axioms

MDPI

Article

Robust Multi-Criteria Traffic Network Equilibrium Problems with Path Capacity Constraints

Xing-Xing Ma and Yang-Dong Xu *

Department of Mathematics, Chongqing University of Posts and Telecommunications, Chongqing 400065, China; maxx0205@126.com
* Correspondence: xyd04010241@126.com

Abstract: With the progress of society and the diversification of transportation modes, people are faced with more and more complicated travel choices, and thus, multi-criteria route choosing optimization problems have drawn increased attention in recent years. A number of multi-criteria traffic network equilibrium problems have been proposed, but most of them do not involve data uncertainty nor computational methods. This paper focuses on the methods for solving robust multi-criteria traffic network equilibrium problems with path capacity constraints. The concepts of the robust vector equilibrium and the robust vector equilibrium with respect to the worst case are introduced, respectively. For the robust vector equilibrium, an equivalent min–max optimization problem is constructed. A direct search algorithm, in which the step size without derivatives and redundant parameters, is proposed for solving this min–max problem. In addition, we construct a smoothing optimization problem based on a variant version of ReLU activation function to compute the robust weak vector equilibrium flows with respect to the worst case and then find robust vector equilibrium flows with respect to the worst case by using the heaviside step function. Finally, extensive numerical examples are given to illustrate the excellence of our algorithms compared with existing algorithms. It is shown that the proposed min–max algorithm may take less time to find the robust vector equilibrium flows and the smoothing method can more effectively generate a subset of the robust vector equilibrium with respect to the worst case.

Keywords: multi-criteria traffic network; robust vector equilibrium; min–max method; smoothing method

Citation: Ma, X.-X.; Xu, Y.-D. Robust Multi-Criteria Traffic Network Equilibrium Problems with Path Capacity Constraints. *Axioms* **2023**, *12*, 662. https://doi.org/10.3390/axioms12070662

Academic Editors: Siamak Pedrammehr and Mohammad Reza Chalak Qazani

Received: 5 June 2023
Revised: 24 June 2023
Accepted: 29 June 2023
Published: 3 July 2023

1. Introduction

Traveling is necessary for everyday human life. However, with the progress of society and the diversification of transportation modes, people also expect to find the most efficient route. Traffic network equilibrium problem can describe the distributions of traffic flows in the logistics industry and transportation network, which is expected to provide an effective method for travelers to choose an optimization route. The fundamental principle in the model is the concept of equilibrium that was initially introduced by Wardrop [1]. The principle asserts that travelers will choose the path only if the cost for this path is the minimum possible among all the paths joining the same O-D pair.

1.1. Literature Reviews

It has been shown that the Wardrop equilibrium concept is a powerful principle which is widely used in supply and demand networks, traffic assignment, optimization of traffic control, and other fields (see, e.g., Athanasenas [2]; Nagurney [3]; Ji and Chu [4]; Xu et al. [5]; Wang et al. [6]; Ma et al. [7]). It is worth noting that most of these equilibrium models in the above references are based on a single criterion. Travelers (in this paper, we use the terms 'user' and 'traveler' interchangeably) will naturally consider multi-criteria when choosing travel paths, including travel time, distance, cost, weather, safety, and other relevant factors. The equilibrium model with multi-criteria was first put forward by Chen and Yen [8], which was an extension of the classical Wardrop user equilibrium principle. Regarding the theoretical analysis for multi-criteria traffic equilibrium models, we refer the reader to Yang and Goh [9], Li et al. [10], Luc et al. [11], and Raith and Ehrgott [12].

Recently, Phuong and Luc [13] established the equivalent relationship between strong vector equilibrium flows and the solutions of variational inequality problems in terms of a kind of increasing functions. Moreover, they presented a modified projection method to handle multi-criteria traffic network equilibrium problems. Subsequently, Luc and Phuong [14] introduced two optimization problems to show that the optimal solutions are exactly the equilibrium of the traffic network and then put forward a modified Frank–Wolfe gradient algorithm for multi-criteria traffic network equilibrium problems. However, this method may lead to the non-differentiability of the objective functions of the two optimization problems. After that, Phuong [15] proposed a smoothing method to solve multi-criteria network equilibrium problems. Although this method solves the defects existing in [14], it does not take into account the data uncertainty.

In the actual traffic network, there are various uncertain factors, such as traveler preferences, weather, traffic congestion, and holidays. Hence, uncertainty in the logistics industry and transportation has received more and more attention. In recent years, some related works with uncertain demands or uncertain parameters in traffic network or ecological networks have been investigated in [16–21]. Daniele and Giuffré [16] investigated a general random traffic equilibrium problem and characterize the random Wardrop equilibrium distribution by means of a random variational inequality. Dragicevic and Gurtoo [17] modeled the maintenance of ecological networks in forest environments based on random processes, such as extreme natural events. However, the two above papers do not consider the multi-criteria. Recently, Ehrgott and Wang [18] presented alternative approaches for combining the principles of multi-objective decision-making with a stochastic user equilibrium model based on random utility theory. However, since uncertain parameters in [18] need to know probabilistic information, this may be inconsistent with the reality because the probabilistic information of related data are usually known. Cao et al. [19] and Wei et al. [20] only discussed relationships between and the solutions of variational inequality and robust equilibrium flows but not give the computational methods. Minh and Phuong [21] paid attention to a modified Frank–Wolfe gradient algorithm for robust equilibrium flows. The uncertain data in the model proposed in [21] are in a parameter set that does not need probability information. However, the computational efficiency of the algorithm is not good, due to the non-differentiability of the objective functions. In all, there are some research gaps on computational methods for robust multi-criteria traffic network equilibrium problems with path capacity constraint. This prompts us to continuously investigate this topic.

1.2. Contributions

To overcome computational inefficiency for the robust vector equilibrium flows in existing methods, this paper proposes two new computational methods for the robust vector equilibrium principle and the robust vector equilibrium principle with respect to the worst case, respectively. Firstly, an equivalent min–max optimization problem is constructed, in which the solution is equivalent to the robust vector equilibrium flow. A direct search algorithm with constraints for solving this problem is proposed. For the robust vector equilibrium with respect to the worst case, we transform it into an deterministic vector equilibrium problem based on a variant version of ReLU activation function. Then, we give an algorithm to solve the robust vector equilibrium with respect to the worst case.

In summary, the contributions of the manuscript are ranked in ascending gathered as follows:

(1) The robust vector equilibrium and the robust vector equilibrium with respect to the worst case principles are introduced.
(2) An equivalent min–max optimization problem is established and then a direct search algorithm is proposed to generate a subset of robust vector equilibrium flows.

(3) To generate a subset of the robust vector equilibrium with respect to the worst case, a two-step strategy is implemented. More specifically, a smoothing optimization problem is constructed based on a variant version of ReLU activation function to compute the robust weak vector equilibrium flows with respect to the worst case, and then, the robust vector equilibrium flows are found with respect to the worst case by using the heaviside step function.

This paper is divided into the following parts. Section 2 mainly introduces the robust vector equilibrium principle and robust vector equilibrium principle with respect to the worst case. Section 3 gives a min–max method to generate the subset of robust vector equilibrium flows. Section 4 presents a smoothing algorithm to find the subset of the robust vector equilibrium principle with respect to the worst case. Finally, conclusions of this paper and discussions for future research are provided in Section 5.

2. Definitions and Main Derivations

We review some fundamental definitions and properties that are relevant to this study. Throughout this paper, let $\mathbb{R}^*(* = n, m)$ denote the $*$-dimensional Euclidean space. Let $\mathbb{R}^m_+ := \{x \in \mathbb{R}^m : x_i \geq 0,\ i = 1, \cdots, m\}$ and $\mathbb{R}^m_{++} := \{x \in \mathbb{R}^m : x_i > 0,\ i = 1, \cdots, m\}$. The superscript $\top$ denotes transpose. The partial order in $\mathbb{R}^m$ is induced by $\mathbb{R}^m_+$, defined by:

$$x \geqq y \ \text{ if } \ x_i \geq y_i \ \text{ for all } \ i = 1, \ldots, m,$$

$$x \succeq y \ \text{ if } \ x_i \geq y_i \ \text{ for all } \ i = 1, \ldots, m \text{ and there exists } i_0 \text{ such that } x_{i_0} > y_{i_0}.$$

and the following stronger relation is given by:

$$x \succ y \ \text{ if } \ x_i > y_i \ \text{ for all } \ i = 1, \ldots, m.$$

Next, we will denote by e the vector of all ones. Given $X \subseteq \mathbb{R}^m$, the set of minimal elements of X is denoted by $\text{Min}(X)$, consists of vectors $x \in X$ such that there is no $x' \in X, x' \preceq x$.

