

Article

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

Xing-Xing Ma , Yang-Dong Xu *

Department of Mathematics, Chongqing University of Posts and Telecommunications, Chongqing, 400065, China; Maxx0205@126.com (Xing-Xing Ma)

* Correspondence: xyd04010241@126.com (Yang-Dong Xu)

Abstract: This paper focuses on robust multi-criteria traffic network equilibrium problems with path capacity constraints. The concepts of robust vector equilibrium and robust vector equilibrium with respect to worst case are introduced, respectively. To generate a subset of robust vector equilibrium, an equivalent min-max optimization problem is constructed. In addition, a direct search algorithm, in which the step size without derivatives and redundant parameters, for solving this min-max problem is proposed. To find a subset of robust vector equilibrium with respect to worst case, an equivalent smoothing optimization problem based on a variant version of ReLU activation function is established. Numerical experiments are also presented to expound the advantages of the algorithms.

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

1. Introduction

The equilibrium problem of traffic network is to describe the distribution of traffic flow in logistics industry and transportation networks, which is expected to provide an effective method to alleviate traffic congestion. The fundamental principle in the model is the concept of equilibrium that was initially introduced by Wardrop [1], which describes the user behavior in traffic networks. In other words, users will choose a path with the minimum travel cost. Now, it has been shown that Wardrop equilibrium concept is a powerful principle which widely used in supply and demand network, traffic assignment, optimization of traffic control and other fields and so on (see, e.g., Athanasesas, [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 user and traveler interchangeably) will naturally consider many criteria when choosing travel paths, including travel time, distance, cost, weather and safety other relevant factors. The equilibrium model ground on multi-criteria is first put forward by Chen and Yen [8], which is an extension of classical Wardrop user optimization principle. Regarding 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], Raith and Ehrgott[12] for further contributions. 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 method to handle multi-criteria network equilibrium problems. Subsequently, Luc and Phuong [14] put forward a modified Frank-Wolfe gradient algorithm for multi-criteria network equilibrium problems. After that, Phuong [15] proposed a smoothing method to solve multi-criteria network equilibrium problems. Numerical methods proposed in Phuong and Luc [13], Luc and Phuong [14] and Phuong [15], can be seen as a first step for the computation of multi-criteria network equilibrium problems.

Considering the influence of uncertain factors in traffic network on traveler behavior, data uncertainty in the logistics industry and transportation receives more and more attentions. Recently, some related works with uncertain demands or cost functions with parameters that are in a set, have been investigated in Daniele and Giuffr [16], Ehrgott et al.[17], Wang et al.[6], Cao et al.[18], Minh

and Phuong [19] and Wei et al.[20]. However, to the best of our knowledge, only one work [19] pays attention to the computation for robust equilibrium flows with uncertain cost functions. This prompts us to continuously investigate this topic.

In this paper, we give robust vector equilibrium principle and robust vector equilibrium principle with respect to worst case. 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 robust vector equilibrium with respect to worst case, we transform it into an equilibrium optimization problem based on a variant version of ReLU activation function.

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

(1) Robust vector equilibrium and robust vector equilibrium with respect to 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 robust vector equilibrium with respect to worst case, a two-step strategy is implemented. Specifically, construct a smoothing optimization problem based on a variant version of ReLU activation function to compute robust weak vector equilibrium flows with respect to worst case and then find robust vector equilibrium flows with respect to worst case by using Heaviside Step function.

This paper is divided into the following parts. Section 2 mainly introduces robust vector equilibrium principle and robust vector equilibrium principle with respect to 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 robust vector equilibrium principle with respect to worst case. Finally, conclusions of this paper and discussions for future research are provided.

