International Journal of Innovative Computing, Information and Control
Volume **20**, Number **3**, June **2024**

ICIC International ©2024 ISSN 1349-4198
pp. 755–770

WEIGHTED ITERATION ALGORITHM FOR SOLVING BI-OBJECTIVE LINEAR PROGRAMMING PROBLEM

Shuoqi Wang* and Zhanzhong Wang

Transportation College
Jilin University
No. 5988, Renmin Road, Changchun 130000, P. R. China
wangzz@jlu.edu.cn
*Corresponding author: wangsq@jlu.edu.cn

Received September 2023; revised December 2023

Abstract. *Based on the multi-objective linear weighted method and the characteristic that the sum of the weighted coefficients of the bi-objective function is 1, a weighted iteration algorithm for solving bi-objective linear optimization has been proposed. The basic principle of the algorithm is that when the weighted coefficients increase gradually from a very small value to 1, an efficient solution can be obtained. The iteration algorithm needs convergence conditions. This paper proves the relationship between the iterative convergence conditions and the optimal solution of the objective function, and multiple efficient solutions can be obtained by iteration. According to the actual demand of the project, a method of determining the most efficient solution is given. The main advantage of the algorithm is that the implementation process of the weighted iteration algorithm only needs the optimal solution of a single objective function, and there is no complex process or algorithm for solving weighted coefficients. The algorithm is simple and effective, and overcomes the shortcomings of the existing algorithms that have the complex parameter setting and solving process. The examples and application show that the weighted iteration algorithm is scientific and correct, and it is easy to be used and programmed and can play an important role in practical engineering application.*
Keywords: Linear programming, Iteration algorithm, Bi-objective linear programming, Multi-objective linear programming, Mathematical programming

1. **Introduction.** A study on the optimization of more than one objective function in a given region is called multi-objective programming. This is typically recorded as MOP (Multi-Objective-Programming). The intended purpose is to find that $\vec{x}^* \in X$ which maximizes (or minimizes) all objective functions simultaneously.

Because a bi-objective linear program is a special case of a multi-objective linear program, some of methods for solving multi-objective linear programs can be also used for solving bi-objective linear programs. Based on the relevant literature, methods for solving multi-objective linear programs can be divided into two categories: traditional optimization methods and intelligent optimization methods. Intelligent optimization methods include genetic algorithm [1], evolutionary algorithm [2], and particle swarm optimization [3]. Its advantage is that more Pareto solutions can be obtained, forming an efficient solution set. The disadvantages are that the algorithm is complex, more parameters are set, and the solution process is tedious. The traditional optimization method transforms multi-objective functions into a single objective function, and the single-objective function optimization method is used to solve the multi-objective function optimization. There are mainly the following methods [4-6]: the ideal point method, fuzzy deviation method,

DOI: 10.24507/ijicic.20.03.755

sub-objective multiplication and division method, max-min method, constraint method, goal programming method, weighted square sum method, linear weighted sum method, efficiency coefficient method, and coordination curve method.

In traditional optimization methods, the linear weighted sum method is widely used, and its core is to determine the weights for each objective function. Using the concept of an efficient solution, we have the following well-known theorem [7-10].

Theorem 1.1. *If a point $\vec{x}^* \in X$ is efficient, then there exists a vector $\vec{\alpha}^* \in E^n$ such that $\sum_{i=1}^{p} \alpha_i^* = 1$, $\alpha_i^* \geq 0$ and $\vec{x}^*$ is the solution of the following equivalent linear programming:*

$$\max(\min) f_{p+1}(\vec{x}) = \sum_{i=1}^{p} \alpha_i^* f_i(\vec{x}) \tag{1}$$
$$\textit{Subject to: } A\vec{x} \leq b, \quad \vec{x} \geq 0$$

That is, an efficient solution is found by selecting some positive weights α_i^* and solving the linear programming problem given in Theorem 1.1. In general, there are many efficient solutions, because there are many possible sets of positive weights. Many scholars have studied methods for determining the weights of objective functions. In early research, several heuristic approaches were proposed for determining the weights of objective functions [11-15]. At present, commonly used methods [16-20] include the zero-sum game method, entropy weight method, triangular fuzzy number method, fuzzy analytic hierarchy process, principal component analysis, and subjective weighted method. The disadvantage of these methods is that they are subjective with respect to the determination of the “best” set of weights, and the algorithms are more complex and more difficult to be programmed.

In recent years, methods for solving bi-objective linear optimization have been uesd. Kaoud et al. [21] used a robust optimization approach to solve a bi-objective optimization model. Azar et al. [22] used non-cooperative game theory to find a Pareto-Optimal Equilibrium solution. Li et al. [23] designed an epsilon-constraint-based two-phase iterative heuristic algorithm based on the characteristics of an optimization problem. Keshavarz-Ghorbani and Pasandideh [24] proposed a Lagrange relaxation algorithm for optimizing a bi-objective agro-supply chain model that considers CO_2 emissions. Khaleghzadeh et al. [25] proposed a method for solving the bi-objective optimization problem for heterogeneous processors. Ulusoy et al. [26] investigated a method to approximate the nondominated set of the DFC problem with guarantees of global nondominance. Shi et al. [27] proposed an epsilon-constrained hybrid evolutionary algorithm that combines a Greedy Randomized Adaptive Search Procedure-Evolutionary Local Search (GRASP-ELS) hybrid approach with the epsilon-constrained method. Guan et al. [28] designed a genetic algorithm with a variable neighborhood search technique and employed it to solve the bDFCAP in a more effective manner. Gai et al. [29] proposed an integrated two-stage multiple-criteria programming approach to solve the model systematically. This approach includes both quantitative and qualitative analyses. Zhao et al. [30] presented a memetic algorithm that integrates a population-based non-dominated sorting genetic algorithm II and two single-solution-based improvement methods, that is, an insertion-based local search and an iterated greedy algorithm. In addition, other related methods have been proposed. These methods are very complex and lack universal application, which have the complex parameter setting and solving process.

In practical engineering applications, there are many bi-objective linear programming problems [31-34], including bi-level programming problems [35], which can also be transformed into bi-objective linear programming problems. Based on Theorem 1.1 and the characteristics of bi-objective linear programming, we propose a simple and efficient weighted iteration algorithm for solving bi-objective linear programming. The basic principle of the algorithm is as follows. According to the relationship that the sum of the weight coefficients of two objective functions equals 1 ($\alpha+\beta=1$), when α increases gradually from a very small value and $\beta=1-\alpha$, an efficient solution can be obtained such that the value difference between the two objective functions with the efficient solution is less than a given number δ. Subsequently, by decreasing δ until $\delta \leq 0$, several efficient solutions can be found to form an efficient solution set from which the most efficient solution can be found. In this paper, a determination method for δ is given, and the most efficient solution discrimination method is also presented.

