0.396	**0.008**
400	9.0	4.3	3.6	**1.6**	0.216	0.024	0.588	**0.014**	8.9	1.6	5.2	**1.3**	0.260	0.024	0.752	**0.010**
500	8.5	1.4	4.3	**1.2**	0.335	0.026	1.034	**0.014**	9.2	1.5	3.9	**1.4**	0.406	0.030	1.050	**0.014**
600	9.0	4.1	3.5	**1.3**	0.538	0.036	1.458	**0.020**	9.5	1.5	4.7	**1.3**	0.640	0.044	1.690	**0.016**
700	8.5	4.6	4.5	**1.5**	0.740	0.042	2.225	**0.022**	9.0	2.8	4.3	**1.8**	0.794	0.052	2.110	**0.022**
800	8.6	3.3	4.2	**1.6**	1.160	0.052	2.632	**0.024**	9.2	4.1	3.8	**1.3**	1.164	0.042	2.736	**0.018**
900	8.9	1.7	4.9	**1.0**	2.212	0.048	3.544	**0.026**	9.4	2.6	4.8	**1.4**	1.502	0.060	3.538	**0.026**
1000	9.5	1.3	5.4	**1.2**	2.372	0.062	5.068	**0.028**	9.1	1.6	5.2	**1.3**	1.944	0.070	4.608	**0.028**
Average Total	8.61	2.72	3.82	**1.36**	0.708	0.030	1.557	**0.015**	8.82	2.05	4.28	**1.34**	0.632	0.033	1.556	**0.014**

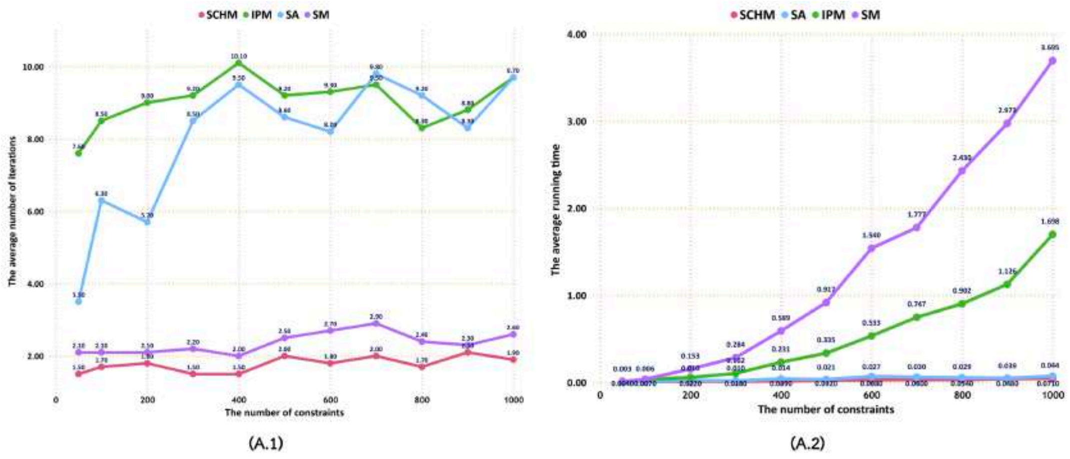

Fig. 7. Graphs of Table 4: (A) The average number of iterations (A.1) and running time (A.2) of randomly generated general degeneracy problems.

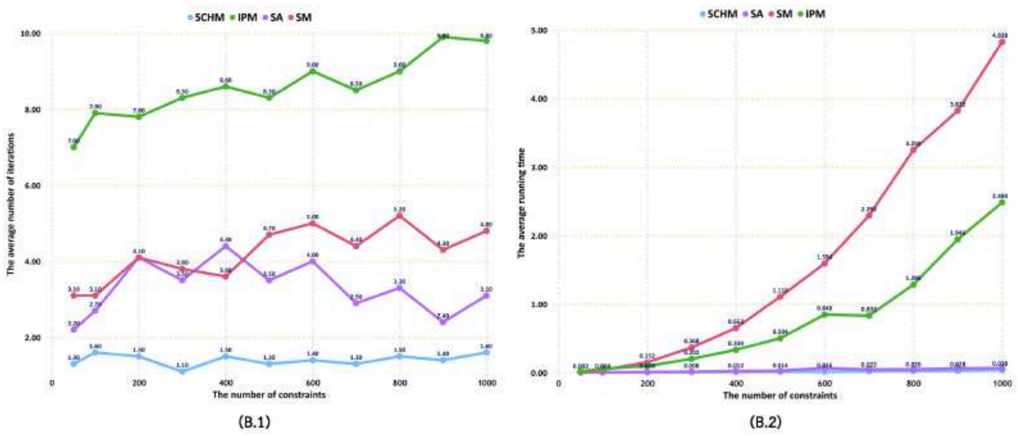

Fig. 8. Graphs of Table 4: (B) The average number of iterations (B.1) and running time (B.2) of randomly generated 25% degeneracy problem.

