



HAL
open science

Equilibria in the multi-criteria traffic networks

Thi Thanh Phuong Truong

► **To cite this version:**

Thi Thanh Phuong Truong. Equilibria in the multi-criteria traffic networks. Networking and Internet Architecture [cs.NI]. Université d'Avignon, 2015. English. NNT : 2015AVIG0413 . tel-01264132

HAL Id: tel-01264132

<https://theses.hal.science/tel-01264132>

Submitted on 28 Jan 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



ACADEMY OF AIX AND MARSEILLE
UNIVERSITY OF AVIGNON AND THE VAUCLUSE

THESIS

presented at University of Avignon and the Vaucluse
of the requirements for the degree of Doctor of Philosophy

SPECIALITY : Mathematics-Optimisation

Doctoral School (ED 536)

Laboratory of Mathematics of Avignon (EA 2151)

*Equilibria in the multi-criteria traffic
networks*

by

Thi Thanh Phuong TRUONG

Submitted publicly on May 26th 2015 in front of a committee composed by :

The Luc DINH (Avignon University)	Thesis Supervisor
Matthias EHRGOTT (Lancaster University)	Reviewer
Cuong LE-VAN (University of Paris I)	Reviewer
Pierre MARECHAL (Toulouse University Paul Sabatier)	Member
Michel THERA (Limoges University)	Member
Michel VOLLE (Avignon University)	Member

Contents

1	Introduction	1
2	Preliminaries	5
	2.0.1 Pareto minimal points	5
	2.0.2 Set-valued maps	6
	2.0.3 Variational inequality problem	7
	2.0.4 Increasing functions	8
3	Traffic network equilibrium	13
3.1	Single-criterion Traffic Network	13
	3.1.1 Wardrop's model	13
	3.1.2 Beckmann, McGuire and Winsten's model	13
	3.1.3 Michael Florian's model	14
3.2	Multi-criteria Traffic Network	16
	3.2.1 Description of multi-product multi-criteria traffic network	16
	3.2.2 Single-product multi-criteria supply demand network	18
	3.2.3 Multi-product single-criterion supply demand network	21
	3.2.4 Multi-product multi-criteria supply demand network	24
4	Equilibrium in a multi-criteria traffic network without capacity constraints	29
4.1	Equivalent problems	29
	4.1.1 Scalarization	30
	4.1.2 Vector variational inequalities	32
	4.1.3 Two optimization problems	33
4.2	Generic differentiability and local calmness of the objective functions	34
4.3	Generating vector equilibrium flows	41
	4.3.1 Description of the algorithm (A)	41
	4.3.2 Convergence of the algorithm	43
	4.3.3 Numerical examples	45
	4.3.4 Smoothing the objective function	49
4.4	Robust equilibrium	52
5	Equilibrium in a multi-criteria traffic network with capacity constraints	59
5.1	Single-product multi-criteria traffic network with capacity constraints	59
5.2	Equivalent optimization problem	60
5.3	Generic differentiability and local calmness of the objective function	61
5.4	Generating vector equilibrium flows	63

5.4.1	Description of the algorithm	63
5.4.2	Numerical examples	64
6	Equilibrium in a multi-product multi-criteria traffic network with capacity constraints	67
6.1	Existence conditions	67
6.2	Equivalent problems	69
6.2.1	Equilibrium with respect to a family of increasing functions	69
6.2.2	Efficiency	71
6.2.3	Variational inequality problems	72
6.3	Algorithms	74
6.4	Numerical examples	77
	Conclusion	81
	References	83
	Appendices	87
	Summary of the thesis in French	115

Introduction

In recent years multi-product multi-criteria supply demand networks have become a subject of intensive study. This is because such networks find abounding applications in several areas of applied sciences such as transport, internet communications, economics, management etc.

The idea of traffic equilibrium dates back to at least 1920 in the work of Pigou. He considered a model where there is only one origin and destination pair connected by two roads: the first one is short and narrow, the second one is wide and long. In the narrow and short road, travel time depends on the flow of vehicles on it. Meanwhile, in the wide and long road, travel time does not depend on the flow. Pigou argued that if the amount of vehicles is equal to the upper bound of capacity on the narrow road, the travel time for each driver on both roads is the same. If one of drivers diverts from the narrow road to the wide one to feel more comfortable in spite of spending the same travel time, then the drivers who remain using the narrow road will perceive a travel time reduction. The more drivers divert to the wide road, the less travel time the drivers remaining on the narrow road spend. However, in practice no driver will, altruistically, travel on a road that reduces his benefit in order to give a spontaneous situation for the network. According to Pigou's point of view, this calls for a State intervention in the form of a tax. Then we impose a tax on the narrow road, some vehicles will be turned away from it towards the wide road. However, all traffic participants will be indifferent with respect to the original situation, that means, the ones that still use the narrow road, despite experiencing a shorter travel time, will pay a tax that equivalent to such travel time reduction. This happens because, otherwise if the toll is greater than the time reduction the drivers would choose the wide road and, in the contrary, if the toll is smaller than the time reduction, some drivers will divert back to the narrow road. Hence, applying a toll policy on the narrow road leads to a situation in which the average cost of all participants in this network is equal; the only welfare difference between two situations with and without the tax is the amount of money collected by the tolls, which corresponds to the net gain to society.

The above mentioned model under conditions of congestions was studied by Knight in 1924 (Some fallacies in the Interpretation of Social cost-Quarterly Journal of Economics). We quote his simple and intuitively clear description of interaction between different users in the network:

"Suppose that between two points there are two highways, one of which is broad enough to accommodate without crowding all the traffic which may care to use it, but is poorly graded and surfaced, while the other is a much better road but narrow and quite limited in capacity. If a large number of trucks operate between the two termini and are free to choose either of the two routes, they will tend to distribute themselves between the roads in such proportions that the cost per unit of transportation, or effective return per unit of investment, will be the same for every truck on both routes. As more trucks use the narrower and better road,

congestion develops, until a certain point it becomes equally profitable to use the broader but poorer highway.”

This demonstrates the following principles of traffic distribution among alternative routes in equilibrium:

- (1) If between an origin and a destination there are at least two routes actually traveled, the average travel cost to each user must be equal on all these routes.
- (2) Since each driver attempts to choose the most convenient route, average cost on all other possible routes cannot be less than that on the route or routes traveled.
- (3) The amount of traffic on the network must equal the demand for transportation which prevails.

Some twenty-eight years later, Wardrop stated two principles: the first principle is identical to the notion of equilibrium described by Knight and the second one introduces the alternative behavior postulate of the minimization of total costs in the network. However, no mathematical model was proposed by Wardrop to describe the above ideas. In 1956 Beckmann, McGuire and Winsten [3] provided optimization reformulations of the governing equilibrium conditions, under a symmetry assumption on the underlying user link cost functions. Subsequently, in a lecture of Micheal Florian in 1984, he presented the elements of the network models used in transportation planning, reviewed their structural properties and most commonly used solution methods and outlined potential applications (see [19] for details). We notice that all these equilibrium models are based on scalar cost, which are not appropriate to describe real-world situations. Indeed, in practice the choice of paths by road users depends on several factors including for instance travel time, travel cost, comfort, safety and many others.

Multicriteria traffic network models, as a class of traffic network equilibrium problems, were first introduced by Quandt [50] and Schneider [54] in which both travel time and travel cost were explicitly considered. Further contributions are due to [12, 14, 29, 40, 43, 44] and [45]. In these works Wardrop’s traffic principle was defined for a weighted sum of the travel time and the travel cost, and therefore the analysis was presented under the angle of single-criterion models. Therefore, a model that takes into account different criteria is necessary to solve traffic network problems.

A vector version of Wardrop’s principle was first given by Chen et Yen [10] and subsequently developed by [9, 24, 61] (see also [11, 28, 34, 47, 57, 63]) for supply-demand networks without capacity constraints. Multi-criteria networks with capacity constraints have recently been studied by [32, 33, 37, 38] and [51]. Because of the multi-dimensionality of the cost space several generalizations of Wardrop’s principle have been introduced and their characterizations are given in terms of variational inequalities. There are two approaches to construct variational inequalities whose solutions may provide equilibrium flows of a multi-criteria network. The first approach is based on scalarization of the vector cost functions and leads to usual (scalar) variational inequality problems. Unfortunately, except for Luc and al. [38], all variational inequality problems in the above cited papers provide weak vector equilibrium flows only. The second approach constructs directly vector variational inclusions without converting the vector cost function to a scalar function. A major drawback of this approach as pointed out in Li and al. [32], is the fact that not every equilibrium can be obtained by solving the associated variational inequality problem. To overcome this defect the authors of [38] introduced the concept of elementary flows and derived a vector variational inequality problem over elementary flows which is equivalent to the network equilibrium problem. We notice that the concept of vector equilibrium treated in Li and al. [32] and Luc and al. [38] engages the products individually once a pattern flow is given. Other definitions of equilibrium, which take multi-product aspects into account, have been introduced in Luc [37]. Namely, this author considered three kinds of equilibrium: weak vector equilibrium,

strong vector equilibrium and ideal vector equilibrium, and constructed equivalent vector variational inequalities over elementary flows. In the above cited works on multi-criteria models we find a number of interesting theoretical results about weak and strong vector equilibria. However, as far as we know, a difficult question of how to compute vector equilibria or solutions of the associated vector variational inequality problems was not addressed.

The purpose of this thesis is to study equilibria in multi-criteria traffic networks and develop numerical methods to find them. The thesis is structured as follows. In the first chapter we present an introduction of the thesis. Chapter 2 is of preliminary character. We recall the concept of Pareto minimal points and some notions related to set-valued maps and variational inequality problem. We introduce some scalarizing functions, in particular the so-called augmented biggest/smallest monotone functions and augmented signed distance functions, and establish some properties we shall use later. Chapter 3 describes the traffic network models to be studied in this thesis. We define equilibrium for each model and determine a relationship between them. We also give some counter examples for some existing results in the recent literature on this topic. In Chapter 4 we develop a new solution method for multi-criteria network equilibrium problems without capacity constraints. To this end we shall construct two optimization problems the solutions of which are exactly the set of equilibria of the model, and establish some important generic continuity and differentiability properties of the objective functions. Then we give the formula to calculate the gradient of the objective functions which enables us to modify Frank-Wolfe's reduced gradient method to get descent direction toward an optimal solution. We prove the convergence of the method which generates a nice representative set of equilibria. Since the objective functions of our optimization problems are not continuous, a method of smoothing them is also considered in order to see how global optimization algorithms may help. We shall also introduce the concept of robust equilibrium, establish criteria for robustness and a formula to compute the radius of robustness. In Chapter 5 we consider vector equilibrium in the multi-criteria single-product traffic network with capacity constraints. We apply the approach of Chapter 4 to obtain an algorithm for generating equilibria of this network. In the last chapter we consider strong vector equilibrium in the multi-criteria multi-product traffic network with capacity constraints. We establish conditions for existence of strong vector equilibrium. We also establish relations between equilibrium and efficient points of the value set of the cost function and with equilibrium with respect to a family of functions. Moreover we exploit particular increasing functions discussed in Chapter 2 to construct variational inequality problems, solutions of which are equilibrium flows. The final part of this chapter is devoted to an algorithm for finding equilibrium flows of a multi-criteria network with capacity constraints. Some numerical examples are given to illustrate our method and its applicability. A list of references and appendices containing the code Matlab of our algorithms follow. We close up the thesis with a summary of main results in French.