The innovations of the algorithm are as follows. 1) The method of determining the weight coefficients for objective functions is not complex; only the optimal solutions (maximum value or minimum value) of two objective functions need to be calculated, and the iterative algorithm for finding efficient solutions can be realized using the powerful computing ability of a computer (see examples 1-5). 2) Many efficient solutions can be obtained and a set of Pareto-optimal solutions can be formed (see examples 1, 3, 4). 3) The most efficient solution discrimination method was identified. At the same time, decision makers can select the most efficient solution from the efficient solution set according to their own preferences and optimization results (see example 4). 4) It can also solve the bi-level programming problem, that is, linear programming, considering the priority of the two objective functions (see example 4). 5) For a linear programming problem whose objective functions are greater than 0, the solution effect is the best, efficient solutions are less, and the discrimination of the most efficient solution is simple (see example 2 and application example). Most real-world engineering problems involve simultaneously optimizing bi-objectives whose functions are greater than 0. 6) The algorithm is simple, and programming is convenient (see Algorithm 1). It has recently been proven to be highly effective and robust for solving bi-objective optimization problems.

Based on the examples provided in the literature, the algorithm in this study was used to solve the same bi-objective optimization problems. Compared to the results calculated in the literature, the results are completely consistent. At the same time, it is also verified by the application case. Compared to the methods discussed in [11-35], it offers good innovation and practical engineering applications.

The remainder of this paper is organized as follows. Section 2 introduces the weighted iteration method. Section 3 discusses the examples. Section 4 provides a case application. Section 5 provides a summary of this study.

2. **Weighted Iteration Method.** As discussed above, it is impossible to obtain an optimal solution $\vec{x}^*$ that satisfies all objective functions. The optimal solution solved by objective function (1) is called the efficient optimal solution, or Pareto optimal (Pareto Optimality). Based on Theorem 1.1, bi-objective linear programming is expressed as follows.

Problem 2.1. *Solve the maximum value of the objective functions.*

$$\begin{aligned} &\max f_1(\vec{x}) = C_1\vec{x} \\ &\max f_2(\vec{x}) = C_2\vec{x} \\ &\max f_3(\vec{x}) = \alpha f_1(\vec{x}) + \beta f_2(\vec{x}) \\ &\textit{Subject to: } A\vec{x} \leq b, \quad \vec{x} \geq 0 \end{aligned} \tag{2}$$

where $\vec{x} = (x_1, x_2, \ldots, x_n)^T$ *are decision variables,* $C_i = (c_{i1}, c_{i2}, \ldots, c_{in})$ *are coefficient vectors of the objective functions,* $i = 1, 2$*,* $A = (a_{ij})_{m*n}$ *is the coefficient matrix of the constraint conditions, and* $b = (b_1, b_2, \ldots, b_m)^T$ *is the right vector of constraint conditions.*

Theorem 2.1. *For solving the maximum value of the bi-objective linear functions, if $f_1(\vec{x}) > 0$, $f_2(\vec{x}) > 0$ and the optimization solution for $f_1(\vec{x})$ is $\vec{x}^{*1}$, the optimization solution for $f_2(\vec{x})$ is $\vec{x}^{*2}$, then there is at least one optimal coefficient α^*, $\beta^* = 1 - \alpha^*$, making objective function $f_3(\vec{x})$ have at least an efficient optimization solution $\vec{x}^{*3}$, and satisfying the following inequality:*

$$\left|f_1\left(\vec{x}^{*3}\right) - f_2\left(\vec{x}^{*3}\right)\right| < \left|f_1\left(\vec{x}^{*1}\right) - f_2\left(\vec{x}^{*2}\right)\right| \tag{3}$$

$$\textit{Denote}\ \ \delta = \left|f_1\left(\vec{x}^{*1}\right) - f_2\left(\vec{x}^{*2}\right)\right|$$

$$\textit{Then}\ \ \left|f_1\left(\vec{x}^{*3}\right) - f_2\left(\vec{x}^{*3}\right)\right| < \delta \tag{4}$$

Proof:

Denote $f_1^* = f_1\left(\vec{x}^{*1}\right),\ f_2^* = f_2\left(\vec{x}^{*2}\right)$

$f_1\left(\vec{x}^{*3}\right) = f_1^* - \Delta f_1,\ f_2\left(\vec{x}^{*3}\right) = f_2^* - \Delta f_2$

Assume $f_1(\vec{x}) > f_2(\vec{x})$

$f_1^* - \Delta f_1 > f_2^* - \Delta f_2$

$f_1^* - f_2^* > \Delta f_1 - \Delta f_2$

Because $f_1^* - f_2^* > 0$

$\Delta f_1 - \Delta f_2 > 0$

we have

$$\begin{aligned} f_1\left(\vec{x}^{*3}\right) - f_2\left(\vec{x}^{*3}\right) &= f_1^* - \Delta f_1 - \left(f_2^* - \Delta f_2\right) \\ &= \left(f_1^* - f_2^*\right) - \left(\Delta f_1 - \Delta f_2\right) < f_1^* - f_2^* \end{aligned} \tag{5}$$

If $f_2(\vec{x}) > f_1(\vec{x})$, by the same principle, we have

$$f_2\left(\vec{x}^{*3}\right) - f_1\left(\vec{x}^{*3}\right) < f_2^* - f_1^* \tag{6}$$

Based on Equations (5) and (6), the following inequality exists:

$$\left|f_1\left(\vec{x}^{*3}\right) - f_2\left(\vec{x}^{*3}\right)\right| < \left|f_1\left(\vec{x}^{*1}\right) - f_2\left(\vec{x}^{*2}\right)\right|$$

Problem 2.2. *Solving the minimum value of the objective functions.*

$$\begin{aligned} &\min f_1(\vec{x}) = C_1\vec{x} \\ &\min f_2(\vec{x}) = C_2\vec{x} \\ &\min f_3(\vec{x}) = \alpha f_1(\vec{x}) + \beta f_2(\vec{x}) \\ &\textit{subject to: } A\vec{x} \le b, \quad \vec{x} \ge 0 \end{aligned} \tag{7}$$

where $\vec{x} = (x_1, x_2, \ldots, x_n)^T$ are decision variables, $C_i = (c_{i1}, c_{i2}, \ldots, c_{in})$ are coefficient vectors of the objective functions, $i = 1, 2$, $A = (a_{ij})_{m*n}$ is the coefficient matrix of the constraint conditions, and $b = (b_1, b_2, \ldots, b_m)^T$ is the right vector of constraint conditions.

Theorem 2.2. *For solving the minimum value of the bi-objective linear functions, if $f_1(\vec{x}) > 0$, $f_2(\vec{x}) > 0$ and the optimization solution for $f_1(\vec{x})$ is $\vec{x}^{*1}$, the optimization solution for $f_2(\vec{x})$ is $\vec{x}^{*2}$, then there is at least one optimal coefficient α^*, $\beta^* = 1 - \alpha^*$, making objective function $f_3(\vec{x})$ have at least an efficient optimization solution $\vec{x}^{*3}$, and satisfying the following inequality:*

$$\left|f_1\left(\vec{x}^{*3}\right) - f_2\left(\vec{x}^{*3}\right)\right| < \left|f_1\left(\vec{x}^{*1}\right) - f_2\left(\vec{x}^{*2}\right)\right| \tag{8}$$

$$\textit{Denote}\ \ \delta = \left|f_1\left(\vec{x}^{*1}\right) - f_2\left(\vec{x}^{*2}\right)\right|$$

$$\textit{Then}\ \ \left|f_1\left(\vec{x}^{*3}\right) - f_2\left(\vec{x}^{*3}\right)\right| < \delta \tag{9}$$