Definition 1. *Given $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, we say that a point (x^*, y^*) is a saddle-point of the function f, if*

$$f(x^*, y) \leq f(x^*, y^*) \leq f(x, y^*),\ \forall (x, y) \in \mathbb{R} \times \mathbb{R}.$$

2.1. *Robust Multi-Criteria Traffic Network*

For a traffic network, $\mathcal{N}$ denotes the set of the nodes and $\mathcal{E}$ denotes the set of directed arcs. Let $\mathcal{W}$ be the set of origin–destination (O-D) pairs and $\mathcal{D} = (d_\omega)_{\omega \in \mathcal{W}}$ be the demand vector, where $d_\omega > 0$ is the flow demand on O-D pair ω. Thus, a traffic network is always denoted by $G = \{\mathcal{N}, \mathcal{E}, \mathcal{W}, \mathcal{D}\}$. For $\omega \in \mathcal{W}$, P_ω is the set of available paths on the O-D pair ω and $P = \cup_{\omega \in \mathcal{W}} P_\omega$ is the set of all available paths of the network. Let $n = \sum_{\omega \in \mathcal{W}} |p_\omega|$. For a given $p_k \in P_\omega$, y_{p_k} is the traffic flow on this path and $y = (y_1, y_2, \cdots, y_n)^\top \in \mathbb{R}^n$ is called a path flow. For given $p_k \in P_\omega$, suppose $l_{p_k} \in \mathbb{R}_+$, $u_{p_k} \in \mathbb{R}_+$ with $l_{p_k} < u_{p_k}$; the path flow y_{p_k} needs to satisfy the capacity constraint $l_{p_k} \leq y_{p_k} \leq u_{p_k}$. The traffic load is always presented by arc flows z_α, $\alpha \in \mathcal{E}$, or path flows y_{p_k}, $p_k \in P$. Given a path flow, the arc flow can be obtained by the following formula:

$$z_\alpha = \sum_{p_k \in P} y_{p_k} \delta_{\alpha p_k},$$

where

$$\delta_{\alpha p_k} = \begin{cases} 1, & \text{if } \alpha \text{ belongs to path } p_k, \\ 0, & otherwise. \end{cases}$$

The arc flow is denoted by $z := (z_\alpha)_{\alpha\in\mathcal{E}}$. A path flow y is said to be feasible flow if it satisfies:

$$\Omega = \{y \in \mathbb{R}_+ : \forall\omega \in \mathcal{W}, \forall p_k \in P_\omega,\ l_{p_k} \le y_{p_k} \le u_{p_k}, \sum_{p_k\in P_\omega} y_{p_k} = d_\omega\}.$$

Let $t_\alpha : \mathbb{R}^n \to \mathbb{R}^m$ be a vector-valued cost function along with arc $\alpha \in \mathcal{E}$. Let $c_{p_k} : \mathbb{R}^n \to \mathbb{R}^m$ be a vector-valued cost function on the path p_k. Thus, we have that the cost function c_{p_k} for path p_k is the sum of cost functions for arcs belonging to path p_k, namely:

$$c_{p_k}(y) = \sum_{\alpha\in\mathcal{E}} \delta_{\alpha p_k} t_\alpha(y). \tag{1}$$

However, the path cost functions may be perturbed in reality. This means that it not only depends on the path flow y but also on parameters of $\xi \in U := U_1 \times U_2 \times \cdots \times U_n$. Throughout this paper, the cost function $c_{p_k}(y, \xi)$ is often given in the form $c_{p_k}(y, \xi) = c_{p_k}(y) + \xi_{p_k}$.

2.2. Robust Vector Equilibrium and Related Concepts

Now, we give the following definitions on a robust vector equilibrium and a robust (weak) vector equilibrium with respect to the worst case.

Definition 2. *A feasible flow $\bar{y} \in \Omega$ is said to be a robust vector equilibrium, if for each O-D $\omega \in W$, path p_k, $p_j \in P_\omega$, one has:*

$$c_{p_k}(\bar{y}, \xi) - c_{p_j}(\bar{y}, \xi) \succeq 0_{\mathbb{R}^m},\ \forall\xi \in U \Rightarrow \text{ either } \bar{y}_{p_k} = l_{p_k} \text{ or } \bar{y}_{p_j} = u_{p_j}.$$

The worst case of the cost function on the path p_k under all possible scenarios is defined as follows:

$$C_{p_k}(y) = \begin{pmatrix} \sup\limits_{\xi\in U} c_{1p_k}(y, \xi) \\ \vdots \\ \sup\limits_{\xi\in U} c_{mp_k}(y, \xi) \end{pmatrix},\ C_{p_j}(y) = \begin{pmatrix} \sup\limits_{\xi\in U} c_{1p_j}(y, \xi) \\ \vdots \\ \sup\limits_{\xi\in U} c_{mp_j}(y, \xi) \end{pmatrix}$$

The following definitions are given based on the worst case of path costs, which is called the robust vector equilibrium with respect to the worst case and the robust weak vector equilibrium with respect to the worst case.

Definition 3. *A feasible flow $\bar{y} \in \Omega$ is a robust vector equilibrium with respect to the worst case, if for $\forall\,\omega \in W$, $\forall\, p_k$, $p_j \in P_\omega$, one has:*

$$C_{p_k}(\bar{y}) - C_{p_j}(\bar{y}) \succeq 0_{\mathbb{R}^m} \Rightarrow \text{ either } \bar{y}_{p_k} = l_{p_k} \text{ or } \bar{y}_{p_j} = u_{p_j}.$$

Definition 4. *A feasible flow $\bar{y} \in \Omega$ is in robust weak vector equilibrium with respect to the worst case , if for $\omega \in W$, p_k, $p_j \in P_\omega$, one has:*

$$C_{p_k}(\bar{y}) - C_{p_j}(\bar{y}) \succ 0_{\mathbb{R}^m} \Rightarrow \text{ either } \bar{y}_{p_k} = l_{p_k} \text{ or } \bar{y}_{p_j} = u_{p_j}.$$

Remark 1. *What should be noteworthy is that a robust vector equilibrium with respect to the worst case is also a robust vector equilibrium when U is a compact set. Conversely, it is not necessarily true. Although there is no parameter in the concept of the robust vector equilibrium with respect to the worst case, it still depends on the values of the parameter or sensitive to parameters. Now, we give the following example to illustrate the above cases.*

Example 1. *Consider a network problem with one O-D pair $\omega = (x, x')$. Two criteria, i.e., travel time and travel cost, and two available paths, i.e., $P_\omega = \{p_1, p_2\}$, with the travel demand $d_\omega = 30$.*

Assume that the path capacity constraints and cost function on the paths p_1 and p_2 are, respectively, given as follows:

$$l_{p_1} = 0,\ l_{p_2} = 0;\ u_{p_1} = 30,\ u_{p_2} = 30.$$

$$c_{p_1}(y,\xi_1) = \begin{pmatrix} y_1 + 2y_2 + \xi_1 \\ 6y_1 + 2y_2 + \xi_1 \end{pmatrix},\ c_{p_2}(y,\xi_2) = \begin{pmatrix} y_1 + 6y_2 \\ 6y_1 + 2y_2 - \xi_2 \end{pmatrix}.$$

with $\xi_1 \in [-1,2]$ and $\xi_2 \in [0,1]$.

Direct computation shows that $\bar{y} = (30,0)$ is the robust vector equilibrium. However, it is not the robust vector equilibrium with respect to the worst case since we have:

$$C_{p_1}(\bar{y}) = \begin{pmatrix} 32 \\ 182 \end{pmatrix},\ C_{p_2}(\bar{y}) = \begin{pmatrix} 30 \\ 180 \end{pmatrix},$$

but $y_{p_1} \neq l_{p_1}$ and $y_{p_2} \neq u_{p_2}$.

If $\xi_1 \in [-1,0]$ and $\xi_2 \in [-2,0]$, then we have $C_{p_2}(\bar{y}) - C_{p_1}(\bar{y}) \succeq 0_{\mathbb{R}^m}$, $y_{p_1} = u_{p_1}$ and $y_{p_2} = l_{p_2}$. Hence, $\bar{y} = (30,0)$ is the robust vector equilibrium with respect to the worst case. It can be seen that the robust vector equilibrium with respect to the worst case is sensitive to parameter perturbations.

3. Min–Max Method for Robust Vector Equilibrium

In this section, a min–max algorithm is proposed to look for a subset of the robust vector equilibrium flows.

3.1. Description of the Algorithm

In this subsection, we construct an optimization problem whose solution is equivalent to the the robust vector equilibrium flow. For $(y,\xi) \in \Omega \times U$, we define:

$$\psi(y,\xi) := \sum_{p_k,p_j \in P_\omega, \omega \in W} (y_{p_k} - l_{p_k})(u_{p_j} - y_{p_j})[c_{p_k}(y,\xi) - c_{p_j}(y,\xi)]^\top H_+[c_{p_k}(y,\xi) - c_{p_j}(y,\xi)].$$

Proposition 1. *Let $\bar{y}$ be a feasible flow. The following statements are equivalent.*

(i) *$\bar{y}$ is a robust vector equilibrium;*

(ii) *There exists $(\bar{y}, \bar{\xi})$ such that it is a saddle-point of the problem, denoted as follows:*

$$\begin{aligned} &\min_{y\in\Omega} \max_{\xi\in U}\ \psi(y,\xi) \\ &s.t.\ \ y \in \Omega. \end{aligned} \tag{2}$$

and $\psi(\bar{y},\bar{\xi})$ is equal to zero.