Preliminaries

In this chapter we recall the concept of Pareto minimal points, the notions of continuity of set-valued maps and the variational inequality problem that we shall use throughout this thesis. We also propose some scalarizing functions including the augmented biggest/smallest monotone functions and the augmented signed distance functions, and establish some of their properties, which will be used to prove equivalence between vector equilibrium and scalarized equilibrium and to construct an equivalent scalar variational inequality problem for vector equilibrium. These functions will be amply employed in Chapter 6.

2.0.1 Pareto minimal points

In the space \mathbb{R}^n with $n > 1$ we distinguish the following order relations: strict inequality ' $<$ ' is understood as 'componentwise strictly smaller than', and ' \leq ' means 'componentwise smaller than or equal to' and not equal to. The binary relations ' $<$ ' and ' \leq ' are actually partial orders generated by the positive orthant \mathbb{R}_+^n of the space \mathbb{R}^n . Namely, for two vectors x and y from \mathbb{R}^n , one has $x \leq y$ (respectively $x < y$) if and only if $y - x \in \mathbb{R}_+^n \setminus \{0\}$ (respectively $y - x \in \text{int}\mathbb{R}_+^n$), where $\text{int}\mathbb{R}_+^n$ is the interior of \mathbb{R}_+^n . The relation ' \preceq ' means either ' \leq ' or ' $=$ '.

We notice that the partial orders in \mathbb{R}^n are not complete in the sense that not every couple of vectors is comparable, and hence the usual notion of minimum or maximum does not apply. Here we recall the notation of Pareto minimal points.

Definition 2.0.1 *Let Q be a nonempty set in \mathbb{R}^n . A point $y \in Q$ is said to be a (Pareto) minimal point of the set Q if there is no point $y' \in Q$ such that $y' \preceq y$ and $y' \neq y$. And it is said to be a Pareto weakly minimal point if there is no $y' \in Q$ such that $y' < y$.*

The sets of minimal points and weakly minimal points of Q are respectively denoted $\text{Min}(Q)$ and $\text{WMin}(Q)$. They are traditionally called the efficient and weakly efficient sets or the non-dominated and weakly non-dominated sets of Q . The set of maximal points $\text{Max}(Q)$ and weakly maximal points $\text{WMax}(Q)$ are called the efficient and weakly efficient sets of Q too. A set $Q \subset \mathbb{R}^n$ is called self-minimal if it coincides with the set of its Pareto-minimal points. If a set is self-minimal, it is self-maximal and vice versa. The terminology efficiency is advantageous in certain circumstances in which we deal simultaneously with minimal points of a set as introduced and minimal elements of a family of subsets which are defined to be minimal with respect to inclusion. In some situations one is interested in an ideal minimal point or utopia point which is defined as follows: If the infimum of Q , denoted by $\text{Inf}(Q)$, which is the vector whose i th component is the infimum of the projection of Q on the i th axis, is finite and belongs to the set Q , it is called the ideal minimal element of Q . In the other words, a point $y \in Q$ is called ideal minimal point if it satisfies

$$y \preceq y' \text{ for all } y' \in Q.$$

Such a point is generally not attainable, and if exists it is unique and denoted $\text{IMin}(Q)$.

Geometrically, a point y of Q is an efficient (minimal) point if the intersection of the set Q with the negative orthant shifted at y consists of y only, that is

$$Q \cap (y - \mathbb{R}_+^n) = \{y\}$$

and it is weakly minimal if the intersection of Q with the interior of the negative orthant shifted at y is empty, that is

$$Q \cap (y - \text{int}\mathbb{R}_+^n) = \emptyset.$$

Of course, minimal points are weakly minimal, and the converse is not true in general. We refer to [16, 39] for more details of the theory of Pareto optimality.

2.0.2 Set-valued maps

Let X and Y be metric spaces and $F : X \rightrightarrows Y$ be a set-valued map. The domain of F is the set

$$\text{dom}(F) := \{x \in X \mid F(x) \neq \emptyset\},$$

and the graph of F is the subset of the product space $X \times Y$ defined by

$$\text{Graph}(F) := \{(x, y) \in X \times Y \mid y \in F(x)\}.$$

We recall some definitions of continuity of set-valued maps.

Definition 2.0.2 *The map F is called upper semi-continuous at $x \in \text{dom}(F)$ if for any neighborhood U of $F(x)$, there exists $\eta > 0$ such that*

$$F(x') \subset U \quad \forall x' \in B_X(x, \eta),$$

where $B_X(x, \eta)$ is the ball of radius η , centered at x . It is said to be upper semi-continuous if it is upper semi-continuous at any point of $\text{dom}(F)$.

The map F is called lower semi-continuous at $x \in \text{dom}(F)$ if for any $y \in F(x)$ and for any sequence of elements $x_n \in \text{dom}(F)$ converging to x , there exists a sequence of elements $y_n \in F(x_n)$ converging to y .

It is said to be lower semi-continuous if it is lower semi-continuous at every point x of $\text{dom}(F)$.

When F is both upper semi-continuous and lower semi-continuous at x , we say that it is continuous at x , and it is continuous if it is so at every point of $\text{dom}(F)$.

We note that when F is single valued, upper semi-continuity and lower semi-continuity signify continuity. Moreover, the following equivalent definition of lower semi-continuity is also in use: for any open subset $U \subset Y$ such that $U \cap F(x) \neq \emptyset$, there exists $\eta > 0$ such that $F(x') \cap U \neq \emptyset$ for every $x' \in B_X(x, \eta)$. We shall use the following results in [2] (Proposition 1.4.8, Theorem 1.4.13 and Theorem 1.4.16 respectively).

Proposition 2.0.3 *Let X, Y be metric spaces. The graph of an upper semi-continuous set-valued map $F : X \rightrightarrows Y$ with closed domain and closed values is closed. The converse is true if we assume that Y is compact.*

Theorem 2.0.4 (Generic Continuity) *Let F be a set valued-map from a complete metric space X to a complete separable metric space Y .*

- i) If F is upper semi-continuous, it is continuous on a countable intersection of dense open subsets $A_n \subset X$.
- ii) If F is lower semi-continuous with compact values, it is continuous on a countable intersection of dense open subsets $A_n \subset X$.
- iii) If F is upper semi-continuous with closed values, then there exists a countable intersection R of dense open subsets $A_n \subset X$ such that

$$\forall x \in R, \limsup_{x' \rightarrow x} F(x') = F(x).$$

Theorem 2.0.5 (*Maximum Theorem*) Let metric spaces X, Y , a set valued-map $F : X \rightrightarrows Y$ and a function $f : \text{Graph}(F) \mapsto \mathbb{R}$ be given. If f and F are lower semi-continuous (respectively upper semi-continuous), the function $g : X \mapsto \mathbb{R} \cup \{+\infty\}$ defined by

$$g(x) := \sup_{y \in F(x)} f(x, y)$$

is also lower semi-continuous (respectively upper semi-continuous).

2.0.3 Variational inequality problem

Let K be a closed convex set in \mathbb{R}^n and F a continuous function from K to \mathbb{R}^n . The finite-dimensional variational inequality problem, denoted $\text{VI}(F, K)$, is to determine a vector $x^* \in K \subseteq \mathbb{R}^n$, such that

$$\langle F(x^*)^T, x - x^* \rangle \geq 0, \forall x \in K$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product in the n -dimensional Euclidean space \mathbb{R}^n . The following existence result is known.

Theorem 2.0.6 (*Existence under compactness and continuity*) If K is a compact convex set and F is continuous from K to \mathbb{R}^n , then the variational inequality problem admits at least one solution x^* .

Proof. Let P_K be a projection onto the set K . Since P_K and $(I - \gamma F)$ are each continuous, $P_K(I - \gamma F)$ is also continuous. According to Brouwer's Fixed Point Theorem there is at least one $x^* \in K$ such that $x^* = P_K(I - \gamma F)(x^*)$. Then

$$\langle x^{*T}, x - x^* \rangle \geq \langle (x^* - \gamma F(x^*))^T, x - x^* \rangle \quad \forall x \in K,$$

and therefore,

$$\langle F(x^*)^T, y - x^* \rangle \geq 0 \quad \forall y \in K.$$

□

For the convergence of numerical algorithms a monotonicity property of F is needed.

Definition 2.0.7 F is monotone on K if

$$\langle (F(x^1) - F(x^2))^T, x^1 - x^2 \rangle \geq 0, \forall x^1, x^2 \in K$$

and it is strictly monotone on K if

$$\langle (F(x^1) - F(x^2))^T, x^1 - x^2 \rangle > 0, \forall x^1, x^2 \in K, x^1 \neq x^2.$$

Under the strict monotonicity the problem $\text{VI}(F, K)$ admits at most one solution.

Theorem 2.0.8 (*Uniqueness*) Suppose that F is strictly monotone on K , then the solution is unique, if one exists.

Proof. Suppose that x^1 and x^* are both solutions and $x^1 \neq x^*$. Then since both x^1 and x^* are solutions, they must satisfy

$$\langle F(x^1)^T, x' - x^1 \rangle \geq 0, \forall x' \in K, \quad (2.1)$$

$$\langle F(x^*)^T, x' - x^* \rangle \geq 0, \forall x' \in K. \quad (2.2)$$

After substituting x^* for x' in (2.1) and x^1 for x' in (2.2), and adding the resulting inequalities, one obtains:

$$\langle F(x^1) - F(x^*)^T, x^* - x^1 \rangle \geq 0. \quad (2.3)$$

But inequality (2.3) is in contradiction to the definition of strict monotonicity. Hence, $x^1 = x^*$. \square

Details on this subject are found in [43].

2.0.4 Increasing functions

Functions that are increasing with respect to the partial orders in \mathbb{R}^n play an important role in the study of vector optimization problems.