Proof:

Denote $f_1^* = f_1\left(\vec{x}^{*1}\right),\ f_2^* = f_2\left(\vec{x}^{*2}\right)$

$f_1\left(\vec{x}^{*3}\right) = f_1^* + \Delta f_1,\ f_2\left(\vec{x}^{*3}\right) = f_2^* + \Delta f_2$

Assume $f_1(\vec{x}) > f_2(\vec{x})$
$f_1^* + \Delta f_1 > f_2^* + \Delta f_2$
$f_1^* - f_2^* > \Delta f_2 - \Delta f_1$
Because $f_1^* - f_2^* > 0$
$\Delta f_2 - \Delta f_1 > 0,\ \Delta f_1 - \Delta f_2 < 0$
we have

$$\begin{aligned} f_1\left(\vec{x}^{*3}\right) - f_2\left(\vec{x}^{*3}\right) &= f_1^* + \Delta f_1 - (f_2^* + \Delta f_2) \\ &= (f_1^* - f_2^*) + (\Delta f_1 - \Delta f_2) < f_1^* - f_2^* \end{aligned} \tag{10}$$

If $f_2(\vec{x}) > f_1(\vec{x})$, by the same principle, we have

$$f_2\left(\vec{x}^{*3}\right) - f_1\left(\vec{x}^{*3}\right) < f_2^* - f_1^* \tag{11}$$

Based on Equations (10) and (11), the following inequality exists:

$$\left|f_1\left(\vec{x}^{*3}\right) - f_2\left(\vec{x}^{*3}\right)\right| < \left|f_1\left(\vec{x}^{*1}\right) - f_2\left(\vec{x}^{*2}\right)\right|$$

Definition 2.1. *For Equations (4) and (9), when δ decreases, it is possible that many efficient optimal solutions exist, and forming an efficient solution set. Denote $\delta_{i+1} = \delta_i - \Delta\delta$, $i = 0, 1, \ldots, n$. If there are n efficient optimal solutions, the most efficient optimal solution should satisfy the following conditions:*

$$\textit{1) } f_1\left(\vec{x}^{*3}\right) \textit{ is closer to } f_1\left(\vec{x}^{*1}\right) \tag{12}$$

$$\textit{2) } f_2\left(\vec{x}^{*3}\right) \textit{ is closer to } f_2\left(\vec{x}^{*2}\right) \tag{13}$$

Definition 2.2. *Objective function value difference. Let fe be the objective function value difference, then*

$$fe = \left|f\left(\vec{x}^{*3}\right) - f(\vec{x}^*)\right| \tag{14}$$

where $\vec{x}^*$ is the optimal solution of the objective function, and $\vec{x}^{*3}$ is the efficient solution.

Definition 2.3. *Efficient solution difference. Let E be efficient solution difference, then*

$$E = fe_1 + fe_2 = \left|f_1\left(\vec{x}^{*3}\right) - f_1\left(\vec{x}^{*1}\right)\right| + \left|f_2\left(\vec{x}^{*3}\right) - f_2\left(\vec{x}^{*2}\right)\right| \tag{15}$$

According to the above definitions and the actual demand, a method for determining the most efficient solution is proposed.

1) First, remove the non-inferior solutions that only satisfy the optimal solution of one objective function, and make another objective function have the largest objective function value difference. That is

$$f_1\left(\vec{x}^{*3}(i)\right) \in f_1\left(\vec{x}^{*1}\right) \text{ and } \max\left(\left|f_2\left(\vec{x}^{*3}(i)\right) - f_2\left(\vec{x}^{*2}\right)\right|\right) \tag{16}$$

$$\text{or } f_2\left(\vec{x}^{*3}(i)\right) \in f_2\left(\vec{x}^{*1}\right) \text{ and } \max\left(\left|f_1\left(\vec{x}^{*3}(i)\right) - f_1\left(\vec{x}^{*2}\right)\right|\right) \tag{17}$$

Because it only satisfies the optimal solution of one objective function and maximizes the value difference of the other objective function, it is not the purpose of bi-objective function optimization and does not satisfy the requirements of the optimal efficient solution.

2) For the remaining efficient solutions, the solution with the smallest efficient solution difference is the most efficient solution without considering the importance or priority of the two objective functions. That is

$$\vec{x}^{*3} \in \min\left(\left|f_1\left(\vec{x}^{*3}(i)\right) - f_1\left(\vec{x}^{*1}\right)\right| + \left|f_2\left(\vec{x}^{*3}(i)\right) - f_2\left(\vec{x}^{*2}\right)\right|\right) \tag{18}$$

3) When considering the importance or priority of two objective functions, the objective function with high priority is given a large weight (usually 0.6), and the other is given

a small weight (usually 0.4). An efficient solution with the smallest weighted efficient solution difference is the most efficient solution. That is

$$\vec{x}^{*3} \in \min(\rho * fe_1(i) + \mu * fe_2(i)), \quad \rho + \mu = 1 \tag{19}$$

If there are two or more efficient solutions with equal weighted efficient solution differences, the smallest objective function value difference of the objective function with high priority is the most efficient solution. That is

$$\vec{x}^{*3} \in \min\left(\left|f_1\left(\vec{x}^{*3}(i)\right) - f_1\left(\vec{x}^{*1}\right)\right|\right) \tag{20}$$

$$\text{or} \quad \vec{x}^{*3} \in \min\left(\left|f_2\left(\vec{x}^{*3}(i)\right) - f_2\left(\vec{x}^{*2}\right)\right|\right) \tag{21}$$

Programming steps are shown as follows.

Step 1: given bi-objective linear programming, solve for each objective function $f_1(\vec{x})$, $f_2(\vec{x})$ individually, obtain the optimal solutions: $\vec{x}_1^*$, $\vec{x}_2^*$, calculate the function value: $f_1^* = f_1(\vec{x}_1^*)$, $f_2^* = f_2(\vec{x}_2^*)$, determine the initial value: $\delta = |f_1^* - f_2^*|$, number of the efficient optimal solutions: $i = 0$, the maximum number of iterations: k, counter of iteration: $n = 0$, and function coefficient: $\alpha = 0$.

Step 2: determine $\alpha = \alpha + \Delta\alpha$, $\beta = 1 - \alpha$. (Weighted iteration)

Step 3: solve for the objective function $f_3(\vec{x})$, get the efficient solution $\vec{x}_3^*$, $n = n + 1$.

Step 4: calculate the function value: $f_1^* = f_1(\vec{x}_3^*)$, $f_2^* = f_2(\vec{x}_3^*)$, and judge whether the following inequality is true.

$$|f_1^* - f_2^*| < \delta$$

A: yes, go to step 6;

B: no, continue to step 5.

Step 5: judge whether the number of iterations meets the requirements.

$$n \geq k$$

A: yes, go to step 7;

B: no, go to step 2.

Step 6: find an efficient solution $\vec{x}_3^*$, $\vec{x}^*(i) = \vec{x}^{*3}$, $i = i + 1$.

Step 7: let $\delta = \delta - \Delta\delta$, $n = 0$, $\alpha = 0$, reset k, and judge whether the following inequality is true:

$$\delta \leq 0$$

A: yes, go to step 8;

B: no, go to step 2.