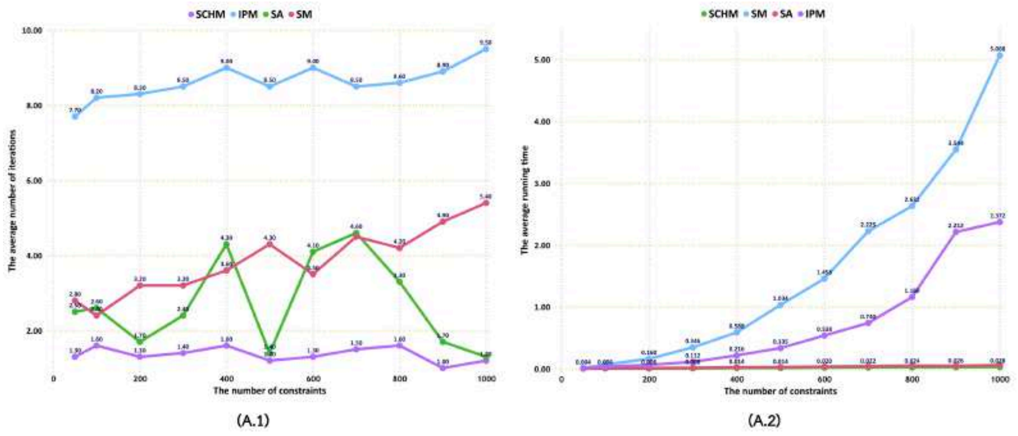

Fig. 9. Graphs of Table 5: (A) The average number of iterations (A.1) and running time (A.2) of randomly generated 50% degeneracy problems.

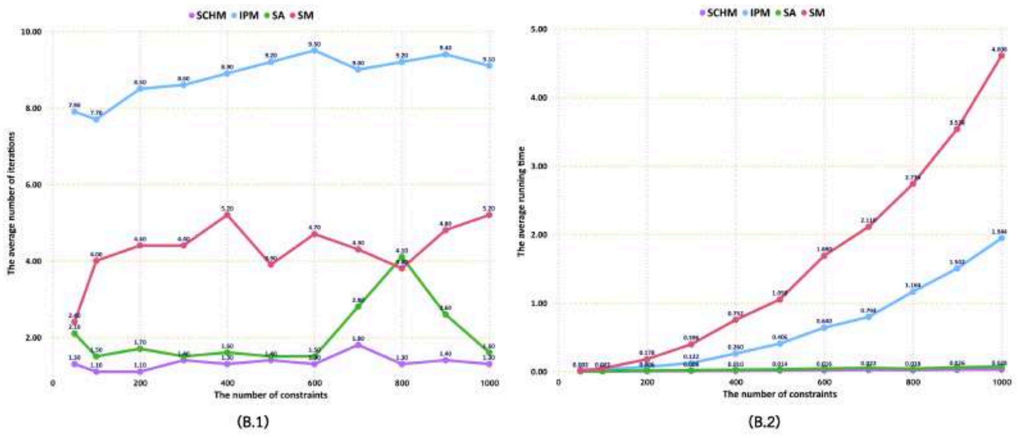

Fig. 10. Graphs of Table 5: (B) The average number of iterations (B.1) and running time (B.2) of randomly generated 75% degeneracy problem.