Proof. Firstly, we prove the implication $(i) \Rightarrow (ii)$. Since $\psi(y,\xi) \geq 0$, it suffices to prove $\psi(\bar{y},\xi) = 0$ for all $\xi \in U$, i.e., $0 = \psi(\bar{y},\xi) = \psi(\bar{y},\bar{\xi}) \leq \psi(y,\xi)$. Hence, for every $\xi \in U$, $p_k \in P_\omega$, $\omega \in \mathcal{W}$, we consider the following term:

$$O_{p_k} = \sum_{p_j \in P_\omega} (y_{p_k} - l_{p_k})(u_{p_j} - y_{p_j})[c_{p_k}(y,\xi) - c_{p_j}(y,\xi)]^\top H_+[c_{p_k}(y,\xi) - c_{p_j}(y,\xi)].$$

If $c_{p_k}(\bar{y},\xi) - c_{p_j}(\bar{y},\xi) \succeq 0_{\mathbb{R}^m}$ for some $p_j \in P_\omega$, then by Definition 2, one has $\bar{y}_{p_k} = l_{p_k}$ or $\bar{y}_{p_j} = u_{p_j}$ and so $O_{p_k} = 0$. If $c_{p_k}(\bar{y},\xi) - c_{p_j}(\bar{y},\xi) \prec 0_{\mathbb{R}^m}$, for some $p_j \in P_\omega$, $H_+[c_{p_k}(\bar{y},\xi) - c_{p_j}(\bar{y},\xi)] = 0_{\mathbb{R}^m}$, and hence, $O_{p_k} = 0$. By the above cases, one has $\psi(\bar{y},\xi) = 0$ for all $\xi \in U$.

Conversely, if (ii) is satisfied, $(\bar{y},\bar{\xi})$ is a saddle-point and $O_{p_k} = 0$. If for every $\xi \in U$, some $p_k,\ p_j \in P_\omega,\ \omega \in W$ one has $c_{p_k}(\bar{y},\xi) - c_{p_j}(\bar{y},\xi) \succeq 0_{\mathbb{R}^m}$, then $[c_{p_k}(\bar{y},\xi) - c_{p_j}(\bar{y},\xi)]^\top H_+[c_{p_k}(\bar{y},\xi) - c_{p_j}(\bar{y},\xi)] > 0$ and so $\bar{y}_{p_k} = l_{p_k}$ *or* $\bar{y}_{p_j} = u_{p_j}$. Consequently, $\bar{y}$ is a robust vector equilibrium. □

Now, a min–max algorithm is proposed to solve problem (2). In our algorithm, we select different steps for the two variables y and ξ, which is different from one proposed in [22]. In addition, we extend the search directions of the algorithm to make the search faster and more suitable for different needs. Thus, our algorithm is an improvement of that in [22].

Direction set: The set D consist of finite unit vectors which can span $\mathbb{R}^n$. Here, in order to reduce the computational cost, we only consider some directions in D. For example, when $n = 2$, in this paper, let $D = \Big\{(1,0), (\frac{\sqrt{3}}{2}, \frac{1}{2}), (1,1), (\frac{1}{2}, \frac{\sqrt{3}}{2}), (0,1), (-\frac{1}{2}, \frac{\sqrt{3}}{2}), (-1,1), (-\frac{\sqrt{3}}{2}, \frac{1}{2}), (-1,0), (-\frac{\sqrt{3}}{2}, -\frac{1}{2}), (-1,-1), (-\frac{1}{2}, -\frac{\sqrt{3}}{2}), (0,-1), (\frac{1}{2}, -\frac{\sqrt{3}}{2}), (1,-1), (\frac{\sqrt{3}}{2}, -\frac{1}{2})\Big\}$.

Step length: Let initial step $t_0 = 1$ and $d_k = \arg\min_{y_i} \psi_k(y_k + t_k d, \xi_k)$, $d \in D$. Let $\tilde{y}_k = y_k + t_k d_k$. If iteration is successful, i.e., $\psi_k(\tilde{y}_k, \xi_k)) < \psi_k(y_k + t_k d, \xi_k) - c t_k^2$ for all $d \in D$ ($c > 0$), then the next step length value $t_{k+1} = 1$; if the iteration is unsuccessful, then $t_{k+1} = \|\tilde{y} - y_k\|/2$.

Remark 2. *It is worth noting that computations for y_t and ξ_t in Algorithm 1 are based on Algorithm 2.*

Algorithm 1: Min–max algorithm (Algorithm 1).

input : ψ: objective function; c: forcing function constant $c > 0$;
T: maximum number of iterations; t_0: initialize step size;
(y_0, ξ_0): initial iteration point; $S = \varnothing$, $SE = \varnothing$.

1 **for** $t = 1, \cdots, T$ **do**
2     $\xi_t = \mathbf{A1}(-\psi(y_{t-1}, .), \xi_{t-1})$
3     $y_t = \mathbf{A1}(\psi(., \xi_t), y_{t-1})$
4   return (y_T, ξ_T), store it in S.
5 Choose a (y_T, ξ_T) from S, compute $\psi(y_T, \xi_T)$.
6 If $\psi(y_T, \xi_T) \leq \epsilon$, store y_T in SE and return to **Step 5** until no element of S left.

Algorithm 2: Algorithm 2 $(\psi(), y_0)$.

input : ψ: objective function; c: forcing function constant $c > 0$;
T: maximum number of iterations; t_0: initialize step size value;
y_0: initial iteration point.

for $k = 0, \cdots, T-1$ **do**
  1. Generate direction set
  $D = \{d^i$: any one unit direction of a certain point$\}$.
  2. Generate the points

$$y = y_k + t_k d \subset \Omega, \quad \forall\, d \in D.$$

  3. Choose $d_k = \arg\min_d \psi(y, \xi_k)$and let $\tilde{y}_k = y_k + t_k d_k$.
  4. **if** $\psi(\tilde{y}, \xi_k) < \psi(y_k, \xi_k) - c t_k^2$ **then**
    (Iteration is successful)
    $y_{k+1} = \tilde{y}, t_{k+1} = 1$;
  else
    (Iteration is unsuccessful)
    $y_{k+1} = y_k, t_{k+1} = \|\tilde{y} - y_k\|/2$.
  return y_T

3.2. Comparison with Other Methods

In this subsection, we will give three numerical examples to show the comparison with that of [22]. In these numerical examples, both Algorithm 1 and the algorithm proposed in [22] start from the same set of initial points. To make a fair comparison, all test problems are run five times to reduce the impact of randomness.

Remark 3. *There is a step calculation method in reference [22]—if the iteration is successful: $t_{k+1} = \min(t_{max}, \gamma t_k)$, $\gamma > 1$, where t_{max} is the largest step size; if iteration is unsuccessful: $t_{k+1} = \frac{1}{\gamma} t_k$. Compared with the step calculation method in reference [22], the step calculation method presented in Algorithm 1 has better performance, since Algorithm 1 selects different step size for different variables and extends the search directions. What is more, Algorithm 1 requires neither gradient information nor redundant parameters.*

Example 2. Consider a network problem depicted in Figure 1, where $\mathcal{N} = \{1,2\}$, $\mathcal{W} = \{\omega\} = \Big\{(1,2)\Big\}, \mathcal{E} = \{\alpha_1, \alpha_2\}, \mathcal{D} = d_\omega = 30$. There are two criteria: travel time and travel cost. The cost functions of arcs and constraints of paths are given as below: $t_{1,\alpha_1}(y,\xi) = y_1^2 + 2y_1y_2 + y_2 - \xi_1$, $t_{2,\alpha_1}(y,\xi) = y_1 + y_2^2$; $t_{1,\alpha_2}(y,\xi) = y_1^2 + 10y_2y_2$, $t_{2,\alpha_2}(y,\xi) = 7y_1 + 6y_2^2 - 6\xi_2$.

$$l_{p_1} = 0,\ l_{p_2} = 0;\ u_{p_1} = 30,\ u_{p_2} = 30.$$

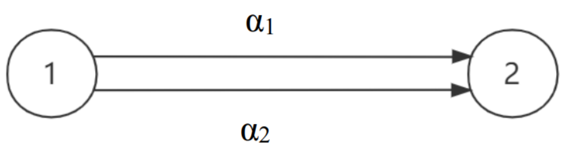

Figure 1. Network topology for Example 2.

Then, we have:

$$c_{p_1}(y,\xi) = \begin{pmatrix} y_1^2 + 2y_1y_2 + y_2 - \xi_1 \\ y_1 + y_2^2 \end{pmatrix},\ c_{p_2}(y,\xi) = \begin{pmatrix} y_1^2 + 10y_2y_2 \\ 7y_1 + 6y_2^2 - 6\xi_2 \end{pmatrix}.$$

where $\xi_1 \in [0,1]$ and $\xi_2 \in [0,1]$. Initial feasible flows and a subset of the robust vector equilibrium flows are obtained in 23.82*s*. The results are shown in Table 1. However, if we use step calculational method in [22], then it takes 25.68 s and the obtain the same robust vector equilibrium flows with our algorithm.

Table 1. Computational results of Algorithm 1.