Definition 2.0.9 Let P be a nonempty subset of \mathbb{R}^n . A real function $f : P \rightarrow \mathbb{R}$ is said to be increasing (respectively weakly increasing) if for every $a, b \in P$,

$$a \geq b \text{ (respectively } a > b) \Rightarrow f(a) > f(b). \quad (2.4)$$

Notice that an increasing function is weakly increasing, but the converse is not true. It is clear that the set of increasing (respectively weakly increasing) functions is a convex cone without apex. In particular, the sum of two increasing functions is increasing and the sum of two weakly increasing functions is weakly increasing. Notice further that the sum of a weakly increasing function and an increasing function is weakly increasing, but not necessarily increasing (see Example 2.0.10 below).

Example 2.0.10 Consider the function $g : \mathbb{R}_+^2 \rightarrow \mathbb{R}$ defined by

$$g(x) = \begin{cases} x_1 + x_2 + 2 & , (x_1, x_2) > (0, 0); \\ \frac{1}{x_1 + 1} & , x_1 > 0, x_2 = 0; \\ \frac{1}{x_2 + 1} & , x_1 = 0, x_2 > 0. \end{cases}$$

Then the function g is weakly increasing, but not continuous on \mathbb{R}_+^2 . It is not difficult to see that $f + g$ is not necessarily increasing for any increasing function f on \mathbb{R}_+^2 .

Here is an exception in which the sum of an increasing function and a weakly increasing function is increasing.

Lemma 2.0.11 If g is a continuous, weakly increasing function and f is an increasing function on P , then the sum function $f + g$ is increasing on P . Consequently, every continuous, weakly increasing function is a pointwise limit of a sequence of increasing functions.

Proof. Let $a, b \in P$ and $a \geq b$. Let e be a strictly positive vector. We have $a + te > b$ for every real number $t > 0$. Since g is weakly increasing, $g(a + te) > g(b)$ for every $t > 0$. Due to the continuity of g , we deduce $g(a) \geq g(b)$. This together with the monotonicity of f implies, consequently, $(f + g)(a) = f(a) + g(a) > f(b) + g(b) = (f + g)(b)$ proving that $f + g$ is increasing.

Now given a continuous, weakly increasing function g , we choose any increasing function f (for instance $f(x) = x_1 + \dots + x_n$ for $x = (x_1, \dots, x_n) \in \mathbb{R}^n$) and put $f_k = g + f/k$. Then for every $x \in \mathbb{R}^n$, we have $g(x) = \lim_{k \rightarrow \infty} f_k(x)$ with f_k increasing. Thus, g is the pointwise limit of the sequence of $f_k, k \geq 1$. \square

Now we present some weakly increasing and increasing functions frequently used in vector optimization (see [23, 27, 39, 48]) which we shall use in our thesis.

Biggest and smallest weakly increasing functions. Let e the vector of ones in \mathbb{R}_+^n and $a = (a_1, \dots, a_n) \in \mathbb{R}^n$. For every $x \in \mathbb{R}^n$ define

$$g_a(x) = \max\{t : x \in a + te + \mathbb{R}_+^n\} = \min_{i=1, \dots, n} (x_i - a_i).$$

We are also interested in a counter part of this function when using $-\mathbb{R}_+^n$ instead of \mathbb{R}_+^n :

$$G_a(x) = \min\{t : x \in a + te - \mathbb{R}_+^n\} = \max_{i=1, \dots, n} (x_i - a_i).$$

These two functions are both continuous weakly increasing, but not increasing. They are called respectively biggest and smallest continuous weakly increasing functions because of the following property: For every continuous weakly increasing function g on \mathbb{R}^n with $g(a) = 0$, one has

$$\{x \in \mathbb{R}^n : G_a(x) < 0\} \subseteq \{x \in \mathbb{R}^n : g(x) < 0\} \subseteq \{x \in \mathbb{R}^n : g_a(x) < 0\},$$

that is the strict level set at 0 of G_a is the smallest and the strict level set of g_a is the biggest among the strict level sets at 0 of continuous weakly increasing functions taking the zero value at a .

Signed distance functions. Let A be a nonempty set in \mathbb{R}^n . The signed distance function (see [27]) Δ_A is defined by

$$\Delta_A(x) = \rho(x, A) - \rho(x, A^c),$$

where $\rho(x, A)$ is the distance from x to A , and A^c is the complement of A . In other words,

$$\Delta_A(x) = \begin{cases} -\rho(x, A^c) & \text{if } x \in A; \\ \rho(x, A) & \text{if } x \in A^c. \end{cases}$$

The particular case of this function when A is either the negative or the positive orthant of the space, is frequently used in vector optimization. Namely, let $a \in \mathbb{R}^n$ be given. Define

$$d_a(x) = \Delta_{(\mathbb{R}_+^n)^c}(x - a) \text{ and } D_a(x) = \Delta_{-\mathbb{R}_+^n}(x - a).$$

Then

$$d_a(x) \leq g_a(x) \leq G_a(x) \leq D_a(x).$$

We notice that $d_a(\cdot)$ and $D_a(\cdot)$ are continuous weakly increasing functions on \mathbb{R}^n with $d_a(a) = 0$ and $D_a(a) = 0$. The following inclusions are clear

$$\{x \in \mathbb{R}^n : G_a(x) < 0\} = \{x \in \mathbb{R}^n : D_a(x) < 0\} \subseteq \{x \in \mathbb{R}^n : g_a(x) < 0\},$$

and

$$\{x \in \mathbb{R}^n : G_a(x) < 0\} \subseteq \{x \in \mathbb{R}^n : d_a(x) < 0\} = \{x \in \mathbb{R}^n : g_a(x) < 0\}.$$

Let us characterize the partial order ' $>$ ' by weakly increasing functions.

Lemma 2.0.12 *Let a and b be two vectors in \mathbb{R}^n . The following assertions are equivalent.*

- (i) $a > b$;
- (ii) $D_u(a) > D_u(b)$ for every $u \in \mathbb{R}^n$;
- (iii) $d_u(a) > d_u(b)$ for every $u \in \mathbb{R}^n$;
- (iv) $0 > D_a(b)$;
- (v) $0 < d_b(a)$.

The above assertions are also true if we replace D by G and d by g .

Proof. We prove equivalence between (i) and (iv). The others equivalences are proven similarly. If $a > b$, then $0 > D_a(b)$ since because the function D_a is weakly increasing and $D_a(a) = 0$. For the converse, $D_a(b) < 0$ implies that $b - a \in -\mathbb{R}_+^n$ and $\rho(b - a, (-\mathbb{R}_+^n)^C) \neq 0$, which means that $a > b$. For the functions G and g , the proof is similar. \square

Let us characterize the partial order ' \succeq ' by weakly increasing functions, but in a more complicated way.

Lemma 2.0.13 *Let a and b be two vectors in \mathbb{R}^n . The following assertions are equivalent.*

- (i) $a \geq b$;
- (ii) $D_u(a) \geq D_u(b)$ for every $u \in \mathbb{R}^n$ and $a \neq b$;
- (iii) $d_u(a) \geq d_u(b)$ for every $u \in \mathbb{R}^n$ and $a \neq b$;
- (iv) $D_b(a) > 0 \geq D_a(b)$;
- (v) $d_b(a) \geq 0 > d_a(b)$.

The above assertions are also true if we replace D by G and d by g .

Proof. As in Lemma 2.0.12 we establish equivalence between (i) and (iv). If $a \geq b$, then $b - a \in -\mathbb{R}_+^n$, which implies that $D_a(b) = -\rho(b - a, (-\mathbb{R}_+^n)^C) \leq 0$ and $D_b(a) = \rho(a - b, -\mathbb{R}_+^n) > 0$. For the converse, we observe that $D_a(b) \leq 0$ implies that either $b \leq a$ or $a = b$, while $D_b(a) > 0$ implies $a \neq b$. By this $a \geq b$. For the functions G and g , the proof is similar. \square

Now we will make use of the following "small" affine increasing function in which ϵ is a strictly positive number:

$$f_a^\epsilon(x) = \epsilon \sum_{i=1}^n (x_i - a_i)$$

and apply Lemma 2.0.11 to obtain the following increasing functions which are called respectively augmented biggest/smallest functions and augmented signed distance functions:

$$\begin{aligned} g_a^\epsilon(x) &= g_a(x) + f_a^\epsilon(x) \\ G_a^\epsilon(x) &= G_a(x) + f_a^\epsilon(x) \\ d_a^\epsilon(x) &= d_a(x) + f_a^\epsilon(x) \\ D_a^\epsilon(x) &= D_a(x) + f_a^\epsilon(x) \end{aligned}$$

which pointwisely converge respectively to $g_a(x)$, $G_a(x)$, $d_a(x)$ and $D_a(x)$ as ϵ tends to 0. The function G_a was used in [11, 32] and some others to find weak vector equilibrium. Like g_a , d_a and D_a , it is weakly increasing but not increasing, hence is not suitable for finding strong vector equilibrium. The function g_a^ϵ was already known, see for instance [37, 38]. To our knowledge the functions d_a^ϵ , G_a^ϵ and D_a^ϵ are given here for the first time. As we will see, they have very nice properties that make them crucial in finding strong vector equilibrium. They may also be very useful in the study of multi-criteria decision making and vector

optimization, particularly in generating the efficient solution set of a vector problem and in establishing its structure by scalarization. Below are some properties of these functions for our use.

Lemma 2.0.14 *Let a and b be two vectors in \mathbb{R}^n . The following assertions hold.*

- (i) $a \geq b$ if and only if $d_b^\epsilon(a) > 0$ for every $\epsilon > 0$.
- (ii) $a \not\geq b$ if and only if there is $\epsilon(a, b) > 0$ such that $d_b^\epsilon(a) \leq 0$ for all $0 < \epsilon < \epsilon(a, b)$.

The above assertions are also true for $g_b^\epsilon(a)$.

Proof. If $a \geq b$, then $d_b^\epsilon(a) > 0$ for all $\epsilon > 0$ because the function d_b^ϵ is increasing and $d_b^\epsilon(b) = 0$. For the converse let $a \not\geq b$. If $a \leq b$, then $d_b^\epsilon(a) \leq 0$ for every $\epsilon > 0$ because d_b^ϵ is increasing. If $a \not\leq b$, then either $\sum_{i=1}^n (a_i - b_i) \leq 0$ or $\sum_{i=1}^n (a_i - b_i) > 0$.

In the first case,

$$d_b^\epsilon(a) := -\rho(a - b, \mathbb{R}_+^n) + \epsilon \sum_{i=1}^n (a_i - b_i) \leq 0 \text{ for every } \epsilon > 0$$

because $a - b \notin \mathbb{R}_+^n$.

In the last case, set

$$\epsilon(a, b) = \frac{\rho(a - b, \mathbb{R}_+^n)}{\sum_{i=1}^n (a_i - b_i)} > 0.$$

Then, for $0 < \epsilon < \epsilon(a, b)$, we have $d_a^\epsilon(b) = -\rho(a - b, \mathbb{R}_+^n) + \epsilon \sum_{i=1}^n (a_i - b_i) < 0$.