Step 8: according to the method above, which determines the most efficient solution, the most efficient solution is obtained.

For step 2, because $0 < \alpha < 1$, referring to the idea of GA binary coding, 8-bit binary coding is carried out with a resolution of 0.0039, which can meet the iterative requirements, that is, $\Delta\alpha = 1/256$, $\alpha = \alpha + 1/256$, and $k = 255$.

For step 6, based on programming verification, $\Delta\delta \leq 1$. Generally, $\Delta\delta = 1$. In practical applications, according to the calculation results of objective functions, we can appropriately increase or decrease $\Delta\delta$, and find more efficient solutions as much as possible to facilitate the determination of the most efficient solution. The algorithm program was implemented as follows, based on the Python programming language.

The premise using Theorems 2.1 and 2.2 is $f_1(\vec{x}) > 0$, $f_2(\vec{x}) > 0$. If $f_1(\vec{x}) < 0$, $f_2(\vec{x}) < 0$ or $f_1(\vec{x}) > 0$, $f_2(\vec{x}) < 0$ or $f_1(\vec{x}) < 0$, $f_2(\vec{x}) > 0$, the following theorem is given.

Theorem 2.3. *For solving the optimization of the bi-objective linear functions, assuming that the minimum value for* $f_1(\vec{x})$ *is* $f_{1\min}$*, the maximum value for* $f_1(\vec{x})$ *is* $f_{1\max}$*, the minimum value for* $f_2(\vec{x})$ *is* $f_{2\min}$*, and the maximum value for* $f_2(\vec{x})$ *is* $f_{2\max}$*, then there*

Algorithm 1. A weighted iteration algorithm for solving the efficient solution

Input: $f_1(\vec{x})$, $f_2(\vec{x})$, $f_3(\vec{x})$ and the constraint conditions
Output: the optimal solutions $\vec{x}_1^*$, $\vec{x}_2^*$, $\vec{x}_3^*$
$f_1^* = f_1(\vec{x}_1^*)$, $f_2^* = f_2(\vec{x}_2^*)$, $f_1^* = f_1(\vec{x}_3^*)$, $f_2^* = f_2(\vec{x}_3^*)$

1. **Initialization**
2. Solve the objective function $f_1(\vec{x})$, $f_2(\vec{x})$
3. Calculate $f_1^* = f_1(\vec{x}_1^*)$, $f_2^* = f_2(\vec{x}_2^*)$
4. $\delta = |f_1^* - f_2^*|$
5. $n = 0$, $i = 0$, $\alpha = 0$, $\Delta\delta = 1$
6. **while True:**
7.   **for** n **in range** (255):
8. $\alpha = \alpha + 1/256$
9. $\beta = 1 - \alpha$
10.   Solve the objective function $f_3(\vec{x})$
11.   Get efficient solution $\vec{x}_3^*$
12.   Calculate $f_1^* = f_1(\vec{x}_3^*)$, $f_2^* = f_2(\vec{x}_3^*)$
13.   **if** $|f_1^* - f_2^*| < \delta$:
14.   Find efficient solution $\vec{x}_3^*$, $i = i + 1$
15.   **break**
16. $n = 0$, $\alpha = 0$
17. $\delta = \delta - \Delta\delta$
18.   **if** $\delta \le 0$:
19.   **break**
20.   **else:**
21.   **continue**
22. Determine the best efficient solution

is at least one optimal coefficient α^*, $\beta^* = 1 - \alpha^*$, *making objective function* $f_3(\vec{x})$ *have at least an efficient solution* $\vec{x}^{*3}$, *and satisfying the following inequality:*

$$\left|f_1\left(\vec{x}^{*3}\right) - f_2\left(\vec{x}^{*3}\right)\right| \le \delta \tag{22}$$

$$\text{and} \quad \delta = \max\left(\left|f_{1\max} - f_{2\min}\right|, \left|f_{2\max} - f_{1\min}\right|\right) \tag{23}$$

Proof:

Because $f_{1\min} \le f_1\left(\vec{x}^{*3}\right) \le f_{1\max}$
$f_{2\min} \le f_2\left(\vec{x}^{*3}\right) \le f_{2\max}$

δ is the maximum error among the values of the two functions,

$$\left|f_1\left(\vec{x}^{*3}\right) - f_2\left(\vec{x}^{*3}\right)\right| \le \delta$$

According to the above programming steps, δ is determined based on Equation (23) in step 1, and the other programming steps are the same.

3. **Examples.** In order to verify the correctness of the algorithm, examples of [4] and [10] are used. [4] used ideal point method, fuzzy deviation method, and linear weighted sum method. [10] used the two-person-zero sum game approach. Compared to the results in the reference, the results of this paper are consistent with those of the references.

3.1. **Example 1.**

$$\max f_1(x) = 0.1x_1 + 0.2x_2$$
$$\max f_2(x) = 10x_1 - 5x_2$$
$$\max f_3(x) = \alpha f_1(x) + (1-\alpha) f_2(x) = (10 - 9.9\alpha)x_1 + (5.2\alpha - 5)x_2$$

$$\begin{aligned} \text{subject to: } & x_1 + x_2 \geq 1 \\ & x_1 + x_2 \leq 7 \\ & 0 \leq x_1 \leq 5 \\ & 0 \leq x_2 \leq 3 \end{aligned}$$

According to Theorem 2.1, the maximum value for $f_1(\vec{x})$ equals 1, and the maximum value for $f_2(\vec{x})$ equals 50, and we have

$$\delta = \left| f_1\left(\vec{x}^{*1}\right) - f_2\left(\vec{x}^{*2}\right) \right| = 49$$

Table 1 shows a comparison of the efficient solution results between the methods in [10] and those in this study. According to Equation (16), the second efficient solution should be removed ($f_1\left(\vec{x}^{*3}\right) = f_1\left(\vec{x}^{*1}\right) = 1$), and the most efficient solution is the same as the result in [10].

TABLE 1. Comparison of the efficient solution results for example 1

Methods	Efficient solution $\left(x_1^{*3}, x_2^{*3}\right)$	Objective function value $f_1\left(x^{*3}\right)$	Objective function value $f_2\left(x^{*3}\right)$	δ	Judgement
Method in [10]	$(5, 2)$	0.9	40		
Method in this paper	$\mathbf{(5, 2)}$	**0.9**	**40**	$\mathbf{40 \leq \delta \leq 49}$	**The most efficient solution**
	$(4, 3)$	1	25	$24 \leq \delta \leq 39$	Deleted
	No			$0 \leq \delta \leq 23$	

3.2. **Example 2.**

$$\begin{aligned} & \min f_1(x) = 0.1x_1 + 0.2x_2 \\ & \min f_2(x) = 10x_1 + 5x_2 \\ & \min f_3(x) = \alpha f_1(x) + (1-\alpha) f_2(x) = (10 - 9.9\alpha)x_1 + (5 - 4.8\alpha)x_2 \\ & \text{subject to: } x_1 + x_2 \geq 1 \\ & \qquad x_1 + x_2 \leq 7 \\ & \qquad 0 \leq x_1 \leq 5 \\ & \qquad 0 \leq x_2 \leq 3 \end{aligned}$$

According to Theorem 2.2, the minimum value for $f_1(\vec{x})$ equals 0.1, and the minimum value for $f_2(\vec{x})$ equals 5, and we have

$$\delta = \left| f_1\left(\vec{x}^{*1}\right) - f_2\left(\vec{x}^{*2}\right) \right| = 4.9$$

Table 2 shows a comparison of the efficient solution results between the methods in [10] and in this study. There is only one efficient solution, which is as same as the result obtained using the two-person-zero sum game approach.