Initial Feasible Flows	Robust Vector Equilibrium Flows
(0.00, 30.00) (3.75, 26.25)	(25.00, 5.00) (25.75, 4.25)
(11.25, 18.75) (11.25, 18.75)	(25.50, 4.50) (25.25, 4.75)
(15.00, 15.00) (18.75, 11.25)	(25.00, 5.00) (25.75, 4.25)
(22.50, 7.50) (26.25, 3.75)	(25.50, 4.50) (26.25, 3.75)
(30.00, 0.00)	(30.00, 0.00)

Example 3. *Consider the network problem depicted in Figure 2, where $\mathcal{N} = \{1,2,3,4\}$, $\mathcal{W} = \Big\{(1,4),(2,4)\Big\}$, and there are two O-D pairs, ω_1, ω_2. $\mathcal{E} = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5\}$, $p_1 = (\alpha_1\alpha_5)$, $p_2 = (\alpha_2)$, $p_3 = (\alpha_3\alpha_5)$, $p_4 = (\alpha_4)$, $\mathcal{D} = \{d_{\omega_1}, d_{\omega_2}\} = \{55, 35\}$. There are two criteria: travel time and travel cost. Constrains of paths are given as follows:*

$$l_{p_1} = 0,\ l_{p_2} = 0,\ l_{p_3} = 0,\ l_{p_4} = 0;$$
$$u_{p_1} = 55,\ u_{p_2} = 55,\ u_{p_3} = 35,\ u_{p_4} = 35.$$

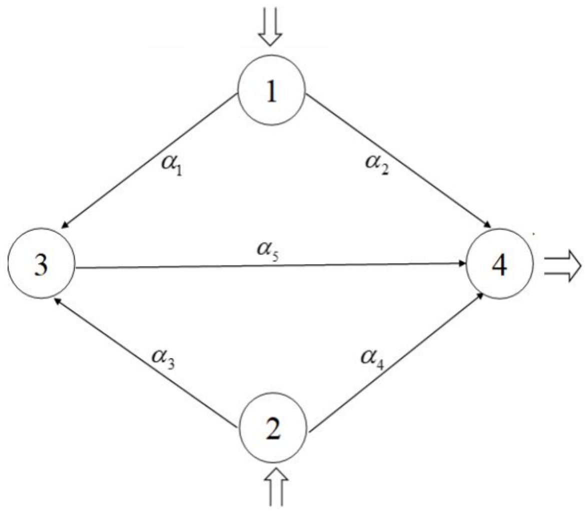

Figure 2. Network topology for Example 3.

The cost functions of arcs are defined as follows:

$$t_{\alpha_1}(y,\xi)=\begin{pmatrix} y_1^2+2y_2+y_3^2+y_4^2+1 \\ 2y_1+y_2^2+2y_3+y_4+\frac{3}{2}-6\xi_1 \end{pmatrix},\ t_{\alpha_2}(y,\xi)=\begin{pmatrix} 2y_1+3y_2+5y_3+y_4+1+\xi_2 \\ 2y_1y_2+y_2+y_3^2+y_4+\frac{1}{2} \end{pmatrix},$$

$$t_{\alpha_3}(y,\xi)=\begin{pmatrix} 2y_1+y_2+y_3^2+y_4+\frac{3}{2}+3\xi_3 \\ y_3^2+5y_4+1 \end{pmatrix},\ t_{\alpha_4}(y,\xi)=\begin{pmatrix} 2y_2+3y_3y_4+y_4^2+1 \\ y_1^2+y_2+y_4+\frac{1}{2}-8\xi_4 \end{pmatrix},\ t_{\alpha_5}(y,\xi)=\begin{pmatrix} y_1+y_2 \\ y_3+y_4 \end{pmatrix}.$$

Then, we have:

$$c_{p_1}(y,\xi)=\begin{pmatrix} y_1^2+y_1+3y_2+y_3^2+y_4^2+1 \\ 2y_1+y_2^2+3y_3+2y_4+1.5-6\xi_1 \end{pmatrix},\ c_{p_2}(y,\xi)=\begin{pmatrix} 2y_1+3y_2+5y_3+y_4+1+\xi_2 \\ 2y_1y_2+y_2+y_3^2+y_4+0.5 \end{pmatrix}$$

$$c_{p_3}(y,\xi)=\begin{pmatrix} 3y_1+2y_2+y_3^2+y_4+1.5+3\xi_3 \\ y_3^2+y_3+6y_4+1 \end{pmatrix},\ c_{p_4}(y,\xi)=\begin{pmatrix} 2y_2+3y_3y_4+y_4^2+1 \\ y_1^2+y_2+8y_4+0.5-8\xi_4 \end{pmatrix}$$

where $\xi_i \in [0,1], i=1,2,3,4$. Initial feasible flows and a subset of the robust vector equilibrium flows are obtained in 40.56 s. The results are shown in Table 2. The time cost of Algorithm 1 is 4% lower than that of the step calculation method in [22].

Table 2. Computational results of Algorithm 1.

Initial Feasible Flows	Robust Vector Equilibrium Flows
$(0.00, 55.00, 0.00, 35.00)$	$(27.00, 28.00, 27.00, 8.00)$
$(5.00, 50.00, 5.00, 30.00)$	$(27.00, 28.00, 27.00, 8.00)$
$(10.00, 45.00, 10.00, 25.00)$	$(27.00, 28.00, 27.00, 8.00)$
$(15.00, 40.00, 15.00, 20.00)$	$(27.00, 28.00, 27.00, 8.00)$
$(20.00, 35.00, 20.00, 15.00)$	$(27.00, 28.00, 27.00, 8.00)$
$(25.00, 30.00, 25.00, 10.00)$	$(29.00, 26.00, 29.00, 6.00)$
$(30.00, 25.00, 30.00, 5.00)$	$(29.00, 26.00, 29.00, 6.00)$
$(35.00, 20.00, 35.00, 0.00)$	

Example 4. *Consider the network problem depicted in Figure 3, where* $\mathcal{N}=\{1,2,3,4,5,6\}$, $\mathcal{W}=\{\omega_1,\omega_2\}=\Big\{(1,5),(2,6)\Big\}$, $\mathcal{E}=\{\alpha_1,\alpha_2,\alpha_3,\alpha_4,\alpha_5,\alpha_6,\alpha_7,\alpha_8,\alpha_9\}$, $\mathcal{D}=\{d_{\omega_1},d_{\omega_2}\}$, $d_{\omega_1}=25$, $d_{\omega_2}=20$, *with two criteria: travel time and travel cost.* $P_\omega=\{p_1,p_2,p_3,p_4,p_5,p_6,p_7\}$, *where* $P_{\omega_1}=\{p_1,p_2,p_3,p_4\}$, $P_{\omega_2}=\{p_5,p_6,p_7\}$, $p_1=(\alpha_3)$, $p_2=(\alpha_2\alpha_5\alpha_8)$, $p_3=(\alpha_1\alpha_4\alpha_5\alpha_8)$, $p_4=(\alpha_1\alpha_6\alpha_8)$, $p_5=(\alpha_7)$, $p_6=(\alpha_6\alpha_9)$, *and* $p_7=(\alpha_4\alpha_5\alpha_9)$.

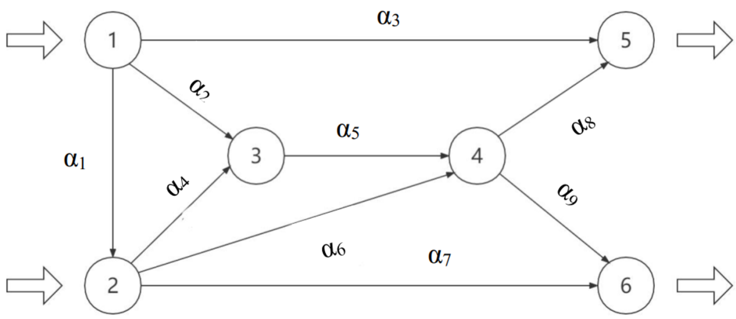

Figure 3. Network topology for Example 4.

The constrains of paths and cost functions are given as follows:

$$l_{p_1}=0,\ l_{p_2}=0,\ l_{p_3}=0,\ l_{p_4}=0,\ l_{p_5}=0,\ l_{p_6}=0,\ l_{p_7}=0;$$

$$u_{p_1}=25,\ u_{p_2}=25,\ u_{p_3}=25,\ u_{p_4}=25,\ u_{p_5}=20,\ u_{p_6}=20,\ u_{p_7}=20.$$

$$t_{1,\alpha_1}(y,\xi)=4(y_6+y_7)+50-2\xi_3,\ t_{2,\alpha_1}(y,\xi)=(y_6+y_7)^2+90-2\xi_3;$$
$$t_{1,\alpha_2}(y,\xi)=2y_2+20-\xi_2+4\xi_6,\ t_{2,\alpha_2}(y,\xi)=3y_2^2+10;$$
$$t_{1,\alpha_3}(y,\xi)=4y_1^2+100+\xi_1,\ t_{2,\alpha_3}(y,\xi)=2y_1^2+110+6\xi_1;$$
$$t_{1,\alpha_4}(y,\xi)=2(y_4+y_7)+10+\xi_2,\ t_{2,\alpha_4}(y,\xi)=(y_4+y_7)^2+30-\xi_2+\xi_3;$$
$$t_{1,\alpha_5}(y,\xi)=2(y_3+y_4+y_7)^2+10-\xi_2,\ t_{2,\alpha_5}(y,\xi)=(y_3+y_4+y_7)^2+20+\xi_2;$$
$$t_{1,\alpha_6}(y,\xi)=5(y_4+y_5)^2+430+2\xi_3+\xi_4,\ t_{2,\alpha_6}(y,\xi)=2(y_4+y_5)+530+2\xi_3-\xi_4;$$
$$t_{1,\alpha_7}(y,\xi)=2y_2^2+100+5\xi_5,\ t_{2,\alpha_7}(y,\xi)=3y_5+300;$$
$$t_{1,\alpha_8}(y,\xi)=(y_3+y_6+y_7)^2+20,\ t_{2,\alpha_8}(y,\xi)=2(y_3+y_6+y_7)^2+10;$$
$$t_{1,\alpha_9}(y,\xi)=(y_5+y_6)^2+30-4\xi_6,\ t_{2,\alpha_9}(y,\xi)=2(y_5+y_6)+10+2\xi_7-\xi_3;$$