The second assertion is obtained from the first assertion by using the proof above. For $g_b^\epsilon(\cdot)$, the proof is similar. \square

We note that the assertion (ii) of Lemma 2.0.14 is a modified version of the negation of (i). The first assertion applied to $g_b^\epsilon(\cdot)$ is a correction of Lemma 4.8 of [38] (the proof given there is correct) and consequently Corollary 4.9 of that paper must be reformulated in a similar manner.

Lemma 2.0.15 *Let a and b be two vectors in \mathbb{R}^n . The following assertions hold.*

- (i) $a \geq b$ if and only if $D_a^\epsilon(b) < 0$ for every $\epsilon > 0$.
- (ii) $a \not\geq b$ if and only if there is $\epsilon(a, b) > 0$ such that $D_a^\epsilon(b) \geq 0$ for all $0 < \epsilon < \epsilon(a, b)$.
- (iii) For every $\epsilon > 0$, one has $D_a^\epsilon(b) + D_b^\epsilon(a) \geq 0$. In particular, if $D_a^\epsilon(b) \leq 0$, then $D_b^\epsilon(a) \geq 0$.

The above assertions are also true for $G_a^\epsilon(\cdot)$.

Proof. For (i) let $a \geq b$. Then $D_a^\epsilon(b) < 0$ because the function D_a^ϵ is increasing and $D_a^\epsilon(a) = 0$. For the converse, suppose $a \not\geq b$. If $a \leq b$, then $D_a^\epsilon(b) \geq 0$ for every $\epsilon > 0$ because D_a^ϵ is increasing. If $a \not\leq b$, then either $\sum_{i=1}^n (b_i - a_i) \geq 0$ or $\sum_{i=1}^n (b_i - a_i) < 0$.

In the first case,

$$D_a^\epsilon(b) := \rho(b - a, -\mathbb{R}_+^n) + \epsilon \sum_{i=1}^n (b_i - a_i) > 0 \text{ for every } \epsilon > 0$$

because $b - a \notin -\mathbb{R}_+^n$.

In the last case, set

$$\epsilon(a, b) = \frac{-\rho(b - a, -\mathbb{R}_+^n)}{\sum_{i=1}^n (b_i - a_i)} > 0.$$

Then, for $0 < \epsilon < \epsilon(a, b)$, we have $D_a^\epsilon(b) = \rho(b - a, -\mathbb{R}_+^n) + \epsilon \sum_{i=1}^n (b_i - a_i) > 0$.

The second assertion is obtained from the first assertion and from the proof above. Now we prove the last assertion. By definition,

$$D_a^\epsilon(b) = \rho(b - a, -\mathbb{R}_+^n) - \rho(b - a, (-\mathbb{R}_+^n)^C) + \epsilon \sum_{i=1}^n (b_i - a_i),$$

and

$$D_b^\epsilon(a) = \rho(a - b, -\mathbb{R}_+^n) - \rho(a - b, (-\mathbb{R}_+^n)^C) + \epsilon \sum_{i=1}^n (a_i - b_i).$$

Then

$$\begin{aligned} & D_a^\epsilon(b) + D_b^\epsilon(a) \\ &= \rho(b - a, -\mathbb{R}_+^n) + \rho(a - b, -\mathbb{R}_+^n) - [\rho(b - a, (-\mathbb{R}_+^n)^C) + \rho(a - b, (-\mathbb{R}_+^n)^C)]. \end{aligned}$$

If $a = b$, then $D_a^\epsilon(b) + D_b^\epsilon(a) = 0$. If $b - a \leq 0$, then $a - b \in (-\mathbb{R}_+^n)^C$, and hence $D_a^\epsilon(b) + D_b^\epsilon(a) = \rho(a - b, -\mathbb{R}_+^n) - \rho(b - a, (-\mathbb{R}_+^n)^C) = \|a - b\| - \rho(b - a, (-\mathbb{R}_+^n)^C) \geq \|a - b\| - \|b - a\| \geq 0$. If $b - a \not\leq 0$, then $D_a^\epsilon(b) + D_b^\epsilon(a) = \rho(b - a, -\mathbb{R}_+^n) + \rho(a - b, -\mathbb{R}_+^n) - \rho(a - b, (-\mathbb{R}_+^n)^C) \geq 0$. For $G_a^\epsilon(\cdot)$, the proof is similar. \square

Note that some other interesting properties of the augmented signed functions such as Lipschitz continuity, quasi-convexity etc. can also be established, but we do not give them in details here because they will not directly be used in the present work.

Traffic network equilibrium

In this chapter we focus on scalar equilibrium and concepts of vector equilibrium in the existing literature and establish some relationships between them. We point out some misunderstandings and inadequacies of certain results in recent works on vector equilibrium.

3.1 Single-criterion Traffic Network

3.1.1 Wardrop's model

Consider a traffic network where there is an origin-destination (O/D for short) pair w connected by m alternative routes named p_1, p_2, \dots, p_m . We denote the set of these paths by P . Let $Y = (y_{p_i})_{p_i \in P}$ denote a flow of traffic where y_{p_i} is the quantity of drivers following the route p_i . Suppose that there are d_w drivers transporting on the O/D pair w . Then we say that a flow Y is feasible if it satisfies the following conditions:

$$y_{p_i} \geq 0, p_i \in P \text{ and } \sum_{p_i \in P} y_{p_i} = d_w.$$

In this model, drivers only pay attention to travel time. The distribution in the network is based on the following two principles:

- 1) The travel time on all routes actually used is equal, and less than those which would be experienced by a single vehicle on any unused route.
- 2) The average travel time is minimum.

The first principle is quite a likely one in practice, since it might be assumed that traffic will tend to settle down into an equilibrium situation in which no driver will want to choose an alternate route. In this case, we will say the system is at a user equilibrium state. This principle has been considered as a sound and simple behavioral principle to describe the spreading of traffics over alternate routes due to congested conditions. On the other hand, the second principle is the most efficient in the sense that it minimizes the vehicle-hours spent on the network, when this goal is achieved we will say that the system is at a social optimum state.

3.1.2 Beckmann, McGuire and Winsten's model

Although Wardrop discussed the equilibrium conditions for a general network, he did not propose any method to compute the corresponding flows. Soon after, the first mathematical model of traffic equilibrium on a network was formulated and analyzed by Beckmann,

McGuire and Winsten [3], which was the starting point for the contributions to follow this area. Their model was described as follows: Let xx' be a road on the network and $y_{xx'}$ be the number of vehicles entering road xx' from either end per unit of time, briefly called the flow on that road. However, the elementary variable will be the flow on a road in a given direction to a particular destination, written $y_{xx',k}$, where the order pair of subscripts xx' denotes the direction from x to x' on road xx' , and k denotes the destination. This flow is distinct from that in the opposite direction and it does not admit of negative values:

$$y_{xx',k} \geq 0, \text{ for all } xx', k.$$

The total flow on a road equals

$$y_{xx'} = y_{x'x} = \sum_k (y_{xx',k} + y_{x'x,k}), \quad (3.1)$$

and of course

$$y_{xx'} \geq 0.$$

The number of vehicles originating at location x with destination k per unit of time is denoted by $y_{x,k}$. Since this rate of origination is indicated at x by the excess of the flow to k on outgoing roads over that on incoming roads we have

$$y_{x,k} = \sum_{x'} (y_{xx',k} - y_{x'x,k}). \quad (3.2)$$

The transportation cost c on road xx' is denoted $c_{xx'}$. Since we do not distinguish between costs in the two directions, we have $c_{xx'} = c_{x'x}$. For points x and x' that are not contiguous, $c_{xx'}$ is left undefined. Travel costs from origin x to destination k are denoted by $c_{x,k}$ (notice that the subscripts are separated by a comma). Now

$$c_{x,k} = \min(c_{xx_1} + c_{x_1x_2} + c_{x_2x_3} + \dots + c_{x_nk}),$$

that is, the minimum of all chain sums of $c_{x_ix_j}$ starting at x and terminating at k in which consecutive elements have one subscript in common. In particular $c_{x,x} = 0$.

Consider $c_{r,k}$ for two locations $r = x$ and $r = x'$ connected by a road xx' . Extending the minimum chain that leads from x' to k by adding $c_{xx'}$, we have a chain from x to k , but not necessarily a minimum chain. Thus

$$c_{x,k} \leq c_{xx'} + c_{x',k}.$$

Then equilibrium flow is determined by

$$c_{x,k} - c_{x',k} \begin{cases} \leq c_{xx'} & \text{if } y_{xx',k} = 0; \\ = c_{xx'} & \text{if } y_{xx',k} > 0 \end{cases} \quad (3.3)$$

that is, the quantity of drivers using the road xx' to a location k not in a shorted one is zero.

3.1.3 Michael Florian's model

Basing on the idea of Beckmann, McGuire and Winsten, in 1984, Michael Florian considered a single-product single-criterion model where the cost function on each path depends on the traffic flows of the entire network. Consider a transportation network $G = [N, A, W]$ in which N is a set of nodes, $A = \{a_1, \dots, a_n\}$ is a set of n directed arcs which represent the transportation infrastructure and W is a set of all origin-destination pairs of nodes $x, x' \in N$

such that there is a path from x to x' . For a pair of nodes $w = (x, x')$, the set of available paths from the origin x to the destination x' is denoted by P_w , the index set I_w consists of all i such that $p_i \in P_w$ and the set of all available paths of the network is denoted by $P = \{p_1, \dots, p_m\} = \bigcup_{w \in W} P_w$.

Let v_a denote the flow of trips on arc $a \in A$ and y_p denote the flow of trips on path $p \in P$, then $v = (v_a)_{a \in A}$ is the vector of arc flows and $Y = (y_p)_{p \in P}$ is the vector of path flows over the entire network. A relationship between arc flows and path flows is given by

$$v_a = \sum_{w \in W} \sum_{p \in P_w} \delta_{ap} y_p \quad (3.4)$$

where

$$\delta_{ap} = \begin{cases} 1 & \text{if } a \in p \\ 0 & \text{otherwise.} \end{cases}$$

The demand on each $w \in W$ is denoted by d_w . The flow $Y = (y_p)_{p \in P}$ on the network is said to be feasible if it satisfies conservation of flow and nonnegativity

$$\sum_{p \in P_w} y_p = d_w, w \in W \quad \text{and} \quad y_p \geq 0, p \in P_w, w \in W. \quad (3.5)$$

The set of all feasible path flows is denoted by K .

One assumes that this network permits the flow of one type of traffic (vehicles or passengers) on its arcs. The cost of travelling in the arc v_a is denoted by a user cost function $c_a(v)$. This cost function may model the time delay for travel on that arc, in which case it is commonly referred to as a volume/delay function, it may however model other costs, such as fuel consumption.