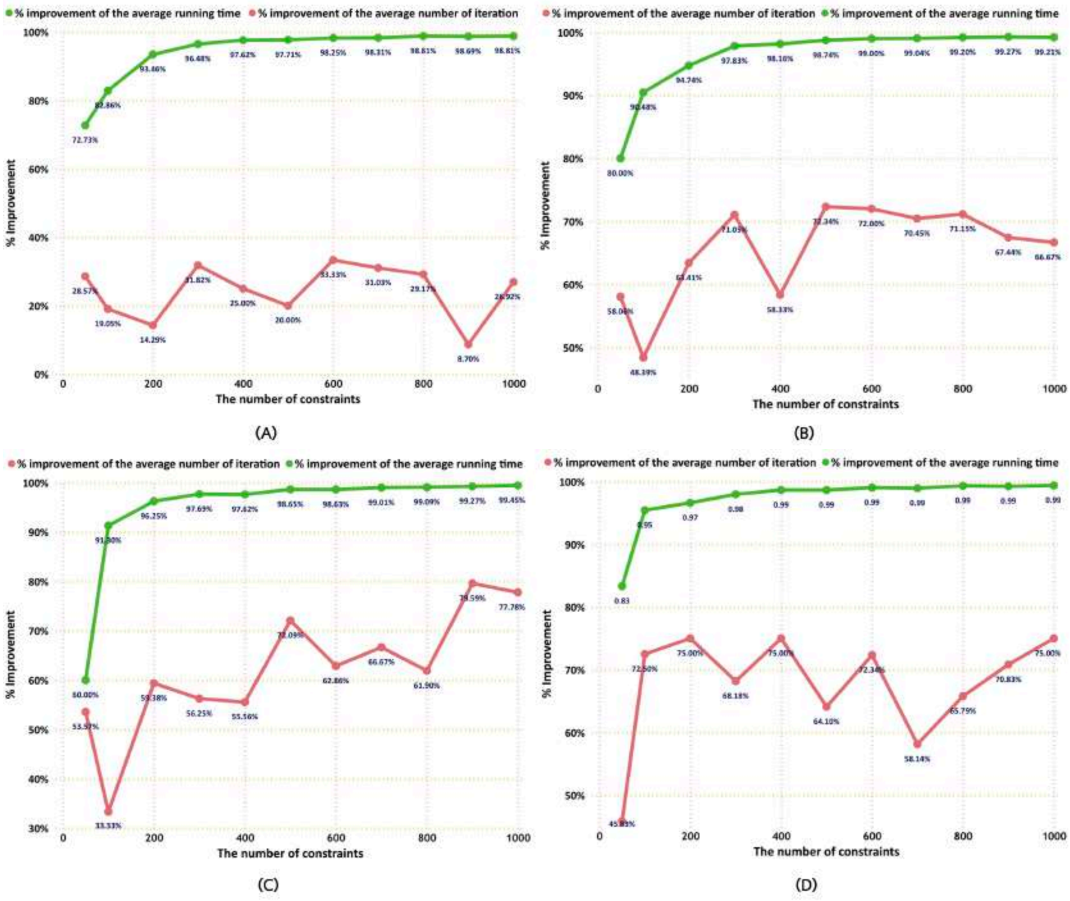

Fig. 11. Graphs of the percentage improvements of average number of iterations and running time of (A) general degeneracy, (B) 25%, (C) 50% degeneracy and (D) 75% degeneracy.

works. By addressing fundamental challenges in LP optimization, SCHM has the potential to significantly impact various fields that rely on linear programming techniques, such as operations research, economics, and engineering.

References

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

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

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

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

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

[6] A. Boonperm and K. Sinapiromsaran, "The artificial-free technique along the objective direction for the simplex algo-

rithm," Journal of Physics: Conference Series. 2014:490;012193.

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

[8] 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.

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

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

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

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

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

[14] 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.

[15] 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.

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

[17] Bland R.G. New finite pivoting rules for the simplex method. Mathematics of Operations Research. 1977:2(2);103–7.

[18] Bachem A, Kern W. Linear Programming Duality: An Introduction to Oriented Matroids. Universitext. Springer-Verlag, 1984.

[19] Fukuda K, Terlaky T. Criss-cross methods: A fresh view on pivot algorithms. Mathematical Programming, Ser. B. 1997:79(1-3);369-95.

[20] Gauthier J.B, Desrosiers J, Lubbecke M. Decomposition theorems for linear programs. Operations Research Letters. 2014:42(8);553–7.

[21] Hemmecke R, Onn S, Weismantel R. A polynomial oracle-time algorithm for convex integer minimization. Mathematical Programming, Ser. A. 2011:126;97–117.

[22] 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.

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

[24] Jamrunroj P, Boonperm A. The Circuit Direction Search Algorithm for Solving Two-Dimensional Linear Programming Problems. Science & Technology Asia. 2023:28(1);48–59.