where $z_{\alpha_i}(i=1,2,\ldots,9)$ denotes the flow on arc α_i. Then we have

$$c_{p_1}(y,\xi)=\begin{pmatrix}4y_1^2+100+\xi_1\\ 2y_1^2+110+6\xi_1\end{pmatrix},\ c_{p_2}(y,\xi)=\begin{pmatrix}2y_2+2(y_3+y_4+y_7)^2+(y_3+y_6+y_7)^2+50-2\xi_2\\ 3y_2^2+(y_3+y_4+y_7)^2+2(y_3+y_6+y_7)^2+40+\xi_2\end{pmatrix}$$

$$c_{p_3}(y,\xi)=\begin{pmatrix}4(y_6+y_7)+2(y_4+y_7)+2(y_3+y_4+y_7)^2+(y_3+y_6+y_7)^2+90-2\xi_3\\ (y_6+y_7)^2+(y_4+y_7)^2+(y_3+y_4+y_7)^2+2(y_3+y_6+y_7)^2+150-\xi_3\end{pmatrix}$$

$$c_{p_4}(y,\xi)=\begin{pmatrix}4(y_6+y_7)+5(y_4+y_5)^2+(y_3+y_6+y_7)^2+500+\xi_4\\ (y_6+y_7)^2+2(y_4+y_5)+2(y_3+y_6+y_7)^2+630-\xi_4\end{pmatrix}$$

$$c_{p_5}(y,\xi)=\begin{pmatrix}2y_2^2+100+5\xi_5\\ 3y_5+300\end{pmatrix},\ c_{p_6}(y,\xi)=\begin{pmatrix}5(y_4+y_5)^2+(y_5+y_6)^2+460-4\xi_6\\ 2(y_4+y_5)+2(y_5+y_6)+540\end{pmatrix}$$

$$c_{p_7}(y,\xi)=\begin{pmatrix}2(y_4+y_7)+2(y_3+y_4+y_7)^2+(y_5+y_6)^2+50\\ (y_4+y_7)^2+(y_3+y_4+y_7)^2+2(y_5+y_6)^2+60+2\xi_7\end{pmatrix}$$

where $\xi_i\in[0,1], i=1,2,3,4,5,6,7$. Initial feasible flows and a subset of the robust vector equilibrium flows are obtained in 538.47 s. The results are shown in Table 3. The algorithm proposed in [22] obtains the same robust vector equilibrium flows, but its time cost is 548.32 s.

Table 3. Computational results of Algorithm 1.

Initial Feasible Flows	Robust Vector Equilibrium Flows
$(0,0,0,25,0,0,20)$ $(0,0,25,0,0,0,20)$	$(9,7,9,0,19,0,1)$ $(12,13,0,0,8,0,12)$
$(0,25,0,0,0,0,20)$ $(25,0,0,0,0,0,20)$	$(15,10,0,0,9,0,11)$ $(15,10,0,0,9,0,11)$
$(0,0,0,25,0,20,0)$ $(0,0,25,0,0,20,0)$	$(12,8,5,0,15,0,5)$ $(15,10,0,0,9,0,11)$
$(0,25,0,0,0,20,0)$ $(25,0,0,0,0,20,0)$	$(9,7,9,0,19,0,1)$ $(12,13,0,0,8,0,12)$
$(0,0,0,25,20,0,0)$ $(0,0,25,0,20,0,0)$	$(11,7,7,0,18,0,2)$
$(0,25,0,0,20,0,0)$ $(25,0,0,0,20,0,0)$	

4. Smoothing Method for the Robust Vector Equilibrium with the Worst Case

It is worth noting that the algorithm in [21] needs to solve a non-smoothing optimization problem. This results in its computational inefficiency. This prompts us to continuously investigate algorithm for solving robust equilibrium flows. In this section, we propose a smoothing method to calculate a subset of the robust vector equilibrium with respect to the worst case. The algorithm is denoted Algorithm 3. To generalize a subset of the robust vector equilibrium flows with respect to the worst case, we use a two-step strategy. The first step is to construct an equivalent optimization problem with the help of a variant version of ReLU activation function for finding the robust weak vector equilibrium flows with respect to the worst case. The second step is to judge whether or not the robust weak vector equilibrium flows with respect to the worst case are equal to the robust vector equilibrium flows with respect to the worst case by an equivalent optimization problem using the vector version of heaviside step function.

Algorithm 3: Robust vector equilibrium algorithm (denoted Algorithm 3).

1 Choose a positive integer q and a tolerance level $\epsilon \geq 0$.

2 Enter $l = (l_{p_k})_{p_k \in P}$ and $u = (u_{p_k})_{P_k \in P}$. Set $\delta_j = d_{\omega_j} / (q|P_{\omega_j}|)$, $j = 1, \cdots, \tilde{l}$.

3 Choose $(k_1, \cdots, k_n)^\top \in \mathbb{N}^n$ satisfying

$$\sum_{i \in I_j} k_i = q|P_{\omega_j}|, \text{ and } \; l_{p_i} \leq k_i \delta_j \leq u_{p_i}, \; i \in I_j, \; j = 1, \cdots, \tilde{l}.$$

4 Store $y = (y_{p_1}, \cdots, y_{p_n})^\top$ in S^0 where

$$y_{p_i} = k_i \delta_j, \; i \in I_j, \; j = 1, \cdots, \tilde{I}$$

and return to **Step 3** for other vectors $(k_1, \cdots, k_n)$ unless no one left.

5 Choose a feasible flow y^0 from S^0 to start. Set $k = 0$, $S^0 = S^0 \setminus \{y^0\}$ and $WE = \varnothing$.

6 For every $i, j \in \{1, \cdots, n\}$, solve

$$\begin{aligned} \text{minimize} \;\; & \phi(y) \\ \text{subject to} \;\; & y \in \Omega \\ & \left| y_{p_i} - y^0_{p_i} \right| \leq \delta_{\omega(i)}, \; i = 1, \cdots, n. \end{aligned}$$

If $\phi(y) \leqq \epsilon$, store y in WE and return to **Step 5** until no element of S^0 left.

7 Choose a feasible flow $y \in WE$, $WE = WE \backslash \{y\}$.

8 Compute

$$\varphi(y) = \sum_{\omega \in W} \sum_{p_k, p_j \in P_\omega} (y_{p_k} - l_{p_k})(u_{p_j} - y_{p_j})(C_{p_k}(y) - C_{p_j}(y))^\top H_+[C_{p_k}(y) - C_{p_j}(y)]$$

9 If $\varphi(y) \leqq \epsilon$, store y in E and return to **Step 7** until no element of WE left.

Define a function $r : \mathbb{R} \to \mathbb{R}$ and give its vector version function $\mathcal{R} : \mathbb{R}^n \to \mathbb{R}^m$ as follows:

$$r(a) = \Big(max\{0, a\}\Big)^2.$$

$$\mathcal{R}(x) = (\prod_{i=1}^{n} r(x_i))e,$$

In addition, the heaviside step function $h_+ : \mathbb{R} \to \mathbb{R}$ and its vector version function $H_+ : \mathbb{R}^n \to \mathbb{R}^m$ are also given below:

$$h_+(a) = \begin{cases} 1, & \text{if } a \geq 0, \\ 0, & otherwise. \end{cases}$$

$$H_+(x) = (\prod_{i=1}^{n} h_+(x_i))e, \ \forall x \in \mathbb{R}^m.$$

4.1. Description of the Algorithm

In this subsection, we construct an optimization problem whose solution is equivalent to a robust weak vector equilibrium flow with respect to the worst case. For $y \in \Omega$, we define:

$$\phi(y) := \sum_{\omega \in W} \sum_{p_k, p_j \in P_\omega} (y_{p_k} - l_{p_k})(u_{p_j} - y_{p_j})(C_{p_k}(y) - C_{p_j}(y))^\top \mathcal{R}[C_{p_k}(y) - C_{p_j}(y)].$$

Proposition 2. *Let $\bar{y}$ be a feasible flow. The following statements are equivalent.*

(i) *$\bar{y}$ is a robust weak vector equilibrium with respect to the worst case;*

(ii) *$\bar{y}$ is an optimal solution of the problem, denoted:*

$$\begin{aligned} &\min \phi(\bar{y}) \\ &s.t. \ \ y \in \Omega. \end{aligned} \tag{3}$$

and the optimal value $\phi(\bar{y})$ is equal to zero.