3.3. **Example 3.**

$$\begin{aligned} & \max f_1(x) = -0.1x_1 \\ & \max f_2(x) = 0.1x_1 + 0.2x_2 \\ & \max f_3(x) = \alpha f_1(x) + (1-\alpha) f_2(x) = (0.1 - 1.1\alpha)x_1 + 0.2(1-\alpha)x_2 \\ & \text{subject to: } -x_1 + x_2 \leq 1 \end{aligned}$$

$$x_1 + x_2 \leq 7$$
$$0 \leq x_1 \leq 5$$
$$0 \leq x_2 \leq 3$$

According to Theorem 2.3 and the calculation results, we have

$$f_{1\max} = 0, \quad f_{1\min} = -5$$
$$f_{2\max} = 1, \quad f_{2\min} = 0.1$$
$$\delta = \max(0.1, 6) = 6$$

TABLE 2. Comparison of the efficient solution results for example 2

Methods	Efficient solution (x_1^{*3}, x_2^{*3})	Objective function value $f_1(x^{*3})$	$f_2(x^{*3})$	δ	Judgement
Method in [10]	$(0, 1)$	0.2	5.0		
Method in this paper	$\mathbf{(0, 1)}$	**0.2**	**5.0**	$\mathbf{3 \leq \delta \leq 4.9}$	**The most efficient solution**
	No			$0 \leq \delta \leq 2$	

Table 3 shows a comparison of the efficient solution results between the methods in [10] and in this study. According to Equations (16) and (17), there are three efficient solutions, and the first and third efficient solutions should be removed (because $f_2(\vec{x}^{*3}) = f_{2\max} = 1$, $f_1(\vec{x}^{*3}) = f_{1\max} = 0$). The second most efficient solution is the most efficient solution, which is the same as the result in [10].

TABLE 3. Comparison of the efficient solution results for example 3

Methods	Efficient solution (x_1^{*3}, x_2^{*3})	Objective function value $f_1(x^{*3})$	$f_2(x^{*3})$	δ	Judgement
Method in [10]	$(2, 3)$	-2	0.8		
Method in this paper	$(4, 3)$	-4	1	$\delta = 5, 6$	Deleted
	$\mathbf{(2, 3)}$	$\mathbf{-2}$	**0.8**	$\boldsymbol{\delta = 4, 3}$	**The most efficient solution**
	$(0, 1)$	0	0.2	$\delta = 2, 1$	Deleted
	No			$\delta = 0$	

3.4. **Example 4.**

$$\max f_1(x) = -2x_1 + x_2$$
$$\max f_2(x) = 3x_1 - x_2$$
$$\max f_3(x) = \alpha f_1(x) + (1-\alpha) f_2(x) = (3 - 5\alpha)x_1 + (2\alpha - 1)x_2$$
$$\text{subject to: } -x_1 + x_2 \leq 1$$
$$x_1 + x_2 \leq 7$$
$$0 \leq x_1 \leq 5$$

$$0 \le x_2 \le 3$$

According to Theorem 2.3 and the calculation results, we have

$$f_{1\max} = 1, \quad f_{1\min} = -10$$
$$f_{2\max} = 15, \quad f_{2\min} = -1$$
$$\delta = \max(2, 25) = 25$$

Table 4 shows a comparison of the efficient solution results between the methods in [10] and in this study. According to Equation (17), there are four efficient solutions and the first efficient solution should be removed ($f_2(\vec{x}^{*3}) = f_{2\max} = 15$). There are three remaining efficient solutions.

1) $\vec{x}^{*3} = (5,2)$, $f_1(\vec{x}^{*3}) = -8$, $f_2(\vec{x}^{*3}) = 13$
2) $\vec{x}^{*3} = (4,3)$, $f_1(\vec{x}^{*3}) = -5$, $f_2(\vec{x}^{*3}) = 9$
3) $\boldsymbol{\vec{x}^{*3} = (2,3)}$, $\boldsymbol{f_1(\vec{x}^{*3}) = -1}$, $\boldsymbol{f_2(\vec{x}^{*3}) = 3}$

According to Equation (19), because $f_1(x)$ has high priority, the weighted efficient solution difference is $\min(0.6 * fe_1(i) + 0.4 * fe_2(i)) = \min(6.2, 6, 6) = (6, 6)$.

According to Equation (20), the fourth efficient solution is the most efficient solution, which is the same as the result in [10]. In fact, the third efficient solution is the one solved for the first time in the literature, and it is also consistent with the results in the literature.

TABLE 4. Comparison of the efficient solution results for example 4

Methods	Efficient solution (x_1^{*3}, x_2^{*3})	Objective function value $f_1(x^{*3})$	Objective function value $f_2(x^{*3})$	δ	Judgement
Method in [10]	$(2,3)$	-1	3		
Method in this paper	$(5,0)$	-10	15	$\delta = 25$	Deleted
	$(5,2)$	-8	13	$22 \le \delta \le 24$	Deleted
	$(4,3)$	-5	9	$14 \le \delta \le 21$	Deleted
	$\mathbf{(2,3)}$	$\mathbf{-1}$	**3**	$\boldsymbol{4 \le \delta \le 13}$	**The most efficient solution**
	No			$0 \le \delta \le 3$	

3.5. Example 5.

$$\max f_1(x) = -3x_1 + 2x_2$$
$$\max f_2(x) = 4x_1 + 3x_2$$
$$\max f_3(x) = \alpha f_1(x) + (1-\alpha) f_2(x) = (4 - 7\alpha)x_1 + (3 - \alpha)x_2$$
$$\text{subject to: } 2x_1 + 3x_2 \le 18$$
$$2x_1 + x_2 \le 10$$
$$x_1, x_2 \ge 0$$

According to Theorem 2.1, the maximum value for $f_1(\vec{x})$ equals 12, the maximum value for $f_2(\vec{x})$ equals 24, and we have

$$\delta = \left| f_1(\vec{x}^{*1}) - f_2(\vec{x}^{*2}) \right| = 12$$

Table 5 shows the comparison of the efficient solution results between the methods in [4] and in this paper.

Table 5. Comparison of the efficient solution results for example 5

Methods	Efficient solution $\left(x_1^{*3}, x_2^{*3}\right)$	Objective function value $f_1\left(x^{*3}\right)$	Objective function value $f_2\left(x^{*3}\right)$	δ	Judgement
Method in [4]	$(0.52, 5.64)$	9.71	19.05		Ideal point method
	$\mathbf{(0, 6)}$	**12**	**18**		Weighted method
	$(1.02, 5.31)$	7.55	20.05		Fuzzy deviation method
Method in this paper	$\mathbf{(0, 6)}$	**12**	**18**	$\mathbf{0 < \delta \leq 12}$	**The most efficient solution**
	No			$\delta = 0$	

According to Equation (18), the result of the linear weighted sum method is the most efficient solution, which is the same as the result obtained by the method in this study. As Table 5 shows, there is only one efficient solution.