2. Preliminaries

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

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

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

and the following stronger relation is given by

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

In the following, 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 2.1. Given $f : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$, 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 Equilibrium

For a traffic network, let \mathcal{N} denote the set of the nodes, \mathcal{E} denote the set of directed arcs. \mathcal{W} is the set of origin-destination O-D pairs, $\mathcal{D} = (d_\omega)_{\omega \in \mathcal{W}}$ is 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 , $P = \cup_{\omega \in \mathcal{W}} P_\omega$ is the set of all available

paths of the network. Let $n = \sum_{\omega \in \mathcal{W}} |p_\omega|$. For each path $p_k \in P_\omega$, y_{p_k} is the traffic flow on this path and $y = (y_1, y_2, \dots, y_n)^\top \in \mathbb{R}^n$ is called a path flow. Suppose $l_{p_k} \in \mathbb{R}_+$, $u_{p_k} \in \mathbb{R}_+$ with $l_{p_k} < u_{p_k}$, the path flow 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

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

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

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

Let $t_\alpha : \mathbb{R}^{|\mathcal{E}|} \rightarrow \mathbb{R}^m$ be a vector-valued cost function along with arc $\alpha \in \mathcal{E}$. Let $c_{p_k} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ 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 , i.e.,

$$c_{p_k}(y) = \sum_{\alpha \in \mathcal{E}} \delta_{\alpha p_k} t_\alpha(z). \quad (2.1)$$

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

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

Definition 2.2. A feasible flow $\bar{y} \in \Omega$ is said to be 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:

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

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

Definition 2.3. A feasible flow $\bar{y} \in \Omega$ is robust vector equilibrium with respect to 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 2.4. A feasible flow $\bar{y} \in \Omega$ is robust weak vector equilibrium with respect to 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 2.1. What should be noteworthy is that robust vector equilibrium with respect to worst case is also robust vector equilibrium when U is a compact set. Conversely, it is not necessarily true. Now, we give the following example to illustrate this case.

Example 2.1. Consider a network problem with one O-D pair $\omega = (x, x')$. Two criteria: travel time and travel cost, two available paths: $P_\omega = \{p_1, p_2\}$ 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.

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

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

with $\xi_1 \in [-1, 2]$ and $\xi_2 \in [0, 1]$. Direct computation shows that $\bar{y} = (30, 0)$ is robust vector equilibrium. However, it is not robust vector equilibrium with respect to worst case since we have

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

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

3. Min-Max Algorithm for Robust Vector Equilibrium

In this section, we propose a min-max algorithm to calculate a subset of robust vector equilibrium flows.

3.1. An Associated Optimization Problem

In this subsection, we construct an optimization problem whose solution is equivalent to 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 3.1. Let \bar{y} be a feasible flow. The following statements are equivalent.

- (i) \bar{y} is a robust vector equilibrium;
- (ii) There exists $(\bar{y}, \bar{\xi})$ is a saddle-point of the problem, denoted

$$\begin{aligned} \min_{y \in \Omega} \max_{\xi \in U} \psi(y, \xi) \\ \text{s.t. } y \in \Omega. \end{aligned} \tag{3.1}$$

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

Proof. Firstly, we prove 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., $0 = \psi(\bar{y}, \bar{\xi}) = \psi(\bar{y}, \xi) \leq \psi(y, \xi)$. Hence, for every $\xi \in U$, $p_k \in P_\omega$, $\omega \in W$, 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 $c_{p_k}(\bar{y}, \bar{\zeta}) - c_{p_j}(\bar{y}, \bar{\zeta}) \succeq 0_{\mathbb{R}^m}$ for some $p_j \in P_\omega$, then by definition $\bar{y}_{p_k} = l_{p_k}$ or $\bar{y}_{p_j} = u_{p_j}$, we get $O_{p_k} = 0$. If $c_{p_k}(\bar{y}, \bar{\zeta}) - c_{p_j}(\bar{y}, \bar{\zeta}) \prec 0_{\mathbb{R}^m}$, for some $p_j \in P_\omega$, $H_+[c_{p_k}(\bar{y}, \bar{\zeta}) - c_{p_j}(\bar{y}, \bar{\zeta})] = 0_{\mathbb{R}^m}$, hence $O_{p_k} = 0$. By the above cases, one has $\psi(\bar{y}, \bar{\zeta}) = 0$ for all $\bar{\zeta} \in U$.