Proof. We first prove the implication $(i) \Rightarrow (ii)$. It is not hard to see $\phi(y) \geq 0$ for every $y \in \Omega$. Thus, if $\bar{y}$ is a robust weak vector equilibrium with respect to the worst case, in order to deduce (ii), it suffices to prove $\phi(\bar{y}) = 0$. In addition, for every $p_k \in p_\omega$, $\omega \in W$, consider the term:

$$Q_p = \sum_{\omega \in \mathcal{W}} \sum_{p_k, p_j \in P_\omega} (y_{p_k} - l_{p_k})(u_{p_j} - y_{p_j})(C_{p_k}(y) - C_{p_j}(y))^\top \mathcal{R}[C_{p_k}(y) - C_{p_j}(y)].$$

If $C_{p_k}(\bar{y}) - C_{p_j}(\bar{y}) \succ 0_{\mathbb{R}^m}$, for some $p_j \in P_\omega$, then by Definition 4, either $\bar{y}_{p_k} = l_{p_k}$ or $\bar{y}_{p_j} = u_{p_j}$; if $C_{p_k}(\bar{y}) - C_{p_j}(\bar{y}) = 0_{\mathbb{R}^m}$, for some $p_j \in P_\omega$, we also get $Q_p = 0$; if $C_{p_k}(\bar{y}) - C_{p_j}(\bar{y}) \prec 0_{\mathbb{R}^m}$, then $\mathcal{R}[C_{p_k}(y) - C_{p_j}(y)] = 0_{\mathbb{R}^m}$, and thus, $Q_p = 0$. As a result, one has $\phi(\bar{y}) = 0$.

Conversely, assume that $\bar{y}$ is an optimal solution of Problem (3) and $\phi(\bar{y}) = 0$. Then, we have $Q_p = 0$ for all $p \in P$. If there exists some p_k, $p_j \in p_\omega$, $\omega \in \mathcal{W}$ such that $C_{p_k}(\bar{y}) - C_{p_j}(\bar{y}) \succ 0_{\mathbb{R}^m}$, then $(C_{p_k}(y) - C_{p_j}(y))'\mathcal{R}[C_{p_k}(y) - C_{p_j}(y) \succ 0_{\mathbb{R}^m}$, and thus, either $\bar{y}_{p_k} = l_{p_k}$ or $\bar{y}_{p_j} = u_{p_j}$ by $Q_p = 0$. Consequently, we deduce that $\bar{y}$ is a robust weak vector equilibrium with respect to the worst case. □

For $y \in \Omega$, we define

$$\varphi(y) = \sum_{\omega \in \mathcal{W}} \sum_{p_k, p_j \in P_\omega} (y_{p_k} - l_{p_k})(u_{p_j} - y_{p_j})(C_{p_k}(y) - C_{p_j}(y))^\top H_+[C_{p_k}(y) - C_{p_j}(y)].$$

Then, by using a similar method of proof, we may establish the following result for the robust vector equilibrium with respect to the worst case.

Proposition 3. *Let $\bar{y}$ be a feasible flow. The following statements are equivalent.*

(i) $\bar{y}$ *is a robust vector equilibrium with respect to the worst case.*
(ii) $\bar{y}$ *is an optimal solution of the problem, denoted as follows:*

$$\begin{aligned} &\min \varphi(y) \\ &s.t.\ \ y \in \Omega. \end{aligned} \tag{4}$$

and the optimal value $\varphi(\bar{y})$ *is equal to zero.*

Algorithm 3 is mainly based on ideas of Propositions 2 and 3. **Steps 1–4** create a subset of feasible flows with the initial conditions, denoted as S^0, with which **Steps 4–6** will start. **Steps 5–6** are aimed at solving Problem (3) given in Proposition 2 by using first-order optimization methods, and then a subset of the robust weak vector equilibrium flows with respect to the worst case is gained. **Steps 7–9** focus on solving Problem (4) given in Proposition 3, and then a subset of the robust vector equilibrium flows with respect to the worst case is generated.

Assume that $\mathcal{W}$ consists of $\tilde{l}$ elements $\omega_1, \ldots, \omega_{\tilde{l}}$ in the network and for each pair ω_i. Let $I_j = \left\{ i \in \{1, \ldots, n\} : p_i \in P_{w_j} \right\}$. Denote WE by the subset of the robust weak vector equilibrium flows with respect to the worst case and E by the subset of the robust vector equilibrium flows with respect to worst case.

4.2. Comparison with Other Methods

In this subsection, we will give two numerical examples to show the comparison with that of [21]. In these numerical examples, both Algorithm 3 and the algorithm proposed in [21] start from the same set of initial points. To make a fair comparison, all test problems are run five times to reduce the impact of randomness.

Example 5. *Consider the network problem depicted in Figure 4, where* $\mathcal{N} = \{1, 2, 3, 4, 5\}$, $\mathcal{W} = \{\omega_1, \omega_2\} = \left\{ (1,4), (1,5) \right\}$, $\mathcal{E} = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_8, \alpha_9\}$, $\mathcal{D} = \{d_{\omega_1}, d_{\omega_2}\}$, $d_{\omega_1} = 25, d_{\omega_2} = 20$, *with two criteria: travel time and travel cost,* $P_\omega = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7\}$, *where* $P_{\omega_1} = \{p_1, p_2, p_3, p_4\}$, $P_{\omega_2} = \{p_5, p_6, p_7\}$.

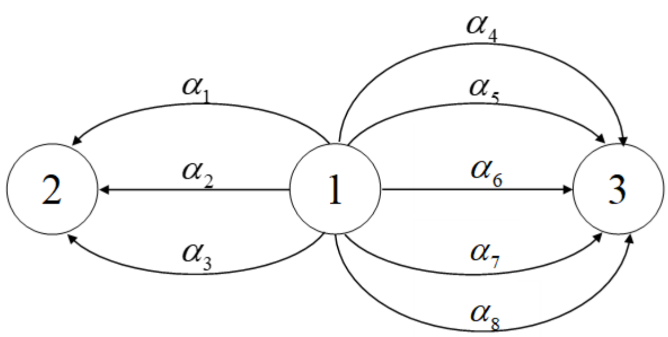

Figure 4. Network topology for Example 5.

Assume that:

$$l_{p_1} = l_{p_2} = l_{p_3} = l_{p_4} = l_{p_5} = l_{p_6} = l_{p_7} = l_{p_8} = 0;$$

$$u_{p_1} = 100,\ u_{p_2} = 100, u_{p_3} = 100, u_{p_4} = 200,\ u_{p_5} = 100, u_{p_6} = 120, u_{p_7} = 150, u_{p_8} = 100;$$

$$t_{1,\alpha_1}(y,\xi) = y_1^2 + y_2^2 + y_3^3 + \xi_1,\ t_{2,\alpha_1}(y,\xi) = 2y_1 + 5y_2 + 3y_3 + y_4 - 2\xi_1;$$
$$t_{1,\alpha_2}(y,\xi) = 8y_1y_2 + y_2^2 + y_7 + y_8 + 4\xi_2,\ t_{2,\alpha_2}(y,\xi) = y_2 + 10y_3 + 2y_7 + y_8 + 2\xi_2;$$
$$t_{1,\alpha_3}(y,\xi) = y_1 + y_2^2 + y_3^3 + y_5 + y_6 - 3\xi_3,\ t_{2,\alpha_3}(y,\xi) = 10y_3^3 + 2y_5 + \xi_3;$$
$$t_{1,\alpha_4}(y,\xi) = y_1 + y_2 + y_4^3 + y_5^2 + y_8^3 + \xi_4,\ t_{2,\alpha_4}(y,\xi) = y_1 + 2y_4 + y_6y_5 + 15y_8;$$
$$t_{1,\alpha_5}(y,\xi) = y_1 + y_3 + y_4^3 + y_5^2 + y_6^2,\ t_{2,\alpha_5}(y,\xi) = y_1 + 5y_3 + 5y_5 + 3y_6 + 12y_7 + 4\xi_5;$$
$$t_{1,\alpha_6}(y,\xi) = y_3 + y_4 + y_5 + y_6^3 - 3\xi_6,\ t_{2,\alpha_6}(y,\xi) = 3y_3 + 10y_5 + y_6 + 2y_8 - 2\xi_6;$$
$$t_{1,\alpha_7}(y,\xi) = y_2 + 8y_4^2 + y_5 + y_7^3 + 2\xi_7,\ t_{2,\alpha_7}(y,\xi) = y_1 + y_2 + 5y_4 + 3y_7;$$
$$t_{1,\alpha_8}(y,\xi) = y_1 + y_3 + 8y_6y_7 + y_8^2 + \xi_8,\ t_{2,\alpha_8}(y,\xi) = y_1 + y_3 + 10y_5^3 + y_8 - \xi_8,$$

where $\xi_i \in [0,1], i = 1,2,3,4,5,6,7,8$. Then, we have

$$C_{p_1}(y) = \begin{pmatrix} y_1^2 + y_2^2 + y_3^3 + 1 \\ 2y_1 + 5y_2 + 3y_3 + y_4 \end{pmatrix},\ C_{p_2}(y) = \begin{pmatrix} 8y_1y_2 + y_2^2 + y_7 + y_8 + 4 \\ y_2 + 10y_3 + 2y_7 + y_8 + 2 \end{pmatrix}$$
$$C_{p_3}(y) = \begin{pmatrix} y_1 + y_2^2 + y_3^3 + y_5 + y_6 \\ 10y_3^3 + 2y_5 + 1 \end{pmatrix},\ C_{p_4}(y) = \begin{pmatrix} y_1 + y_2 + y_4^3 + y_5^2 + y_8^3 + 1 \\ y_1 + 2y_4 + y_6y_5 + 15y_8 \end{pmatrix}$$
$$C_{p_5}(y) = \begin{pmatrix} y_1 + y_3 + y_4^3 + y_5^2 + y_6^2 \\ y_1 + 5y_3 + 5y_5 + 3y_6 + 12y_7 + 4 \end{pmatrix},\ C_{p_6}(y) = \begin{pmatrix} y_3 + y_4 + y_5 + y_6^3 \\ 3y_3 + 10y_5 + y_6 + 2y_8 \end{pmatrix}$$
$$C_{p_7}(y) = \begin{pmatrix} y_2 + 8y_4^2 + y_5 + y_7^3 + 2 \\ y_1 + y_2 + 5y_4 + 3y_7 \end{pmatrix},\ C_{p_8}(y) = \begin{pmatrix} y_1 + y_3 + 8y_6y_7 + y_8^2 + 1 \\ y_1 + y_3 + 10y_5^3 + y_8 \end{pmatrix}$$