The above five examples include the different cases involved in Theorems 2.1, 2.2 and 2.3. Examples 1, 2 and 5 are the cases where the value of the objective function is greater than zero, and examples 3 and 4 are the cases where the value of the objective function is not greater than zero. At the same time, example 4 is the case where the objective function has priority, and example 5 is the comparison with other methods. By examples 1, 2 and 5, we can see that there are few efficient solutions if the value of objective function is greater than zero. From examples 1 to 5, only the optimal solutions (maximum value or minimum value) of two objective functions need to be calculated, so the algorithm is simple and programming is convenient. At the same time, there are many efficient solutions to be solved with the decrease of δ. The algorithm has been also used to solve the following problem of transportation volume regulation.

4. Case and Application.

4.1. Transportation volume regulation based on transportation time and transportation cost.

For a given urban agglomeration U, there are n cities, $C = \{1, 2, 3, \ldots, n\}$, $c_i \in C$ and the cities are connected by a given highway transportation network. Based on historical data, f_i, the transmitted transportation volume of one city per day can be predicted, $F = \{f_1, f_2, \ldots, f_n\}$. Total transmitted transportation volume of the urban agglomeration per day is $tf = \sum_{i=1}^{n} f_i$. Assume that there is transportation from city i to city j every day, x_{ij} represents the value of the transmitted transportation volume from city i to city j, and $f_i = \sum_{j=1}^{n} x_{ij}$. $x_{\min}(i, j)$ represents the minimum value of the transmitted transportation volume, t_{ij} represents the highway transportation time per unit transportation volume, and p_{ij} represents the highway transportation cost per unit transportation volume.

The problem of transportation volume regulation can be explained by the OD matrix of transportation volume; that is, the transportation volume between cities can be determined according to different requirements and constraints. The OD matrix is shown in Figure 1.

If f_i, t_{ij}, p_{ij} are known, the problem is how to solve x_{ij} based on the shortest transportation time and the minimum transportation cost.

O/D	c_1	c_2	$\cdots$	c_n
c_1	x_{11}	x_{12}	$\cdots$	x_{1n}
c_2	x_{21}	x_{22}	$\cdots$	x_{2n}
$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$
c_n	x_{n1}	x_{n2}	$\cdots$	x_{nn}

FIGURE 1. OD matrix of transportation volume

4.2. **Transportation volume regulation model.** Set $\vec{x}=\{x_{11}, x_{12}, \ldots, x_{nn}\}$, the above problem is transformed into the following bi-objective linear programming model, which includes the shortest transportation time, the minimum transportation cost, the limited total supply, and the constrained demand interval.

$$
\begin{aligned}
&\min f_1(\vec{x}) = \sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij}t_{ij} \\
&\min f_2(\vec{x}) = \sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij}p_{ij} \\
&\min f_3(\vec{x}) = \alpha f_1(\vec{x}) + \beta f_2(\vec{x}),\ \alpha+\beta=1 \\
&\text{subject to: } \sum_{i=1}^{n}\sum_{j}^{n} x_{ij} = tf, \quad j=1,2,\ldots,n \\
&\qquad \sum_{j=1}^{n} x_{ij} = f_i, \quad i=1,2,\ldots,n \\
&\qquad x_{ij} \geq x_{\min}(i,j), \quad i,j=1,2,\ldots,n
\end{aligned} \tag{24}
$$

The most efficient solution of the objective function (24) can be obtained by using the above weighted iteration method.

$$
\begin{aligned}
\text{Set } & X = [x_{11}, x_{12}, \ldots, x_{nn}]^T \\
& T = [t_{11}, t_{12}, \ldots, t_{nn}] \\
& P = [p_{11}, p_{12}, \ldots, p_{nn}]
\end{aligned}
$$

$$\text{Then } f_1(X) = T * X, \quad f_2(X) = P * X \tag{25}$$

$$f_3(X) = \alpha * f_1(X) + (1-\alpha) * f_2(X) = [\alpha * (T-P) + P] * X \tag{26}$$

The matrix operation can be carried out by using Equations (25) and (26) to solve X.

4.3. **Real example.** An expressway transport network was selected as an example, as shown in Figure 2. The five major cities connected by the G12 expressway in Jilin Province, China, are Yanbian (Y), Jilin (J), Changchun (C), Songyuan (S), and Baicheng (B).

1) OD matrix of expressway transportation time per unit transportation volume. To obtain real-time and dynamic data, the average transportation volume is calculated based on the historical data from the previous week. The OD matrix of the expressway transportation time per unit transportation volume is shown as follows, ignoring the transportation from city i to itself.

$$
T = \begin{bmatrix}
0.0 & 0.017 & 0.024 & 0.032 & 0.041 \\
0.015 & 0.0 & 0.007 & 0.016 & 0.025 \\
0.022 & 0.005 & 0.0 & 0.012 & 0.020 \\
0.029 & 0.013 & 0.011 & 0.0 & 0.013 \\
0.038 & 0.024 & 0.019 & 0.012 & 0.0
\end{bmatrix}
$$