The cost of each path $c_p = c_p(v)$ is the sum of the user costs of the arcs in the path

$$c_p = \sum_a \delta_{ap} c_a(v), \quad p \in P_w, w \in W. \quad (3.6)$$

Let $u_w = u_w(v)$ be the cost of the least cost path for any O/D pair w , that means

$$u_w = \min_{p \in P_w} c_p, \quad w \in W. \quad (3.7)$$

In this model, the demands d_w depends on the vector of least cost travel times for all the O/D pairs of the network $u = (u_w)_{w \in W}$ and are given by function $D_w(u)$

$$(0 <) d_w = D_w(u), \quad w \in W. \quad (3.8)$$

Then the flow $\bar{Y} \in K$ is in equilibrium if for every O/D pair $w \in W$, and every path $p \in P_w$, the following condition holds

$$c_p(\bar{Y}) \begin{cases} = u_w(\bar{Y}) & \text{if } \bar{y}_p > 0 \\ \geq u_w(\bar{Y}) & \text{if } \bar{y}_p = 0, \end{cases} \quad (3.9)$$

that is all the used directed paths are of equal cost.

It is relatively straightforward to show that (3.9) may be restated in the "complementarity" form

$$u_w(\bar{Y}) \leq c_p(\bar{Y}) \quad \text{and} \quad (c_p(\bar{Y}) - u_w(\bar{Y})) \bar{y}_p = 0, p \in P_w, w \in W \quad (3.10)$$

and that (3.9) and (3.10) are equivalent to the following statement: For every O/D pair $w \in W$, and paths $p, p' \in P_w$, one has

$$c_p(\bar{Y}) \leq c_{p'}(\bar{Y}) \quad \text{if } \bar{y}_p > 0.$$

Thus, Wardrop's first principle for single-class single-criterion model was stated mathematically in several forms.

3.2 Multi-criteria Traffic Network

As introduced at the beginning, the pattern of the traffic flows through a network is the result of a subtle and complex interaction between drivers, and in practice their decision in selecting one route of travel depends on many criteria simultaneously. Therefore, it is important to extend the basic model to multi-criteria one in which vector-valued cost function is considered.

3.2.1 Description of multi-product multi-criteria traffic network

Let us consider a supply-demand network $G = [N, A, W]$ in which N is a set of nodes, $A = \{a_1, \dots, a_n\}$ is a set of n directed arcs and W is a set of all origin-destination pairs of nodes $x, x' \in N$ such that there is a path from x to x' . For a pair of nodes $w = (x, x')$, the set of available paths from the origin x to the destination x' is denoted by P_w , the index set I_w consists of all i such that $p_i \in P_w$ and the set of all available paths of the network is denoted by $P = \{p_1, \dots, p_m\} = \bigcup_{w \in W} P_w$.

We assume that there are q different classes of products to traverse in the network. Given a path $p_i \in P$, let $y_{p_i}^j \in \mathbb{R}$ denote the amount of the j th class of product to transport on the path p_i . The matrix $Y = (y_{p_i}^j)_{m \times q}$ is called a path flow in the network. Thus, each row vector $Y_{p_i} = (y_{p_i}^1, \dots, y_{p_i}^q)$ of the matrix Y represents the vector of q classes of products to traverse the path p_i , while the column vector $Y^j = (y_{p_1}^j, \dots, y_{p_m}^j)^T$ (here $(\cdot)^T$ denotes the transpose) represents the vector of the j th class of product to traverse m paths of the network.

To evaluate the transportation of products in the network, a cost function for the path flow Y is given in form of a matrix $C(Y) = (c_{p_i}^j(Y))$ with vector entries $c_{p_i}^j(Y) = (c_{p_i,k}^j(Y)) \in \mathbb{R}^l$, with $l > 1$. The i th row of entries $C_{p_i}(Y) = (c_{p_i}^1(Y), \dots, c_{p_i}^q(Y))$ of the matrix $C(Y)$ represents the cost for the path p_i , and the j th column $C^j(Y) = (c_{p_1}^j(Y), \dots, c_{p_m}^j(Y))^T$ represents the cost concerning the j th class of product on m paths of the network. For every origin-destination pair $w \in W$, the set $C(w)$ consists of all vectors $C_{p_i}(Y)$ with $p_i \in P_w$.

Sometimes arc flows are also considered in association with path flows. If z_a^j denotes the amount of the j th product to be transported on the arc a , then the matrix Z whose entries are z_a^j , $a \in A$ and $j = 1, \dots, q$ represents an arc flow in the network. A vector-valued cost function for the arc flow Z is given by a matrix $\hat{C}(Z)$ with entries $\hat{c}_a^j(Z)$, $a \in A$ and $j = 1, \dots, q$. It is known that given a path flow Y , an associated arc flow Z can be determined by the formula

$$Z = \Delta Y,$$

where Δ is the so-called incident matrix whose entries δ_{ap} are given by

$$\delta_{ap} = \begin{cases} 1 & \text{if } a \text{ belongs to path } p \\ 0 & \text{otherwise.} \end{cases}$$

The cost functions of the arc flow Z and the path flow Y are then linked by the following matrix equality:

$$C(Y) = \Delta^T \hat{C}(Z).$$

From now on, for a path flow \bar{Y} , write \bar{C} and $\bar{c}_{p_i}^j$ instead of $C(\bar{Y})$ and $c_{p_i}^j(\bar{Y})$ if no misunderstanding occurs. We further assume that the demand function depends on the costs for all O/D pairs, that means we can suppose directly that the demand is a function of the path flow \bar{Y} . A positive demand function $d_w^j(\bar{Y})$ is given which expresses the quantity of the j th class of product to be transported from the origin x to the destination x' of the pair $w = (x, x') \in W$, and that the demand vector $d_w = (d_w^1(\bar{Y}), \dots, d_w^q(\bar{Y}))$ is non null. The lower and upper capacity constraints on each class of product j and on each path p_i are

respectively $l_{p_i}^j \in \mathbb{R}$ and $u_{p_i}^j \in \mathbb{R}$ with $l_{p_i}^j < u_{p_i}^j$. For a path p_i , the upper and lower capacity bound vectors $(u_{p_i}^1, \dots, u_{p_i}^q)$ and $(l_{p_i}^1, \dots, l_{p_i}^q)$ are respectively denoted by U_{p_i} and L_{p_i} . It is common to impose the following restrictions on the demand

$$\sum_{p_i \in P_w} L_{p_i} \leq d_w \leq \sum_{p_i \in P_w} U_{p_i}, \quad \forall w \in W. \quad (3.11)$$

Otherwise, the network would have no feasible flows. Moreover, if either of equalities holds in the above restrictions, then the network has a unique feasible path flow. Moreover if there exists $j \in \{1, \dots, q\}$ such that

$$d_w^j = \sum_{p_i \in P_w} l_{p_i}^j \quad \text{or} \quad d_w^j = \sum_{p_i \in P_w} u_{p_i}^j, \quad \forall w \in W, \quad (3.12)$$

then the feasible flow of the j -th product is unique. These cases are not interesting from the mathematical point of view. Therefore, from now on we assume

$$\sum_{p_i \in P_w} L_{p_i} < d_w < \sum_{p_i \in P_w} U_{p_i}, \quad \forall w \in W. \quad (3.13)$$

We say that a path flow Y is feasible if it satisfies the capacity constraints and the conservation of flows equations:

$$l_{p_i}^j \leq y_{p_i}^j \leq u_{p_i}^j \quad \forall i = 1, \dots, m; \forall j = 1, \dots, q; \quad (3.14)$$

$$\sum_{p_i \in P_w} y_{p_i}^j = d_w^j \quad \forall j = 1, \dots, q; \forall w \in W. \quad (3.15)$$

The set of all feasible flows for the flow Y is denoted by K and $G = [N, A, W]$ is called the network with capacity constraints.

We say that a path flow Y is feasible for the flow \bar{Y} if it satisfies the capacity constraints and the conservation of flows equations:

$$l_{p_i}^j \leq y_{p_i}^j \leq u_{p_i}^j \quad \forall p_i \in P; \forall j = 1, \dots, q; \quad (3.16)$$

$$\sum_{p_i \in P_w} y_{p_i}^j = d_w^j(\bar{Y}) \quad \forall j = 1, \dots, q; \forall w \in W. \quad (3.17)$$

The set of all feasible flows for the flow \bar{Y} is denoted by $K(\bar{Y})$ and $G = [N, A, W]$ is called the network with capacity constraints and elastic demand with respect to the feasible flow \bar{Y} . It is clear that $K(\bar{Y})$ is a closed convex set for every fixed \bar{Y} .

We notice that for a given path flow \bar{Y} , it may not satisfy demands for oneself, i.e., there exists j_0 and w_0 such that

$$\sum_{p_i \in P_w} \bar{y}_{p_i}^{j_0} \neq d_{w_0}^{j_0}(\bar{Y}). \quad (3.18)$$

When in the network, the value of d changes, the feasible set of flows/demands $K(d)$ can be defined as follows

$$K(d) = \{(Y, d) \mid \sum_{p_i \in P_w} y_{p_i}^j = d_w^j, l_{p_i}^j \leq y_{p_i}^j \leq u_{p_i}^j, p_i \in P_w, w \in W, j = 1, \dots, q\},$$

then we call $G = [N, A, W]$ the network with capacity constraints and elastic demand.

When $l_{p_i}^j = 0$ and $u_{p_i}^j = \infty$ for all p_i and j the network is called without capacity constraints.

In the subsections 3.2.2 and 3.2.3, we wish to compare different concepts of vector equilibrium. We restrict ourselves to the case of network without capacity constraints for the sake of simplicity and compatibility with existing definitions we meet in the literature.

3.2.2 Single-product multi-criteria supply demand network

In this model, there is only one product to traverse in the network. Let z_a denote the traffic load on arc $a \in A$ and let y_p denote the traffic flow on path $p \in P$. As before we have $z_a = \sum_{p \in P} y_p \delta_{ap}$ and $z = \Delta y$ whose entries are δ_{ap} for $a \in A$ and $p \in P$.