Choosing $q = 2$, we have 32 feasible flows and 2 robust (weak) vector equilibrium flows with respect to the worst case, which are obtained in 0.18 s. Robust (weak) vector equilibrium flows with respect to the worst case are shown in Table 4. However, using the algorithm proposed in [21], it will take 13.826 s to obtain five robust vector equilibrium flows with respect to the worst case, which are shown in Table 5.

Table 4. Computational results of Algorithm 3.

Robust Weak Vector Equilibrium Flows (Worst Case)	Robust Vector Equilibrium Flows (Worst Case)
$(100, 100, 100, 145.125, 0, 120, 134.875, 100)$	$(100, 100, 100, 145.125, 0, 120, 134.875, 100)$
$(100, 100, 100, 150, 0, 120, 130, 100)$	$(100, 100, 100, 150, 0, 120, 130, 100)$

Table 5. Computational results of algorithm in [21].

Robust Vector Equilibrium Flows (Worst Case)
$(100, 100, 100, 30, 100, 120, 150, 100)$
$(100, 100, 100, 50, 100, 120, 130, 100)$
$(100, 100, 100, 100, 100, 120, 80, 100)$
$(100, 100, 100, 150, 100, 120, 30, 100)$
$(100, 100, 100, 200, 100, 100, 100, 0)$

Example 6. *Consider the network problem depicted in Figure 5, where* $\mathcal{N} = \{1,2,3,4,5\}$, $\mathcal{W} = \{\omega_1, \omega_2\} = \Big\{(1,4),(1,5)\Big\}, \mathcal{E} = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_8, \alpha_9\}, \mathcal{D} = \{d_{\omega_1}, d_{\omega_2}\}$ *with* $d_{\omega_1} = 25, d_{\omega_2} = 20$, *with two criteria: travel time and travel cost,* $P_\omega = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7\}$,

where $P_{\omega_1} = \{p_1, p_2, p_3, p_4\}$, $P_{\omega_2} = \{p_5, p_6, p_7\}$. $p_1 = (\alpha_4)$, $p_2 = (\alpha_2\alpha_8)$, $p_3 = (\alpha_2\alpha_6\alpha_3)$, $p_4 = (\alpha_1\alpha_3)$, $p_5 = (\alpha_5)$, $p_6 = (\alpha_2\alpha_9)$, $p_7 = (\alpha_1\alpha_7)$. *Let:*

$$l_1 = 0,\ l_2 = 0,\ l_3 = 0,\ l_4 = 0,\ l_5 = 0,\ l_6 = 0,\ l_7 = 0;$$

$$u_1 = 15,\ u_2 = 20,\ u_3 = 15,\ u_4 = 10,\ u_5 = 15,\ u_6 = 10,\ u_7 = 15.$$

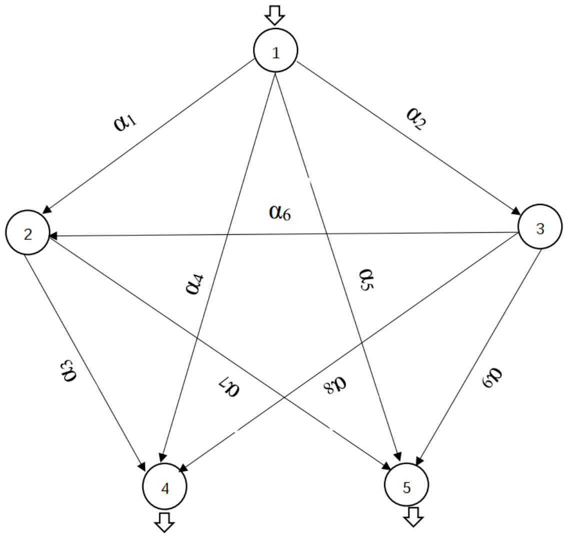

Figure 5. Network topology for Example 6.

Now, we give the cost function of arcs as follows:

$$t_{\alpha_1}(y,\xi) = \begin{pmatrix} 4(y_6+y_7) + 2(y_3+y_4+y_7)^2 + 300 - 2\xi_1 \\ (y_6+y_7)^2 + (y_3+y_4+y_7)^2 + 330 - \xi_1 \end{pmatrix},\ t_{\alpha_2}(y,\xi) = \begin{pmatrix} 2(y_3+y_4+y_7)^2 + 50 + \xi_2 \\ (y_3+y_4+y_7)^2 \end{pmatrix}$$

$$t_{\alpha_3}(y,\xi) = \begin{pmatrix} 2(y_3+y_7) + 2(y_4+y_7)^2 + 300 + 2\xi_1 + \xi_6 \\ (y_3+y_7)^2 + 2(y_4+y_7) + 100 + \xi_1 - \xi_3 \end{pmatrix}, t_{\alpha_4}(y,\xi) = \begin{pmatrix} 4y_1^2 + 100 + \xi_4 \\ 2y_1^2 + 110 + 6\xi_4 \end{pmatrix}$$

$$t_{\alpha_5}(y,\xi) = \begin{pmatrix} 2y_2^2 + 100 + 5\xi_5 \\ 3y_2 + y_5 + 300 \end{pmatrix},\ t_{\alpha_6}(y,\xi) = \begin{pmatrix} (y_3+y_6+y_7)^2 - 260 - \xi_6 - \xi_2 \\ 2(y_3+y_6+y_7)^2 + 50 - \xi_1 \end{pmatrix}$$

$$t_{\alpha_7}(y,\xi) = \begin{pmatrix} 2(y_3+y_7) + 2(y_4+y_7)^2 - 150 + 3\xi_1 - \xi_7 \\ (y_3+y_7)^2 + (y_4+y_7)^2 - 310 + \xi_1 \end{pmatrix}, t_{\alpha_8}(y,\xi) = \begin{pmatrix} 2y_2 + (y_3+y_6+y_7)^2 - \xi_2 \\ 3y_2^2 + 2(y_3+y_6+y_7)^2 + 40 - \xi_8 \end{pmatrix}$$

$$t_{\alpha_9}(y,\xi) = \begin{pmatrix} 5(y_5+y_6)^2 + 410 + \xi_2 \\ 2(y_5+y_6)^2 + 540 - \xi_9 \end{pmatrix}$$

where $\xi_i \in [0,1], i = 1,2,3,4,5,6,7,8,9$. Then, we have:

$$C_{p_1}(y) = \begin{pmatrix} 4y_1^2 + 101 \\ 2y_1^2 + 116 \end{pmatrix},\ C_{p_2}(y) = \begin{pmatrix} 2y_2 + 2(y_3+y_4+y_7)^2 + (y_3+y_6+y_7)^2 + 50 \\ 3y_2^2 + (y_3+y_4+y_7)^2 + 2(y_3+y_6+y_7)^2 + 41 \end{pmatrix},$$

$$C_{p_3}(y) = \begin{pmatrix} 4(y_3+y_7) + 2(y_4+y_7)^2 + 2(y_3+y_4+y_7)^2 + (y_3+y_6+y_7)^2 + 90 \\ (y_3+y_7)^2 + 2(y_4+y_7) + (y_3+y_4+y_7)^2 + 2(y_3+y_6+y_7)^2 + 150 \end{pmatrix},$$

$$C_{p_4}(y) = \begin{pmatrix} 4(y_3+y_7) + 2(y_4+y_7)^2 + 4(y_6+y_7) + 2(y_3+y_4+y_7)^2 + 601 \\ (y_3+y_7)^2 + 2(y_4+y_7) + (y_6+y_7)^2 + (y_3+y_4+y_7)^2 + 430 \end{pmatrix},$$

$$C_{p_5}(y) = \begin{pmatrix} 2y_2^2 + 105 \\ 3y_2 + y_5 + 300 \end{pmatrix},\ C_{p_6}(y) = \begin{pmatrix} 5(y_5 + y_6)^2 + 2(y_3 + y_4 + y_7)^2 + 460 \\ 2(y_5 + y_6) + (y_3 + y_4 + y_7)^2 + 540 \end{pmatrix},$$

$$C_{p_7}(y) = \begin{pmatrix} 4(y_6 + y_7) + 2(y_4 + y_7)^2 + 2(y_3 + y_4 + y_7)^2 + 2(y_3 + y_7) + 152 \\ (y_6 + y_7)^2 + (y_4 + y_7)^2 + (y_3 + y_4 + y_7)^2 + (y_3 + y_7)^2 + 20 \end{pmatrix}.$$

Choosing $q = 1$, 80 feasible flows are created and a subset of robust vector equilibrium flows with respect to the worst case (displayed in Tables 6) are obtained. This takes about 0.62 s. However, using the algorithm presented in [21], it spends 579 s to obtain a subset of robust vector equilibrium flows with respect to the worst case, which are shown in Table 7.