Conversely. If (ii) is satisfied, say $(\bar{y}, \bar{\zeta})$ is a saddle-point and $O_{p_k} = 0$. If for every $\zeta \in U$, some $p_k, p_j \in P_\omega$, $\omega \in W$ one has $c_{p_k}(\bar{y}, \zeta) - c_{p_j}(\bar{y}, \zeta) \succeq 0_{\mathbb{R}^m}$, then $[c_{p_k}(\bar{y}, \zeta) - c_{p_j}(\bar{y}, \zeta)]^\top H_+[c_{p_k}(\bar{y}, \zeta) - c_{p_j}(\bar{y}, \zeta)] > 0$ 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. \square

3.2. Description of the Algorithm

In this subsection, a min-max algorithm is proposed to solve problem (3.1). In our algorithm, we select different steps for the two variables y and ζ , which is different from one proposed in [21]. 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 [21].

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

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

Algorithm 1: Min-Max Algorithm (Algorithm A)

input : ψ : objective function; c : forcing function constant $c > 0$;
 T : maximum number of iterations; t_0 : initialize step size;
 (y_0, ζ_0) : initial iteration point; $S = \emptyset$, $SE = \emptyset$.

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

Algorithm 2: Algorithm A1($\psi(\cdot), y_0$)

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.

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

$$y = y_k + t_k d \subset \Omega, \quad \forall d \in D.$$
 - 4 3. Choose $d_k = \arg \min_d \psi(y, \zeta_k)$ and let $\tilde{y}_k = y_k + t_k d_k$.
 - 5 4. **if** $\psi(\tilde{y}_k, \zeta_k) < \psi(y_k, \zeta_k) - ct_k^2$ **then**
 - 6 (Iteration is successful)
 - 7 $y_{k+1} = \tilde{y}_k, t_{k+1} = 1$;
 - 8 **else**
 - 9 (Iteration is unsuccessful)
 - 9 $y_{k+1} = y_k, t_{k+1} = \|\tilde{y} - y_k\|/2$.
 - 10 **return** y_T
-

Remark 3.1.

- (i) It is worth noting that computations for y_t and ξ_t in **Algorithm A** are based on **Algorithm A1**.
(ii) The step strategy given in [21] is as follows: If iteration is successful, then $t_{k+1} = \min(t_{max}, \gamma t_k)$, $\gamma > 1$, where t_{max} is the largest step size; Otherwise $t_{k+1} = \frac{1}{\gamma} t_k$. In the following Subsection 3.3, we will give three numerical examples to show that the step strategy in **Algorithm A** has better performance compared with that in [21].

3.3. Numerical Examples

In this subsection, there are some numerical examples with linear and nonlinear cost functions to illustrate **Algorithm A**. In these numerical examples, both **Algorithm A** and algorithm proposed in [21] start from the same set of initial points. To make a fair comparison, all test problems are ran 5 times to reduce the impact of randomness.

Example 3.1. Consider a network problem depicted in Figure 1, where $\mathcal{N} = \{1, 2\}$, $\mathcal{W} = \{\omega\} = \{(1, 2)\}$, $\mathcal{E} = \{\alpha_1, \alpha_2\}$, $\mathcal{D} = d_\omega = 30$. Two criteria: travel time and travel cost. The cost functions of arcs and constraints of paths are given as bellow: $t_{1,\alpha_1}(y, \xi) = y_1^2 + 2y_1y_2 + y_2 - \xi_1$, $t_{2,\alpha_1}(y, \xi) = y_1 + y_2^2$; $t_{1,\alpha_2}(y, \xi) = y_1^2 + 10y_2y_2$, $t_{2,\alpha_2}(y, \xi) = 7y_1 + 6y_2^2 - 6\xi_2$.

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

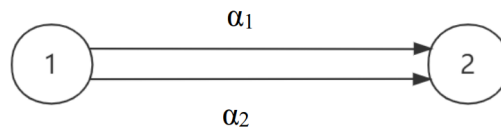


Figure 1. Network topology for Example 3.1.