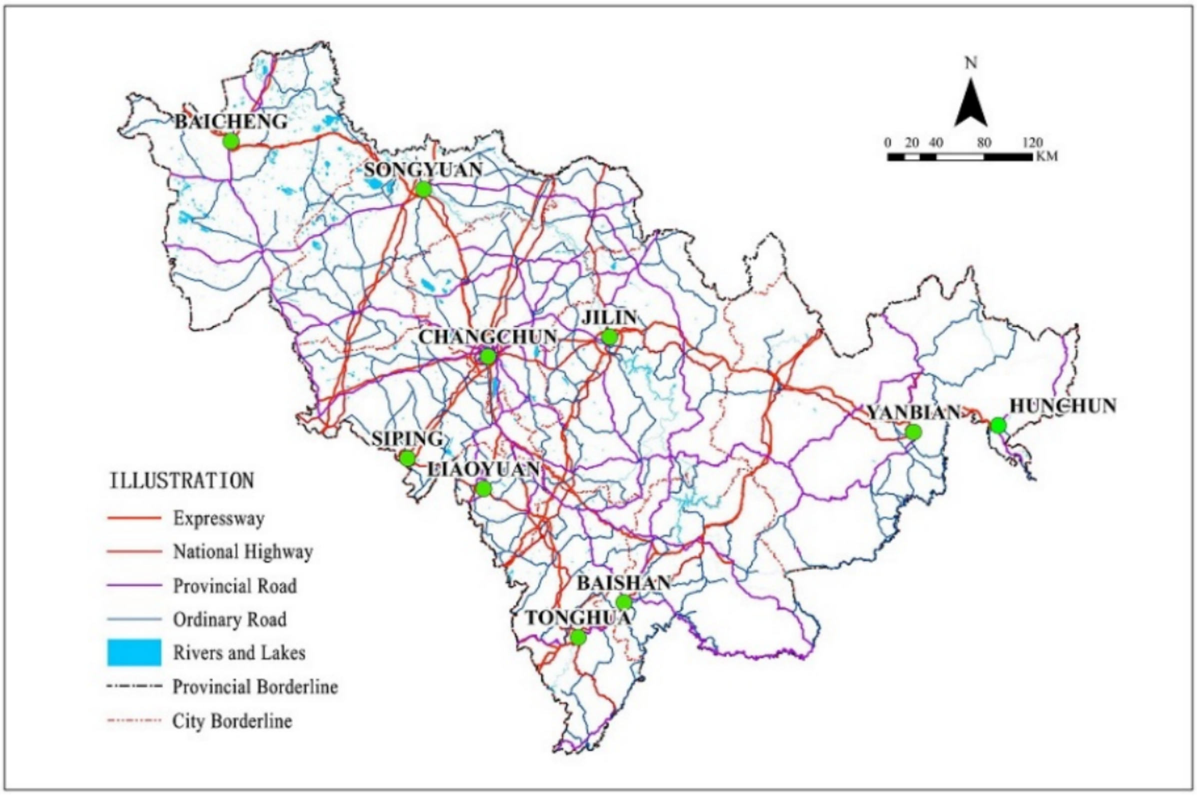

FIGURE 2. (color online) Main road networks in Jilin Province, China

2) OD matrix of expressway transportation cost per unit transportation volume. According to the calculation method [36], the OD matrix of the expressway transportation cost per unit transportation volume is as follows:

$$P = \begin{bmatrix} 0.0 & 340 & 405 & 536 & 699 \\ 340 & 0.0 & 145 & 279 & 404 \\ 405 & 145 & 0.0 & 172 & 339 \\ 536 & 279 & 172 & 0.0 & 188 \\ 699 & 404 & 339 & 188 & 0.0 \end{bmatrix}$$

where $c_{ii} = 0$, $c_{ij} = c_{ji}$, and P is symmetrical matrix.

3) Static regulation of the transmitted transportation volume per day. According to the static regulation and control method based on the game control principle [36], the transportation volume of each city is shown in Table 6 after the static regulation was completed.

TABLE 6. Estimated transportation volume

City	Y	J	C	S	B
Transportation volume	1250	3347	4800	2050	1257

Constraint conditions:

a) Variable interval: $x_{11} = (0, 0)$, $x_{12} = (182, 3000)$, $x_{13} = (630, 3000)$, $x_{14} = (120, 3000)$, $x_{15} = (150, 3000)$, $x_{21} = (250, 3000)$, $x_{22} = (0, 0)$, $x_{23} = (2330, 3000)$, $x_{24} = (340, 3000)$, $x_{25} = (280, 3000)$, $x_{31} = (200, 3000)$, $x_{32} = (2350, 4000)$, $x_{33} = (0, 0)$, $x_{34} = (760, 3000)$, $x_{35} = (420, 3000)$, $x_{41} = (200, 3000)$, $x_{42} = (300, 3000)$, $x_{43} = (1100, 3000)$, $x_{44} =$

$(0,0)$, $x_{45} = (300, 3000)$, $x_{51} = (100, 3000)$, $x_{52} = (200, 3000)$, $x_{53} = (600, 3000)$, $x_{54} = (132, 3000)$, $x_{55} = (0, 0)$.

b) Equality constraint:
$$\begin{aligned} x_{11} + x_{12} + x_{13} + x_{14} + x_{15} &= 1250 \\ x_{21} + x_{22} + x_{23} + x_{24} + x_{25} &= 3347 \\ x_{31} + x_{32} + x_{33} + x_{34} + x_{35} &= 4800 \\ x_{41} + x_{42} + x_{43} + x_{44} + x_{45} &= 2050 \\ x_{51} + x_{52} + x_{53} + x_{54} + x_{55} &= 1257 \end{aligned}$$

4) Calculation result. The regulated volume is solved using the weighted iteration method that is discussed in this study, and the calculation results are shown in Table 7.

TABLE 7. Calculation results

City	Y	J	C	S	B	The transmitted volume
Y	0	322	638	130	160	1250
J	260	0	2437	350	300	3347
C	230	3350	0	780	440	4800
S	220	350	1160	0	320	2050
B	130	250	636	241	0	1257

The above problem includes the solution of 25 variables, and if the two-person zero-sum game and other algorithms are used, the solving process will be very tedious. Because the value of the objective function is greater than 0, there is only one efficient solution solved by the weighted iteration algorithm, which is also the most efficient solution. This algorithm can also use matrix operation in the solving process, as shown in Equations (25) and (26), and the programming process is more concise.

5. **Conclusions.** In practical applications, there are many real-world bi-objective linear programming problems. At present, the methods for solving the efficient solution of bi-objective linear programming problems are more complex, and programming is also difficult. Sometimes, the most efficient solution cannot be found. A weighted iteration algorithm for solving the efficient solution of the bi-objective linear programming problem is proposed in this study, and the algorithm has the following advantages. 1) Only the maximum or minimum optimal solution of the two objective functions is required, and the most efficient solution can be obtained by iterating the weights of the objective function. 2) It has no complex calculation process and no complex reasoning, and it is fast, effective, and easy to program.

Later, we will study whether the weighted iteration method can be extended to solve multi-objective linear programming problems with more than two objective functions, thereby making the algorithm universal and providing a simple and effective method for solving multi-objective linear programming problems.

Acknowledgment. This work was partially supported by the National Natural Science Foundation of China under Grant 61873109, and the Natural Science Foundation of Jilin Province under Grant 20230101112JC. The authors gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] K. Abdullah, W. C. David and E. S. Alice, Multi-objective optimization using genetic algorithms: A tutorial, *Reliability Engineering & System Safety*, vol.91, no.9, pp.992-1007, 2006.