Table 6. Computational results of Algorithm 3.

Robust Weak Vector Equilibrium Flows (Worst Case)	Robust Vector Equilibrium Flows (Worst Case)
$(11.88, 11.29, 1.83, 0, 13.23, 0, 6.77)$	$(11.88, 11.29, 1.83, 0, 13.23, 0, 6.77)$
$(10.56, 12.5, 1.94, 0, 13.33, 0, 6.67)$	$(10.56, 12.5, 1.94, 0, 13.33, 0, 6.67)$
$(10.04, 14.92, 0.04, 0, 11.89, 0, 8.11)$	$(10.04, 14.92, 0.04, 0, 11.89, 0, 8.11)$
$(11.03, 12.4, 1.57, 0, 13.33, 0, 6.67)$	$(11.03, 12.4, 1.57, 0, 13.33, 0, 6.67)$
$(11.42, 13.58, 0, 0, 11.93, 0, 8.07)$	$(11.42, 13.58, 0, 0, 11.93, 0, 8.07)$
$(10.45, 12.5, 2.05, 0, 13.33, 0, 6.67)$	$(10.45, 12.5, 2.05, 0, 13.33, 0, 6.67)$
$(11.12, 10, 3.88, 0, 15, 0, 5)$	$(11.12, 10, 3.88, 0, 15, 0, 5)$

Table 7. Computational results of algorithm in [21].

Robust Vector Equilibrium Flows (Worst Case)
$(10.94, 12.5, 1.56, 0, 12.29, 0, 7.71)$
$(10.94, 13.28, 0.78, 0, 12.29, 0, 7.71)$
$(12.66, 10.78, 1.56, 0, 12.29, 0, 7.71)$
$(10.94, 12.5, 1.56, 0, 12.5, 0, 7.5)$

5. Conclusions

In this paper, we mainly consider a robust multi-criteria traffic network equilibrium problem with path capacity constraints. Firstly, the robust vector equilibrium principle and the robust vector equilibrium principle with respect to the worst case are given. We pay attention to constructing an equivalent min–max optimization problem for the robust vector equilibrium, in which the solution is equivalent to a robust vector equilibrium. Then, a direct search algorithm is proposed for solving the corresponding min–max optimization problem. The step size in the algorithm requires neither gradient information nor redundant parameters. What is more, we select different step sizes for different variables and extend the search directions. The results of three numerical experiments show that it takes less time than the method in [22] to find the robust vector equilibrium flows.

To generate a subset of the robust vector equilibrium with respect to the worst case, we employ a two-step strategy. The first step is to construct a smoothing optimization problem based on a variant version of the ReLU activation function to compute the robust weak vector equilibrium flows with respect to the worst case. The second step is to judge whether or not the robust weak vector equilibrium flows with respect to the worst case are equal to the robust vector equilibrium flows with respect to the worst case. Compared with the algorithm in [21], the results of two numerical experiments show that our algorithm can greatly reduce the computational cost.

Recently, robust vector optimization based on set orders is widely used in the uncertain optimization environment [23,24]. It is noteworthy that the robust vector equilibrium principles considered in this paper are all based on vector order. In addition, our method can only be applied to small-scale traffic networks. Therefore, an interesting topic for future research is to investigate large-scale, multi-criteria traffic networks based on set orders.

Author Contributions: Conceptualization, X.-X.M. and Y.-D.X.; Writing—original draft preparation, X.-X.M. and Y.-D.X.; Writing—review and editing, X.-X.M. and Y.-D.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Natural Science Foundation of China (Grant number: 11801051) and the Natural Science Foundation of Chongqing (Grant number: cstc2019jcyj-msxmX0075).

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wardrop, J. Some theoretical aspects of road traffic research. *Proc. Inst. Civil Eng. Part II* **1952**, *1*, 325–362. [CrossRef]
2. Athanasenas, A. Traffic simulation model for rural road network management. *Trans. Res. Part E* **1997**, *33*, 233–243. [CrossRef]
3. Nagurney, A. On the relationship between supply chain and transportation network equilibria: A super network equivalence with computations. *Trans. Res. Part E* **2006**, *42*, 293–316. [CrossRef]
4. Ji, X.F.; Chu, Y.Y. A target-oriented bi-attribute user equilibrium model with travelers perception errors on the tolled traffic network. *Trans. Res. Part E* **2020**, *144*, 102–150. [CrossRef]
5. Xu, Z.T.; Chen, Z.B.; Yin, Y.F.; Ye, J.P. Equilibrium analysis of urban traffic networks with ride-sourcing services. *Trans. Sci.* **2021**, *55*, 1227–1458. [CrossRef]
6. Wang, J.; Du, M.Q.; Lu, L.L.; He, X.Z. Maximizing network throughput under stochastic user equilibrium with elastic demand. *Netw. Spat. Econ.* **2018**, *18*, 115–143. [CrossRef]
7. Ma, J.; Meng, Q.; Cheng, L.; Liu, Z.Y. General stochastic ridesharing user equilibrium problem with elastic demand. *Trans. Res. Part B* **2022**, *162*, 162–194. [CrossRef]
8. Chen, G.Y.; Yen, N.D. *On the Variational Inequality Model for Network Equilibrium*; Internal Report 3; Department of Mathematics, University of Pisa: Pisa, Italy, 1993; Volume 196.
9. Yang, X.Q.; Goh, C.J. On vector variational inequalities: Application to vector equilibria. *J. Optim. Theory Appl.* **1997**, *95*, 431–443. [CrossRef]
10. Li, S.J.; Teo, K.L.; Yang, X.Q. A remark on a standard and linear vector network equilibrium problem with capacity constraints. *Eur. J. Oper. Res.* **2008**, *184*, 13–23. [CrossRef]
11. Luc, D.T.; Rocca,M.; Papalia, M. Equilibrium in a vector supply demand network with capacity. *Appl. Anal.* **2011**, *90*, 1029–1045. [CrossRef]
12. Raith, A.; Ehrgott, A. On vector equilibria, vector optimization and vector variational inequalities. *J. Multi-Crit. Decis. Anal.* **2011**, *18*, 39–54. [CrossRef]
13. Phuong, T.T.T.; Luc, D.T. Equilibrium in multi-criteria supply and demand networks with capacity constraints. *Math. Methods Oper. Res.* **2015**, *81*, 83–107.
14. Luc, D.T.; Phuong, T.T.T. Equilibrium in multi-criteria transportation networks. *J. Optim. Theory Appl.* **2016**, *169*, 116–147. [CrossRef]
15. Phuong, T.T.T. Smoothing method in multi-criteria transportation network equilibrium problem. *Optimization* **2019**, *68*, 1577–1598. [CrossRef]
16. Daniele, P.; Giuffré, S. Random variational inequalities and the random traffic equilibrium problem. *J. Optim. Theory Appl.* **2015**, *167*, 363–381. [CrossRef]
17. Dragicevic, A.; Gurtoo, A. Stochastic Control of Ecological Networks. *J. Math. Biol.* **2022**, *85*, 7. [CrossRef]
18. Ehrgott, M.; Wang, J.Y.T.; Watling, D.P. On multi-objective stochastic user equilibrium. *Trans. Res. Part B* **2015**, *81*, 704–717. [CrossRef]
19. Cao, J.D.; Li, R.X.; Huang, W.; Guo, J.H.; Wei, Y. Traffic network equilibrium problems with demands uncertainty and capacity constraints of arcs by scalarization approaches. *Sci. China Technol. Sci.* **2018**, *61*, 1642–1653. [CrossRef]
20. Wei, H.Z.; Chen, C.R.; Wu, B.W. Vector network equilibrium problems with uncertain demands and capacity constraints of arcs. *Optim. Lett.* **2021**, *15*, 1113–1131. [CrossRef]
21. Minh, N.B.; Phuong, T.T.T. Robust equilibrium in transportation networks. *Acta Math. Vietnam.* **2020**, *45*, 635–650. [CrossRef]
22. Dzahini, K.J. Expected complexity analysis of stochastic direct-search. *Comput. Optim. Appl.* **2022**, *81*, 1–22. [CrossRef]
23. Bouza, G.; Quintana, E.; Tammer, C. A steepest descent method for set optimization problems with set-valued mappings of finite cardinality. *J. Optim. Theory Appl.* **2021**, *190*, 711–743. [CrossRef]
24. Eichfelder, G.; Quintana, E.; Rocktäschel, S. A vectorization scheme for nonconvex set optimization problems. *SIAM J. Optim.* **2022**, *32*, 1184–1209. [CrossRef]