We shall assume throughout this subsection that the demand d_w of the traffic flow for each O/D pair $w \in W$ is fixed. A path flow y is said to be feasible if $y \geq 0$ and it satisfies the conservation flow equation

$$\sum_{p \in P_w} y_p = d_w \text{ for all } w \in W.$$

The set of all feasible path flows is denoted K . Assume further that a vector cost function \hat{c}_a is given on each arc $a \in A$, depending on the traffic arc flow z and taking values in a finite dimensional space \mathbb{R}^l with $l \geq 2$. In many classical models $l = 2$, which corresponds to two criteria: travel time and travel cost. Then the vector cost function c_p on path p depends on the path flow y and is computed by

$$c_p(y) = \sum_{a \in A} \hat{c}_a(z) \delta_{ap}. \quad (3.19)$$

Let $C(y)$ denote the $l \times m$ -matrix, the columns of which are $c_p, p \in P$ and $\hat{C}(z)$ the $l \times n$ -matrix, the columns of which are $\hat{c}_a, a \in A$. These matrices are linked by the formula

$$C(y) = \hat{C}(z) \Delta.$$

We recall below definitions of vector equilibrium and weak vector equilibrium corresponding with this kind of network, which have been originally proposed by Chen and Yen [10] in 1993.

Definition 3.2.1 *Let \bar{Y} be a feasible flow. We say that \bar{Y} is a vector equilibrium if for every $w \in W$ and $p_\alpha, p_\beta \in P_w$ one has implication*

$$\bar{c}_{p_\alpha} \geq \bar{c}_{p_\beta} \Rightarrow \bar{y}_{p_\alpha} = 0.$$

And \bar{Y} is a weak vector equilibrium if for every $w \in W$ and $p_\alpha, p_\beta \in P_w$ one has implication

$$\bar{c}_{p_\alpha} > \bar{c}_{p_\beta} \Rightarrow \bar{y}_{p_\alpha} = 0.$$

It is clear that every vector equilibrium is weak vector equilibrium, but the converse is not true in general. These two kinds of equilibrium are natural generalizations of the Wardrop equilibrium for a scalar valued cost in which inequality $\bar{c}_{p_\alpha} \geq \bar{c}_{p_\beta}$ means $\bar{c}_{p_\alpha} > \bar{c}_{p_\beta}$, and therefore there is no distinction between weak vector equilibrium and vector equilibrium.

In 1999 Chen, Goh and Yang in [9] introduced G_a -equilibrium, where the function G_a is given in Chapter 1.

Definition 3.2.2 *A feasible flow \bar{Y} is said to be G_a -equilibrium if there exists $a \in \mathbb{R}_+^l$ such that for every $w \in W, p_\alpha, p_\beta \in P_w$,*

$$G_a(\bar{c}_{p_\alpha}) > G_a(\bar{c}_{p_\beta}) \Rightarrow \bar{y}_{p_\alpha} = 0.$$

The authors of the above mentioned work proved the following result (Theorem 4.5).

Theorem 3.2.3 *A feasible flow is weak vector equilibrium if and only if it is G_a -equilibrium.*

Unfortunately this theorem is not always true. Actually the "if" part is true. In fact, by assuming that the feasible flow \bar{y} is G_a -equilibrium. If for some $p_\alpha, p_\beta \in P_w$ we have $\bar{c}_{p_\alpha} > \bar{c}_{p_\beta}$, then for $a \in \mathbb{R}_+^l$, $G_a(\bar{c}_{p_\alpha}) > G_a(\bar{c}_{p_\beta})$ by weakly increasing property of the function G_a . By hypothesis, we obtain $\bar{y}_{p_\alpha} = 0$. It deduces that \bar{Y} is weak vector equilibrium. The "only if" part is not always true which is seen in the next counterexample.

Example 3.2.4 Consider a network problem with one pair of origin-destination nodes $w = (x, x')$ and three available paths: $P_w = \{p_1, p_2, p_3\}$. Assume that the travel demand for w is $d_w = 15$, and

$$\begin{aligned} c_{p_1}(y) &= (3y_{p_1} + 2y_{p_2}, y_{p_1} + y_{p_2})^T \\ c_{p_2}(y) &= (y_{p_1} + 5y_{p_2}, 2y_{p_2})^T \\ c_{p_3}(y) &= (y_{p_1} + y_{p_3}, 2y_{p_1} + y_{p_3})^T. \end{aligned}$$

With the feasible flow $\bar{y}_{p_1} = 3, \bar{y}_{p_2} = 7, \bar{y}_{p_3} = 5$, we have

$$\bar{c}_{p_1} = (23, 10)^T \quad \bar{c}_{p_2} = (38, 6)^T \quad \bar{c}_{p_3} = (12, 11)^T.$$

Clearly, \bar{Y} is weak vector equilibrium. Nevertheless, \bar{Y} is not G_a -equilibrium. In fact, take any $a \in \mathbb{R}^2$, we obtain either

$$G_a(\bar{c}_{p_1}) < G_a(\bar{c}_{p_2})$$

or

$$G_a(\bar{c}_{p_1}) < G_a(\bar{c}_{p_3})$$

and $\bar{y}_{p_2} = 7 > 0, \bar{y}_{p_3} = 5 > 0$.

Consequently neither Corollary 4.7 [9] which gives necessary and sufficient conditions for weak vector equilibrium, nor Theorem 3.2 and Corollary 3.1 in Chen [8] are correct. In 2006 Li, Yang and Chen [31] proposed another kind of G_a -equilibrium flow, which is called weak G_a -equilibrium and established a necessary and sufficient condition of a weak vector equilibrium.

Definition 3.2.5 A feasible flow \bar{Y} is said to be weak G_a -equilibrium if for every $w \in W, p_\alpha, p_\beta \in P_w$ one has

$$G_{\bar{c}_{p_\alpha}}(\bar{c}_{p_\beta}) < 0 \Rightarrow \bar{y}_{p_\alpha} = 0.$$

It is clear that every G_a -equilibrium is weak G_a -equilibrium, but the converse is not true in general. Here is an example.

Example 3.2.6 Consider a network problem with one pair of origin-destination nodes $w = (x, x')$ and three available paths: $P_w = \{p_1, p_2, p_3\}$. Assume that the travel demand for w is $d_w = 20$, and

$$\begin{aligned} c_{p_1}(y) &= (2y_{p_1}, 5y_{p_1} + 3y_{p_2})^T \\ c_{p_2}(y) &= (3y_{p_2}, y_{p_1} + y_{p_3})^T \\ c_{p_3}(y) &= (2y_{p_2} + y_{p_3}, 3y_{p_3})^T. \end{aligned}$$

With the feasible flow $\bar{y}_{p_1} = 3, \bar{y}_{p_2} = 10, \bar{y}_{p_3} = 7$, we have

$$\bar{c}_{p_1} = (6, 45)^T \quad \bar{c}_{p_2} = (30, 17)^T \quad \bar{c}_{p_3} = (27, 21)^T.$$

Then

$$\begin{aligned} G_{\bar{c}_1}(\bar{c}_2) &= \max \{30 - 6, 17 - 45\} = 24, \\ G_{\bar{c}_1}(\bar{c}_3) &= \max \{27 - 6, 21 - 45\} = 21, \\ G_{\bar{c}_2}(\bar{c}_1) &= \max \{6 - 30, 45 - 17\} = 28, \\ G_{\bar{c}_2}(\bar{c}_3) &= \max \{27 - 30, 21 - 17\} = 4, \\ G_{\bar{c}_3}(\bar{c}_1) &= \max \{6 - 27, 45 - 21\} = 24, \\ G_{\bar{c}_3}(\bar{c}_2) &= \max \{30 - 27, 17 - 21\} = 3. \end{aligned}$$

Therefore \bar{y} is weak G_a -equilibrium. Nevertheless, \bar{y} is not G_a -equilibrium. In fact, for any $a = (a_1, a_2) \in \mathbb{R}^2$, we have

$$\begin{aligned} G_a(\bar{C}_1) &= \max \{6 - a_1, 45 - a_2\}, \\ G_a(\bar{C}_2) &= \max \{30 - a_1, 17 - a_2\}, \\ G_a(\bar{C}_3) &= \max \{27 - a_1, 21 - a_2\}. \end{aligned}$$

If $G_a(\bar{C}_3) = 27 - a_1$, then we have $G_a(\bar{C}_3) < G_a(\bar{C}_2)$ and $\bar{y}_{p_2} = 10 \neq 0$.
If $G_a(\bar{C}_3) = 21 - a_2$, then we have $G_a(\bar{C}_3) < G_a(\bar{C}_1)$ and $\bar{y}_{p_1} = 3 \neq 0$.

It turns out that weak vector equilibrium and weak G_a -equilibrium are equivalent, which was proved in [31].

Proposition 3.2.7 [31] *A flow $\bar{y} \in K$ is a weak vector equilibrium if and only if $\bar{y} \in K$ is a weak G_a -equilibrium.*

To go further we recall some notations.

$$\begin{aligned} \Lambda &= \left\{ \lambda = (\lambda_1, \dots, \lambda_l)^T \in \mathbb{R}^l \mid \lambda_i \geq 0, \sum_{i=1}^l \lambda_i = 1 \right\}, \\ ri(\Lambda) &= \left\{ \lambda = (\lambda_1, \dots, \lambda_l)^T \in \mathbb{R}^l \mid \lambda_i > 0, \sum_{i=1}^l \lambda_i = 1 \right\}, \\ C_w(y) &= \{c_p(y) : p \in P_w\}, \\ V\text{-min}(C_w(y)) &= \{c_p(y) \mid \nexists c_{p'}(y) \in C_w(y) \text{ such that } c_p(y) - c_{p'}(y) \in \mathbb{R}_+^l \setminus \{0\}\}. \end{aligned}$$

Goh and Yang [24] introduced parametric equilibrium (λ -equilibrium).

Definition 3.2.8 *A feasible flow \bar{Y} is said to be a parametric equilibrium if for every $w \in W, p_\alpha \in P_w$ and for a parametric $\lambda \in \Lambda$ given, there exists $\bar{c}_w \in \text{Min}(\bar{C}_w)$ such that*

$$\lambda^T \bar{c}_{p_\alpha} > \lambda^T \bar{c}_w \Rightarrow \bar{y}_{p_\alpha} = 0.$$

They proved the following sufficient and necessary condition for a vector equilibrium.

Proposition 3.2.9 [24] *Let the following assumption hold*

$$\text{Min}(\bar{C}_w) \subset \text{Min}(\text{co}(\bar{C}_w)), \text{ where } \text{co}(\bar{C}_w) \text{ is the convex hull of } \bar{C}_w, \quad (3.20)$$

Then the following assertions hold:

- i) *If a feasible flow \bar{y} is a vector equilibrium and assumption 3.20 holds, then there exists $\lambda \in \Lambda$ such that it is a parametric equilibrium.*
- ii) *If a feasible flow \bar{y} is a parametric equilibrium for some $\lambda \in ri(\Lambda)$, then it is a vector equilibrium.*

Again the assertion i) does not always hold. This is seen in the next example. We notice also that Example 2.1 in [31] fails because the assumption 3.20 in that example does not hold.