Then, we have

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

where $\xi_1 \in [0, 1]$ and $\xi_2 \in [0, 1]$. Initial feasible flows and a subset of robust vector equilibrium flows are obtained in 23.12s. The results are shown in the Table 1. However, if we use step calculational method in [21], then it takes 25.86s and the obtain the same robust vector equilibrium flows with our algorithm.

Table 1. Computational results of **Algorithm A**.

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

$$l_{p_1} = 0, l_{p_2} = 0, l_{p_3} = 0, l_{p_4} = 0;$$

$$u_{p_1} = 55, u_{p_2} = 55, u_{p_3} = 35, u_{p_4} = 35.$$

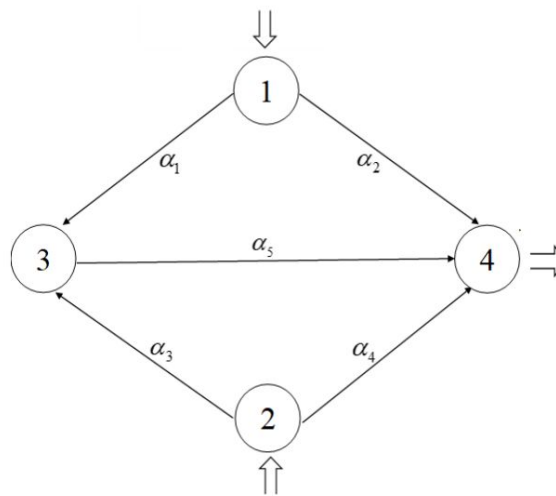


Figure 2. Network topology for Example 3.2.

The cost functions of arcs are defined as follows:

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

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

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 robust vector equilibrium flows are obtained in 40.56s. The results are shown in the Table 2. The time cost of **Algorithm A** is 4% lower than that of the step calculation method in [21].

Table 2. Computational results of **Algorithm A**.

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

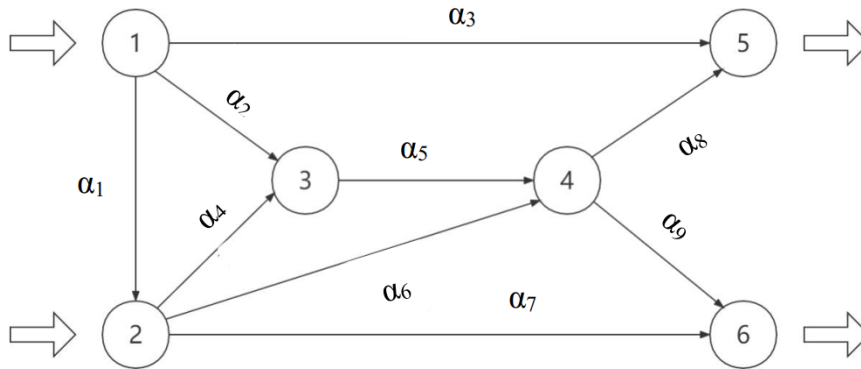


Figure 3. Network topology for Example 3.3.