[2] S. Ruhul, K. H. Liang and C. Newton, A new multi-objective evolutionary algorithm, *European Journal of Operational Research*, vol.140, no.1, pp.12-23, 2002.
[3] J. F. Liu, H. Y. Zhang, K. He and S. Y. Jiang, Multi-objective particle swarm optimization algorithm based on objective space division for the unequal-area facility layout problem, *Expert Systems with Applications*, vol.102, pp.179-192, 2018.
[4] L. Liu, Some solutions of multi-objective linear programming and realized by LINGO, *Journal of Fujian Chuanzhen Communication Vocational College*, vol.10, no.6, pp.20-22, 2011.
[5] F. H. Han and Z. H. Zhang, Comparison between two solutions of multi-objective linear programming, *Journal of University of Science and Technology Beijing*, vol.25, no.1, pp.74-80, 2007.
[6] B. Noel, Identifying the efficient extreme-points of the three-objective linear programming problem, *Journal of the Operational Research Society*, vol.44, no.1, pp.81-85, 1993.
[7] R. Benayoun, J. de Motgolfier, J. Terguy et al., Linear programming with multiple objective functions: Step method (STEM), *Mathematical Programming*, pp.366-375, DOI: 10.1007/BF01584098, 1971.
[8] A. M. Geoffrion, Solving bicriterion mathematical programs, *Operation Research*, vol.15, pp.39-45, 1967.
[9] A. M. Geoffrion, Proper efficiency and theory of vector maximization, *Journal of Mathematical Analysis and Applications*, vol.22, no.3, pp.618-634, 1968.
[10] S. Belenson and K. Kapur, An algorithm for solving multicriterion linear programming problems with examples, *Journal of the Operational Research Society*, vol.24, no.1, pp.65-77, 1973.
[11] K. C. Kapur, Mathematical methods of optimization for multi-objective transportation systems, *Socio-Economic Planning Sciences*, vol.4, no.4, pp.451-463, 1970.
[12] C. Maier-Rothe and M. F. Stankard, *A Linear Programming Approach to Choosing between Multi-Objective Alternatives*, Arthur D. Little, Research Paper, 1970.
[13] R. Roy, Problems and methods with multiple objective functions. Introduction to the session: Multiple objective functions, *The 7th Int. Symp. on Mathematical Programming*, Hague, Netherlands, pp.24-28, 1970.
[14] J. L. Cohon, R. J. Church and D. P. Sheer, Generating multi-objective trade-offs: An algorithm for bicriterion problems, *Water Resource*, vol.15, no.5, pp.1001-1010, 1979.
[15] S. Moradi, A. Raith and M. Ehrgott, A bi-objective column generation algorithm for multi-commodity minimum cost flow problem, *European Journal of Operational Research*, vol.244, no.2, pp.369-378, 2015.
[16] R. Baucke, A. Downward and G. Zakeri, A deterministic algorithm for solving multistage stochastic minimax dynamic programmes, *Optimization Online*, http://www.optimization-online.org/db_html/2018/02/6449.html, 2018.
[17] Y. P. Huang and D. Christophe, Bi-objective methods for road network problems with disruptions and connecting requirements, *Journal of the Operational Research Society*, vol.71, no.12, pp.1959-1971, 2020.
[18] Y. Y. Tan and C. X. Guo, Bi-objective capacitated assortment planning under the MNL model: Trade-offs between revenue and market share, *Journal of the Operational Research Society*, DOI: 10.1080/01605682.2022.2157341, 2022.
[19] O. Dowson, D. P. Morton and A. Downward, Bi-objective multistage stochastic linear programming, *Mathematical Programming*, vol.196, pp.907-933, 2022.
[20] N. Jorge and C. Luís, Project portfolio management considering the commitment of agents: A bi-objective model applied to administrative services, *Journal of the Operational Research Society*, DOI: 10.1080/01605682.2022.2056530, 2022.
[21] E. Kaoud, M. A. M. Abdel-Aal and T. Sakaguchi, Robust optimization for a bi-objective green closed-loop supply chain with heterogeneous transportation system and presorting consideration, *Sustainability*, vol.14, no.16, pp.1023-1026, 2023.
[22] M. M. Azar, M. A. S. Monfared and S. E. Monabbati, Non-cooperative two-player games and linear bi-objective optimization problems, *Computers & Industrial Engineering*, vol.162, pp.275-296, 2021.
[23] Y. Li, F. Chu, F. F. Zheng and M. Liu, A bi-objective optimization for integrated berth allocation and quay crane assignment with preventive maintenance activities, *IEEE Transactions on Intelligent Transportation Systems*, vol.23, pp.2938-2955, 2022.
[24] F. Keshavarz-Ghorbani and S. H. R. Pasandideh, A Lagrangian relaxation algorithm for optimizing a bi-objective agro-supply chain model considering CO_2 emissions, *Annals of Operations Research*, vol.314, pp.497-527, 2022.

[25] H. Khaleghzadeh, M. Fahad and A. Shahid, Bi-objective optimization of data-parallel applications on heterogeneous HPC platforms for performance and energy through workload distribution, *IEEE Transactions on Intelligent Transportation Systems*, vol.32, pp.543-560, 2020.
[26] A. J. Ulusoy, F. Pecci and I. Stoianov, Bi-objective design-for-control of water distribution networks with global bounds, *Optimization and Engineering*, vol.23, pp.527-577, 2021.
[27] W. X. Shi, N. M. Wang and M. Zhang, A bi-objective pollution routing optimisation problem with decentralised cooperation and split delivery, *IEEE Transactions on Intelligent Transportation Systems*, vol.30, pp.643-660, 2023.
[28] C. Guan, Z. Q. Zhang and J. H. Gong, Mixed integer linear programming model and an effective algorithm for the bi-objective double-floor corridor allocation problem, *Computers & Operations Research*, vol.132, pp.1643-1660, 2021.
[29] L. Gai, H. C. Liu and Y. J. Wang, Green supplier selection and order allocation using linguistic Z-numbers MULTIMOORA method and bi-objective non-linear programming, *Fuzzy Optimization and Decision Making*, vol.22, pp.267-288, 2022.
[30] Z. Zhao, S. Liu, M. Zhou and A. Abusorrah, Dual-objective mixed integer linear program and memetic algorithm for an industrial group scheduling problem, *IEEE/CAA Journal of Automatica Sinica*, vol.8, no.6, pp.1199-1209, DOI: 10.1109/JAS.2020.1003539, 2021.
[31] M. Bacel, B. Fouad and H. T. Abdelaziz, Bi-objective optimization of retailer's profit and customer surplus in assortment and pricing planning, *Annals of Operations Research*, vol.296, pp.195-210, 2021.
[32] X. Y. Man, F. F. Zheng and F. Chu, Bi-objective optimization for a two-depot automated storage retrieval system, *Annals of Operations Research*, vol.296, pp.243-262, 2021.
[33] A. J. Ulusoy and I. Stoianov, Bi-objective design-for-control for improving the pressure management and resilience of water distribution networks, *Water Research*, vol.222, pp.118914-118925, 2022.
[34] Y. Wei, J. G. Jin, J. Yang and L. Lu, Strategic network expansion of urban rapid transit system: A bi-objective programming model, *Computer-Aided Civil and Infrastructure Engineering*, vol.34, no.5, pp.431-443, 2019.
[35] S. Y. Liu, Y. H. Yang and K. D. Chen, Efficient solutions of linear bilevel programming problem, *Journal of Systems Engineering*, vol.16, no.6, pp.438-442, 2001.
[36] S. Q. Wang and Z. Z. Wang, Collaborative development and transportation volume regulation strategy for an urban agglomeration, *Sustainability*, vol.15, no.20, pp.14742-14758, 2023.

Author Biography

Shuoqi Wang received the B.S. degree in City Planning from Shandong Jianzhu University, Jinan, China, in 2017; the M.S. degree in City Planning from Shandong Jianzhu University, Jinan, China, in 2020. She is currently pursuing the doctor's degree in the Transportation College, Jilin University, Changchun, China. Her research interests include integrated freight transportation corridors, data mining, machine learning, and traffic big data analysis.

Zhanzhong Wang received the B.S. degree in Transportation Planning and Management from Jilin University of Technology, Changchun, China, in 1986; the M.S. degree in Transportation Planning and Management from Jilin University, Changchun, China, in 1989. After working in Jilin Province Transportation Administration Bureau from 1989 to 1997 and in Jilin Ji Transport Group Co., Ltd. from 1997 to 2002, received the Ph.D. degree in Carrying Tools Applied Engineering from Jilin University in 2007. He is currently a Professor with Jilin University, devotes himself to transport resources optimization technology, mainly in the direction of production logistics operation, transportation economy and national economy evaluation, integrated transportation, and traffic big data analysis.