Example 3.2.10 *Consider a network problem with one pair of origin-destination nodes $w = (x, x')$ and three available paths: $P_w = \{p_1, p_2, p_3\}$. Assume that the travel demand for w is $d_w = 10$, and*

$$\begin{aligned} c_1(Y) &= (2y_1 + 2y_2, y_1 + 2y_2)^T \\ c_2(Y) &= (y_1 + 2y_2 + y_3, 3y_2)^T \\ c_3(Y) &= (y_1 + y_2 + 2y_3, y_3)^T. \end{aligned}$$

With the feasible flow $\bar{y}_1 = 3, \bar{y}_2 = 2, \bar{y}_3 = 5$, we have

$$\bar{c}_1 = (10, 7)^T \quad \bar{c}_2 = (12, 6)^T \quad \bar{c}_3 = (15, 5)^T.$$

Clearly, \bar{Y} is a vector equilibrium. Nevertheless, \bar{Y} is not a parametric equilibrium. In fact, although $\bar{C}(w) = \text{Min}(\bar{C}(w)) = \{\bar{c}_1, \bar{c}_2, \bar{c}_3\}$ and assumption 3.20 holds, for any $\lambda \in \Lambda$, we have $\lambda^T \bar{c}_3 > \lambda^T \bar{c}_1$ and $\bar{y}_3 = 5 > 0$.

To obtain a complete characterization for a vector equilibrium, Li, Yang and Chen [31] introduced another parametric equilibrium, which is called "weakened parametric equilibrium".

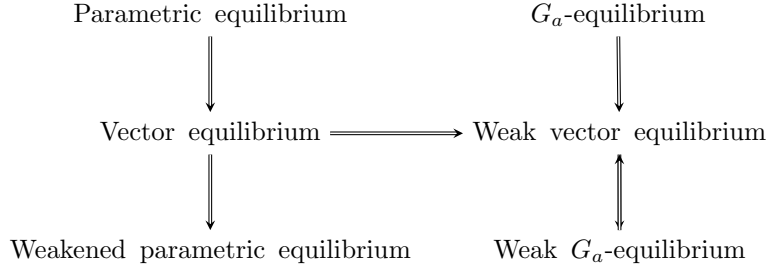
Definition 3.2.11 A feasible flow \bar{Y} is said to be a weakened parametric equilibrium if for every $w \in W, p_\alpha \in P_w$ and for any $\lambda \in \Lambda$, there exists $\bar{c}_w \in \text{Min}(\bar{C}_w)$ such that

$$\lambda^T \bar{c}_{p_\alpha} > \lambda^T \bar{c}_w \Rightarrow \bar{y}_{p_\alpha} = 0.$$

Under the assumption 3.20 they obtained a necessary condition for vector equilibrium.

Proposition 3.2.12 [31] Let \bar{y} be a feasible pattern flow on the network G and assumption 3.20 holds. Then if \bar{y} is a vector equilibrium, it is a weakened parametric equilibrium.

We summarize a relationship between the aforementioned concepts of equilibrium in the following diagram.



3.2.3 Multi-product single-criterion supply demand network

The multi-product single-criterion supply demand network can be explained as a network in which certain goods are produced by suppliers and need to be shipped to destination points according to given demand. The cost of transporting different products along an arc may differ.

Consider a supply demand model without capacity constraint. Let \bar{Y} be a feasible flow. We are interested in the following conditions which can also be considered as different types of equilibrium.

- (B1) For every $w \in W$ and $p_\alpha \in P_w$,

$$\begin{aligned} \bar{Y}_{p_\alpha} \geq 0 &\Rightarrow \bar{C}_{p_\alpha} = \text{Inf}(\bar{C}(w)), \\ \bar{Y}_{p_\alpha} = 0 &\Rightarrow \bar{C}_{p_\alpha} \geq \text{Inf}(\bar{C}(w)). \end{aligned}$$

- (B2) For every $w \in W$ and $p_\alpha, p_\beta \in P_w$,

$$[(\bar{C}(w) - \bar{C}_{p_\beta}) \cap (-\mathbb{R}_+^q) = \{0\}, \bar{C}_{p_\alpha} - \bar{C}_{p_\beta} \neq 0] \Rightarrow \bar{Y}_{p_\alpha} = 0.$$

- (B3) For every $w \in W$ and $p_\alpha, p_\beta \in P_w$,

$$[\text{clcone}(\overline{C}(w) + \mathbb{R}_+^q - \overline{C}_{p_\beta}) \cap (-\mathbb{R}_+^q) = \{0\}, \overline{C}_{p_\alpha} - \overline{C}_{p_\beta} \neq 0] \Rightarrow \overline{Y}_{p_\alpha} = 0.$$

- (B4) For every $w \in W$ and $p_\alpha, p_\beta \in P_w$,

$$\overline{C}_{p_\alpha} - \overline{C}_{p_\beta} \geq 0 \Rightarrow \overline{Y}_{p_\alpha} = 0.$$

- (B5) For every $w \in W$ and $p_\alpha \in P_w$,

$$\overline{C}_{p_\alpha} \notin \text{Min}(\overline{C}(w)) \Rightarrow \overline{Y}_{p_\alpha} = 0.$$

- (B6) For every $w \in W, p_\alpha, p_\beta \in P_w$, and $j = 1, \dots, q$,

$$\overline{c}_{p_\alpha}^j - \overline{c}_{p_\beta}^j > 0 \Rightarrow \overline{y}_{p_\alpha}^j = 0.$$

Remark 3.2.13

1) Condition (B1) has been introduced by Cheng and Wu [11] and the pattern flow \overline{Y} satisfying it is called a Wardrop equilibrium.

Notice that the second implication of (B1) is superfluous because for any path $p_\alpha \in P_w$ inequality $\overline{C}_\alpha \geq \text{Inf}(\overline{C}(w))$ is always true and if there exists an equilibrium satisfying the condition (B1), that is the unique solution of the network.

2) Condition (B3) has been introduced by Wu and Cheng [66]. They used it to define the so-called Benson equilibrium which is a kind of Benson proper efficient solutions of vector optimization problems.

3) Condition (B4) has been studied in [11] for multi-criteria networks. In [51] Raciti called it a strong vector Wardrop equilibrium.

4) Cheng and Wu [11] proved that the conditions (B1) and (B4) are equivalent. However, if the set $\overline{C}(w)$ has no ideal minimal element, the implication (B4) \Rightarrow (B1) may fail as it is shown in the next example. Proposition 2.1 of [11] and Proposition 3.2 of [66] are then not always available.

Example 3.2.14 Consider a network problem with one pair of origin-destination nodes $w = (x, x')$, two products traverse the network with two available paths: $P_w = \{p_1, p_2\}$. The other data are given as below $d_w^1 = 5$, $d_w^2 = 2$, and

$$\begin{aligned} c_1^1(Y) &= 10y_1^1 & c_2^1(Y) &= 2y_2^1 + y_2^2 \\ c_1^2(Y) &= 3y_1^1 + y_1^2 & c_2^2(Y) &= y_2^1 + 10y_2^2 \end{aligned}$$

With the feasible flow $\overline{y}_1^1 = 3, \overline{y}_1^2 = 1, \overline{y}_2^1 = 2, \overline{y}_2^2 = 1$, we have

$$C_1(\overline{Y}) = (30, 10) \quad C_2(\overline{Y}) = (5, 12) \quad \text{Inf}(\overline{C}(w)) = (5, 10).$$

Clearly, \overline{Y} is an equilibrium according to (B4) as both $C_1 \not\geq C_2$ and $C_2 \not\geq C_1$. Nevertheless, \overline{Y} does not satisfy (B1) since $\overline{Y}_1 \geq 0$ but $\overline{C}_1 \neq \text{Inf}(\overline{C}(w))$.

As a matter of fact conditions (B1) and (B4) lead to different concepts of equilibrium whenever the multiplicity of products for transport in the network is present. Moreover the operation of taking closed cone in (B3) by Wu and Cheng [66] is unnecessary. To see this let us recall the following result from [37] with its proof.

Lemma 3.2.15 [37] Let D be a finite subset of \mathbb{R}^q and $d \in D$. Then the following relations are equivalent

- (i) $\text{clcone}(D + \mathbb{R}_+^q - d) \cap (-\mathbb{R}_+^q) = \{0\}$
- (ii) $(D - d) \cap (-\mathbb{R}_+^q) = \{0\}$.

Proof. The implication (i) \Rightarrow (ii) is clear because the set $D-d$ is a subset of $\text{clcone}(D+\mathbb{R}_+^q-d)$ and the origin of the space belongs to both of them. For the converse suppose the contrary that (ii) is true, but (i) is not. There is a nonzero vector a belonging to the intersection on the left hand side of (i), say

$$a = \lim_{\alpha \rightarrow \infty} t_\alpha (d_\alpha - d + u_\alpha) \quad (3.21)$$

for some positive numbers t_α , some vectors d_α from D and $u_\alpha \in \mathbb{R}_+^q$. Since D is a finite set, we may assume without loss of generality that $d_\alpha = d_0$ for some $d_0 \in D$. If $d_0 - d = 0$, we arrive at a contradiction that $a \in \mathbb{R}_+^q \cap (-\mathbb{R}_+^q)$ and $a \neq 0$. It remains to consider the case $d_0 - d \neq 0$. We may also assume that the sequence $\{t_\alpha\}_\alpha$ converges to some limit t among three possible values: 1) $t = 0$, 2) $t = \infty$, and 3) $t \in (0, \infty)$. In the first case, $a = \lim_{\alpha \rightarrow \infty} t_\alpha u_\alpha \in \mathbb{R}_+^q$, which contradicts the hypothesis. In the second case, it follows from (3.21) that $a = t_\alpha (d_\alpha - d + u_\alpha) + o(t_\alpha)$ with $\lim_{\alpha \rightarrow \infty} o(t_\alpha)/t_\alpha = 0$. By dividing the latter equality by t_α and passing to the limit as α tends to ∞ , we obtain $d_0 - d = \lim_{\alpha \rightarrow \infty} u_\alpha \in -\mathbb{R}_+^q \setminus \{0\}$ which contradicts (ii). In the case 3), a similar argument yields

$$d_0 - d = \frac{a}{t} - \lim_{\alpha \rightarrow \infty} u_\alpha \in -\mathbb{R}_+^q \setminus \{0\} \quad (3.22)$$

which is a contradiction too. \square

We remark that the conclusion of Lemma 3.2.15 remains true under a milder condition on D . For instance, when D is not finite, but the set $\text{cone}(D-d)$ has a compact base, which means that there is a compact set B not containing the origin of the space such that $\text{cone}(D-d) = \text{cone}(B)$, then the argument of the proof above goes through. In particular, the conclusion of Lemma 3.2.15 is true when D is a polyhedral set. Here are some relationships between (B1)-(B6).