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

$$\begin{aligned}
 l_{p_1} &= 0, l_{p_2} = 0, l_{p_3} = 0, l_{p_4} = 0, l_{p_5} = 0, l_{p_6} = 0, l_{p_7} = 0; \\
 u_{p_1} &= 25, u_{p_2} = 25, u_{p_3} = 25, u_{p_4} = 25, u_{p_5} = 20, u_{p_6} = 20, u_{p_7} = 20. \\
 t_{1,\alpha_1}(z_{\alpha_1}, \xi) &= 4z_{\alpha_1} + 50 - 2\xi_3, t_{2,\alpha_1}(z_{\alpha_1}, \xi) = z_{\alpha_1}^2 + 90 - 2\xi_3; \\
 t_{1,\alpha_2}(z_{\alpha_2}, \xi) &= 2z_{\alpha_2} + 20 - \xi_2 + 4\xi_6, t_{2,\alpha_2}(z_{\alpha_2}, \xi) = 3z_{\alpha_2}^2 + 10; \\
 t_{1,\alpha_3}(z_{\alpha_3}, \xi) &= 4z_{\alpha_3}^2 + 100 + \xi_1, t_{2,\alpha_3}(z_{\alpha_3}, \xi) = 2z_{\alpha_3}^2 + 110 + 6\xi_1; \\
 t_{1,\alpha_4}(z_{\alpha_4}, \xi) &= 2z_{\alpha_4} + 10 + \xi_2, t_{2,\alpha_4}(z_{\alpha_4}, \xi) = z_{\alpha_4}^2 + 30 - \xi_2 + \xi_3; \\
 t_{1,\alpha_5}(z_{\alpha_5}, \xi) &= 2z_{\alpha_5}^2 + 10 - \xi_2, t_{2,\alpha_5}(z_{\alpha_5}, \xi) = z_{\alpha_5}^2 + 20 + \xi_2; \\
 t_{1,\alpha_6}(z_{\alpha_6}, \xi) &= 5z_{\alpha_6}^2 + 430 + 2\xi_3 + \xi_4, t_{2,\alpha_6}(z_{\alpha_6}, \xi) = 2z_{\alpha_6} + 530 + 2\xi_3 - \xi_4; \\
 t_{1,\alpha_7}(z_{\alpha_7}, \xi) &= 2z_{\alpha_7}^2 + 100 + 5\xi_5, t_{2,\alpha_7}(z_{\alpha_7}, \xi) = 3z_{\alpha_7} + 300; \\
 t_{1,\alpha_8}(z_{\alpha_8}, \xi) &= z_{\alpha_8}^2 + 20, t_{2,\alpha_8}(z_{\alpha_8}, \xi) = 2z_{\alpha_8}^2 + 10; \\
 t_{1,\alpha_9}(z_{\alpha_9}, \xi) &= z_{\alpha_9}^2 + 30 - 4\xi_6, t_{2,\alpha_9}(z_{\alpha_9}, \xi) = 2z_{\alpha_9} + 10 + 2\xi_7 - \xi_3;
 \end{aligned}$$

where z_{α_i} ($i = 1, 2, \dots, 9$) denotes the flow on arc α_i .

Then we have

$$\begin{aligned}
 c_{p_1}(y, \xi) &= \begin{pmatrix} 4y_1^2 + 100 + \xi_1 \\ 2y_1^2 + 110 + 6\xi_1 \end{pmatrix}, c_{p_2}(y, \xi) = \begin{pmatrix} 2y_2 + 2(y_3 + y_4 + y_7)^2 + (y_3 + y_6 + y_7)^2 + 50 - 2\xi_2 \\ 3y_2^2 + (y_3 + y_4 + y_7)^2 + 2(y_3 + y_6 + y_7)^2 + 40 + \xi_2 \end{pmatrix} \\
 c_{p_3}(y, \xi) &= \begin{pmatrix} 4(y_6 + y_7) + 2(y_4 + y_7) + 2(y_3 + y_4 + y_7)^2 + (y_3 + y_6 + y_7)^2 + 90 - 2\xi_3 \\ (y_6 + y_7)^2 + (y_4 + y_7)^2 + (y_3 + y_4 + y_7)^2 + 2(y_3 + y_6 + y_7)^2 + 150 - \xi_3 \end{pmatrix} \\
 c_{p_4}(y, \xi) &= \begin{pmatrix} 4(y_6 + y_7) + 5(y_4 + y_5)^2 + (y_3 + y_6 + y_7)^2 + 500 + \xi_4 \\ (y_6 + y_7)^2 + 2(y_4 + y_5) + 2(y_3 + y_6 + y_7)^2 + 630 - \xi_4 \end{pmatrix}
 \end{aligned}$$

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

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

where $\xi_i \in [0, 1], i = 1, 2, 3, 4, 5, 6, 7$. Initial feasible flows and a subset of robust vector equilibrium flows are obtained in 538.47s. The results are shown in the Table 3. The algorithm proposed [21] obtain the same robust vector equilibrium flows but its time cost is 548.32s.

Table 3. Computational results of **Algorithm A**.

Initial feasible flows	Robust vector equilibrium flows
(0, 0, 0, 25, 0, 0, 20)	(0, 0, 25, 0, 0, 0, 20)
(0, 25, 0, 0, 0, 0, 20)	(25, 0, 0, 0, 0, 0, 20)
(0, 0, 0, 25, 0, 20, 0)	(0, 0, 25, 0, 0, 20, 0)
(0, 25, 0, 0, 0, 20, 0)	(25, 0, 0, 0, 0, 20, 0)
(0, 0, 0, 25, 20, 0, 0)	(0, 0, 25, 0, 20, 0, 0)
(0, 25, 0, 0, 20, 0, 0)	(25, 0, 0, 0, 20, 0, 0)

4. Smoothing Method for Robust Vector equilibrium with Worst Case

In this section, we propose a smoothing method to calculate a subset of robust vector equilibrium with respect to worst case. The algorithm is denoted **Algorithm B**. To generalize a subset of robust vector equilibrium flows with respect to 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 robust weak vector equilibrium flows with respect to worst case. The second step is to judge whether robust weak vector equilibrium flows with respect to worst case is robust vector equilibrium flows with respect to worst case by an equivalent optimization problem using the vector version of Heaviside step function.

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

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

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

In addition, the Heaviside Step function $h_+ : \mathbb{R} \rightarrow \mathbb{R}$ and its vector version function $H_+ : \mathbb{R}^n \rightarrow \mathbb{R}^m$ are also given below.

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

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

4.1. Associated Optimization Problems

In this subsection, we construct an optimization problem whose solution is equivalent to a robust weak vector equilibrium flow with respect to 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 4.1. Let \bar{y} be a feasible flow. The following statements are equivalent.
(i) \bar{y} is a robust weak vector equilibrium with respect to worst case;

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

$$\begin{aligned} & \min \phi(\bar{y}) \\ & \text{s.t. } y \in \Omega. \end{aligned} \quad (4.1)$$

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

Proof. We first prove 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 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, 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 so $Q_p = 0$. As a result, one has $\phi(\bar{y}) = 0$.

Conversely, assume that \bar{y} is an optimal solution of problem (4.1) 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 $(C_{p_k}(y) - C_{p_j}(y))^\top \mathcal{R}[C_{p_k}(y) - C_{p_j}(y)] \succ 0_{\mathbb{R}^m}$ and so 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 worst case. \square

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

Proposition 4.2. Let \bar{y} be a feasible flow. The following statements are equivalent.

(i) \bar{y} is a robust vector equilibrium with respect to worst case.

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

$$\begin{aligned} & \min \varphi(y) \\ & \text{s.t. } y \in \Omega. \end{aligned} \quad (4.2)$$

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

4.2. Description of the Algorithm

Algorithm B is mainly based on ideas of Proposition 4.1 and Proposition 4.2. **Steps 1 – 4** create a subset of feasible flows with the initial conditions, denoted S^0 , with which **Steps 4 – 6** will start. **Steps 5 – 6** are aim at solving problem (4.1) given in Proposition 4.1 by using first-order optimization methods, and then a subset of robust weak vector equilibrium flows with respect to worst case is gained. **Steps 7 – 9** focus on solving problem (4.2) given in Proposition 4.2, and then a subset of the robust vector equilibrium flows with respect to worst case is generated.

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

Algorithm 3: Robust Vector Equilibrium Algorithm (Denoted Algorithm B)

- 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 $\delta_j = d_{\omega_j} / (q|P_{\omega_j}|)$, $j = 1, \dots, \tilde{l}$.
- 3 Choose $(k_1, \dots, k_n)^\top \in \mathbb{N}^n$ satisfying

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

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

$$y_{p_i} = k_i \delta_j, \quad i \in I_j, \quad j = 1, \dots, \tilde{l}$$

and return to **Step 3** for other vectors (k_1, \dots, 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 $WE = \emptyset$.
- 6 For every $i, j \in \{1, \dots, n\}$, solve

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

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

- 7 Choose a feasible flow $y \in WE$, $WE = WE \setminus \{y\}$.
- 8 Compute

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

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

4.3. Numerical Examples

Example 4.1. Consider the network problem depicted in Figure 4, where $\mathcal{N} = \{1, 2, 3, 4, 5\}$, $\mathcal{W} = \{\omega_1, \omega_2\} = \{(1, 4), (1, 5)\}$, $\mathcal{E} = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_8, \alpha_9\}$, $\mathcal{D} = \{d_{\omega_1}, d_{\omega_2}\}$, $d_{\omega_1} = 25$, $d_{\omega_2} = 20$, two criteria: travel time and travel cost, $P_\omega = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7\}$, where $P_{\omega_1} = \{p_1, p_2, p_3, p_4\}$, $P_{\omega_2} = \{p_5, p_6, p_7\}$.

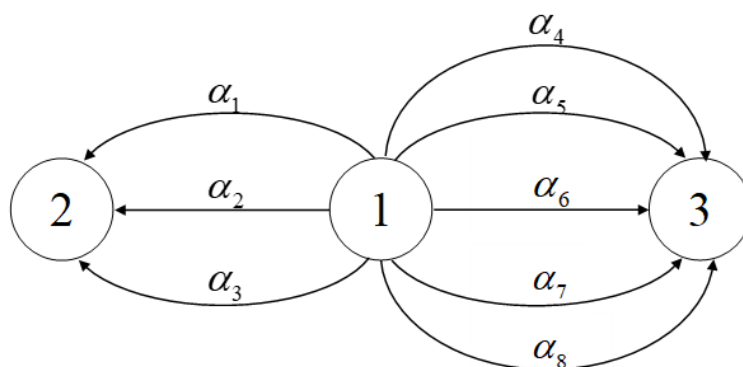


Figure 4. Network topology for Example 4.1.

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 worst case are obtained in 0.18s. Robust (weak) vector equilibrium flows with respect to worst case are shown in the Table 4. However, using algorithm proposed in [19], it will take 13.826s to obtain 5 robust vector equilibrium flows with respect to worst case which are shown in Table 5.

Table 4. Computational results of Algorithm B.

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 [19].

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 4.2. Consider the network problem depicted in Figure 5, where $\mathcal{N} = \{1, 2, 3, 4, 5\}$, $\mathcal{W} = \{\omega_1, \omega_2\} = \{(1, 4), (1, 5)\}$, $\mathcal{E} = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_8, \alpha_9\}$, $\mathcal{D} = \{d_{\omega_1}, d_{\omega_2}\}$ with $d_{\omega_1} = 25$, $d_{\omega_2} =$

20, 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

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

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

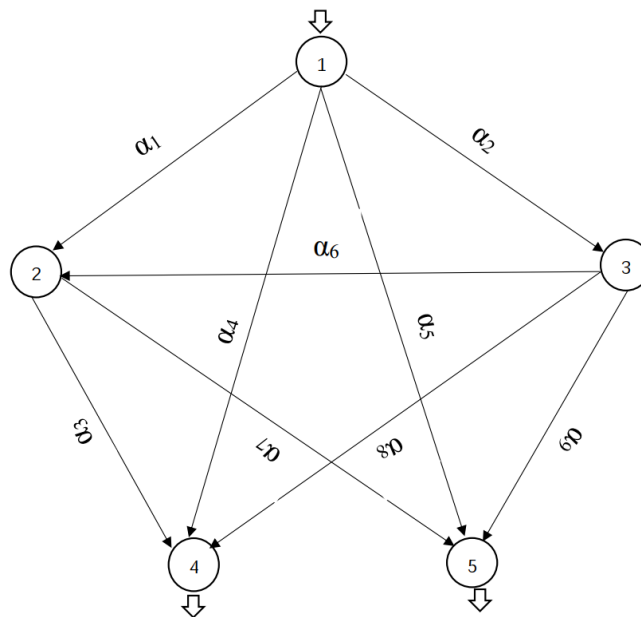


Figure 5. Network topology for Example 4.2.

Now, we given the cost function of arcs as follow:

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

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

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

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

$$t_{\alpha_9}(y, \xi) = \left(\frac{5(y_5 + y_6)^2 + 410 + \xi_2}{2(y_5 + y_6)^2 + 540 - \xi_9} \right)$$

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

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

$$\begin{aligned}
C_{p_3}(y) &= \left(\begin{array}{l} 4(y_3 + y_7) + 2(y_4 + y_7)^2 + 2(y_3 + y_4 + y_7)^2 + (y_3 + y_6 + y_7)^2 + 90 \\ (y_3 + y_7)^2 + 2(y_4 + y_7) + (y_3 + y_4 + y_7)^2 + 2(y_3 + y_6 + y_7)^2 + 150 \end{array} \right), \\
C_{p_4}(y) &= \left(\begin{array}{l} 4(y_3 + y_7) + 2(y_4 + y_7)^2 + 4(y_6 + y_7) + 2(y_3 + y_4 + y_7)^2 + 601 \\ (y_3 + y_7)^2 + 2(y_4 + y_7) + (y_6 + y_7)^2 + (y_3 + y_4 + y_7)^2 + 430 \end{array} \right), \\
C_{p_5}(y) &= \left(\begin{array}{l} 2y_2^2 + 105 \\ 3y_2 + y_5 + 300 \end{array} \right), \quad C_{p_6}(y) = \left(\begin{array}{l} 5(y_5 + y_6)^2 + 2(y_3 + y_4 + y_7)^2 + 460 \\ 2(y_5 + y_6) + (y_3 + y_4 + y_7)^2 + 540 \end{array} \right), \\
C_{p_7}(y) &= \left(\begin{array}{l} 4(y_6 + y_7) + 2(y_4 + y_7)^2 + 2(y_3 + y_4 + y_7)^2 + 2(y_3 + y_7) + 152 \\ (y_6 + y_7)^2 + (y_4 + y_7)^2 + (y_3 + y_4 + y_7)^2 + (y_3 + y_7)^2 + 20 \end{array} \right).
\end{aligned}$$

Choosing $q = 1$, 80 feasible flows are created and a subset of robust vector equilibrium flows with respect to worst case (displayed in Tables 6) are obtained. This takes about 0.62s. However, using algorithm presented in [19], it spends 579s to obtain a subset of robust vector equilibrium flows with respect to worst case which are shown in Table 7.

Table 6. Computational results of Algorithm B.

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 [19].

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. Concluding Remarks

In this paper, we mainly consider a robust multi-criteria traffic network equilibrium problem with path capacity constraints. Robust vector equilibrium principle and robust vector equilibrium principle with respect to worst case are given. We pay attention to constructing an equivalent min-max optimization problem for robust vector equilibrium, in which the solution is equivalent to a robust vector equilibrium flow. Then, a direct search algorithm is proposed for solving corresponding min-max optimization problem. To generate a subset of robust vector equilibrium with respect to worst case, we take a two-step strategy. The first step is to construct a smoothing optimization problem whose solutions are equivalent to robust weak vector equilibrium flows with respect to worst case. The second step is to judge whether robust weak vector equilibrium flows with respect to worst case is or not robust vector equilibrium flows with respect to worst case based on a optimization problem.

Recently, robust vector optimization based on set orders is widely used in the uncertain optimization environment [22,23]. It is noteworthy that the robust vector equilibrium principles considered in this paper are all based on vector order. An interesting topic for future research is to investigate this issue.

Acknowledgments: 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).

References

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

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.