Proposition 3.2.16 [37] *Given a feasible pattern flow \bar{Y} on the network G . The following assertions hold:*

(i) $(B1) \Leftrightarrow (B2) \Leftrightarrow (B3)$.

Each of these conditions implies that for every $w \in W$, the set $C(w)$ has ideal minimal elements. Moreover, under the latter condition on $C(w)$, all conditions (B1) through (B5) are equivalent.

(ii) $(B4) \Leftrightarrow (B5)$

(iii) $(B1) \Rightarrow (B6)$.

The converse $(B6) \Rightarrow (B1)$ is true provided $q = 1$.

Proof. We note that for every $w \in W$, the set $\bar{C}(w)$ is finite, hence in view of Lemma 3.2.15, conditions (B2) and (B3) are equivalent. To prove the first part of (i), it suffices to establish equivalence between (B1) and (B2). We assume (B1). Since for each $w \in W$ the demand vector d_w is non null, there must be some path p_{β_0} on which the flow $\bar{y}_{p_{\beta_0}}$ is non null. Hence the cost $\bar{C}_{p_{\beta_0}}$ is an ideal minimal element of $\bar{C}(w)$. Let $p_\beta \in P_w$ satisfy $(\bar{C}(w) - \bar{C}_{p_\beta}) \cap -\mathbb{R}_+^q = \{0\}$. Then $\bar{C}_{p_\beta} = \bar{C}_{p_{\beta_0}} = \text{Inf}(\bar{C}(w))$, and $\bar{C}_{p_\alpha} \geq \text{Inf}(\bar{C}(w))$ for every $p_\alpha \in P_w$ with $\bar{C}_{p_\alpha} - \bar{C}_{p_\beta} \neq 0$. By (B1), $\bar{y}_{p_\alpha} = 0$, which shows that (B2) holds. Now assume (B2). Since the set $\bar{C}(w)$ is finite, it has minimal elements. Let \bar{C}_{p_β} be one of them. Then $(\bar{C}(w) - \bar{C}_{p_\beta}) \cap -\mathbb{R}_+^q = \{0\}$. For any $p_\alpha \in P_w$, if \bar{C}_{p_α} is not minimal, then $\bar{C}_{p_\alpha} - \bar{C}_{p_\beta} \neq 0$ and by (B2), the corresponding flow \bar{y}_{p_α} is null. If \bar{C}_{p_α} is minimal, but $\bar{C}_{p_\alpha} - \bar{C}_{p_\beta} \neq 0$, then we also have $\bar{y}_{p_\alpha} = 0$ by (B2). With \bar{C}_{p_α} minimal, switching the roles of \bar{C}_{p_α} and \bar{C}_{p_β} we obtain $\bar{y}_{p_\beta} = 0$ too. Thus, if the set $\text{Min}(\bar{C}(w))$ consists of more than two elements, the flow \bar{y} is null on every path joining w , which is impossible because the demand is not null. Consequently, the set $\text{Min}(\bar{C}(w))$ has only one value, say \bar{C}_* . We deduce $\bar{C}_{p_\alpha} = \bar{C}_*$ for

all $p_\alpha \in P_w$, which shows that C_* is the ideal minimal element of $\overline{C}(w)$ and (B1) follows. For the second part of (i), assume that for every $w \in W$, the set $\overline{C}(w)$ has ideal minimal elements. It suffices to prove equivalence between (B1) and (B4), because the equivalence between (B4) and (B5) will be given in (ii). Let $p_\alpha, p_\beta \in P_w$ satisfy $\overline{C}_{p_\alpha} - \overline{C}_{p_\beta} \geq 0$. Then \overline{C}_{p_α} is not ideal minimal. Under (B1), one has $\overline{y}_{p_\alpha} = 0$ and obtains (B4). Conversely, if (B4) holds and if $\overline{y}_{p_\alpha} \geq 0$, then \overline{C}_{p_α} must be ideal minimal, which shows that (B1) is true. Indeed, if \overline{C}_{p_α} were not ideal minimal, there would exist some ideal element \overline{C}_{p_β} such that $\overline{C}_{p_\alpha} - \overline{C}_{p_\beta} \geq 0$ which yields $\overline{y}_{p_\alpha} = 0$, a contradiction. By this, (B4) is equivalent to (B1). We proceed to (ii) by assuming (B4). Let $\overline{C}_{p_\alpha} \notin \text{Min}(\overline{C}(w))$. By definition, there is some $\overline{C}_{p_\beta} \in \overline{C}(w)$ such that $\overline{C}_{p_\alpha} \geq \overline{C}_{p_\beta}$. In view of (B4) one has $\overline{y}_{p_\alpha} = 0$ and (B5) follows. Conversely, if (B5) holds and if $\overline{C}_{p_\alpha} - \overline{C}_{p_\beta} \geq 0$ for some $p_\alpha, p_\beta \in P_w$, then \overline{C}_{p_α} is not a minimal element of $\overline{C}(w)$ and in view of (B5) the flow \overline{y}_{p_α} is null. Thus, (B4) is true and we obtain the equivalence between (B4) and (B5). Finally, suppose (B1). Strict inequality $\overline{c}_{p_\alpha}^i > \overline{c}_{p_\alpha}^{i'}$ for some $i, i' \in \{1, \dots, q\}$ and $p_\alpha \in P_w$ in (B6) implies that C_{p_α} is not an ideal minimal element of $\overline{c}(w)$. By (B1), one has $\overline{y}_{p_\alpha} = 0$. In particular, $\overline{y}_{p_\alpha}^i = 0$ and (B6) follows. When $q = 1$ inequality $\overline{y}_{p_\alpha} \geq 0$ means $\overline{y}_{p_\alpha}^1 > 0$, and so under (B6) one has $\overline{c}_{p_\alpha}^1 \leq \overline{c}_{p_\beta}^1$ for all $p_\beta \in P_w$, that is $\overline{c}_{p_\alpha}^1 = \text{Inf}(\overline{C}(w))$. Thus, for $q = 1$, conditions (B1) and (B6) are equivalent. \square

In multi-product networks, equilibria defined via (B4) and (B6) do not follow from each other. We can see that in the following examples:

Example 3.2.17 Consider a network consisting of four nodes $\{N_i : i = 1, \dots, 4\}$, one origin destination pair $w = (N_1, N_4)$ and two paths p_1 and p_2 connecting w via N_2 and N_3 respectively. We assume there are two products in the network. Let a feasible pattern flow Y be given by its rows $Y_1 = (20, 320)$ and $Y_2 = (10, 500)$ representing the quantities of the two products to traverse the paths p_1 and p_2 respectively. Assume further that the cost matrix associated to the path flow Y has its rows $C_1 = (2, 16)$ and $C_2 = (1, 25)$. Then (B4) holds, but not (B6) because $C_1^1 = 2 > 1 = C_2^1$ with $y_{p_1}^1 = 20 \neq 0$.

Example 3.2.18 Consider the network of the previous example. Let a feasible pattern flow Y be given by its rows $Y_1 = (0, 830)$ and $Y_2 = (30, 0)$ representing the quantities of the two products to traverse the paths p_1 and p_2 respectively. Assume further that the cost matrix associated to the path flow Y has its rows $C_1 = (2, 16)$ and $C_2 = (2, 25)$. Then (B6) holds, but not (B4) because $C_2 \geq C_1$ with $Y_2 \neq 0$.

3.2.4 Multi-product multi-criteria supply demand network

In this subsection we study a multi-product multi-criteria supply demand network which is one of the topics of our attention. In the definition below inequality of matrices is understood as vector inequality in the space $\mathbb{R}^{l \times q}$, and the negation of strict inequality $Y_{p_\alpha} \not\leq L_{p_\alpha}$ means there is at least one component of Y_{p_α} less than or equal to the corresponding component of L_{p_α} .

Let \overline{Y} be a feasible solution. We consider the following conditions:

- (H1) For every $w \in W$ and $p_\alpha \in P_w$,

$$\overline{C}_{p_\alpha} \geq \text{Inf}\overline{C}(w) \implies \overline{Y}_{p_\alpha} = L_{p_\alpha};$$

- (H2) For every $w \in W$ and $p_\alpha \in P_w$,

$$\overline{C}_{p_\alpha} \geq \text{Inf}\overline{C}(w) \implies \text{either } \overline{Y}_{p_\alpha} = L_{p_\alpha} \text{ or } \overline{Y}_{p_\beta} = U_{p_\beta}$$

$$\text{for all } p_\beta \in P_w \text{ with } \overline{C}_{p_\beta} = \text{Inf}\overline{C}(w);$$

- (H3) For every $w \in W$ and $p_\alpha \in P_w$,

$$\begin{aligned} \bar{C}_{p_\alpha} \geq \text{Inf}\bar{C}(w) &\implies \text{either } \bar{Y}_{p_\alpha} = L_{p_\alpha} \text{ or } \bar{Y}_{p_\beta} = U_{p_\beta} \\ &\text{for some } p_\beta \in P_w \text{ with } \bar{C}_{p_\beta} = \text{Inf}\bar{C}(w). \end{aligned}$$

The following implications are clear:

$$(H1) \implies (H2) \implies (H3).$$

Note that the converse implications are not true in general. Actually these conditions are closely related to the existence of ideal minimal costs.

Proposition 3.2.19 *If the feasible pattern flow \bar{Y} satisfies either of (H1), (H2) and (H3), then for every origin destination pair $w \in W$ the set of vector costs $C(w)$ has ideal minimal elements.*

Proof. Due to the implications of (H1), (H2) and (H3) we have mentioned, it suffices to prove the proposition when the flow \bar{Y} satisfies (H3). Suppose to the contrary that for some origin destination pair $w \in W$ the set $C(w)$ has no ideal elements. This means that $\bar{C}_{p_\alpha} \geq \text{Inf}\bar{C}(w)$ for all $p_\alpha \in P_w$. In view of (H3), we have $\bar{Y}_{p_\alpha} = L_{p_\alpha}$. Summing up \bar{Y}_{p_α} over all paths p_α joining w , we obtain

$$d_w = \sum_{p_\alpha \in P_w} \bar{Y}_{p_\alpha} = \sum_{p_\alpha \in P_w} \bar{L}_{p_\alpha}$$

which contradicts (3.13). □

Since in most situations, ideal elements of a set of vectors do not exist, conditions (H1), (H2) and (H3) are very difficult to be fulfilled. Instead, we offer a better choice for equilibrium in multi-product multi-criteria models with capacity constraints. We consider also the following conditions:

- (H4) For every $w \in W$ and $p_\alpha, p_\beta \in P_w$,

$$\bar{C}_{p_\alpha} \geq \bar{C}_{p_\beta} \implies \text{either } \bar{Y}_{p_\alpha} = L_{p_\alpha} \text{ or } \bar{Y}_{p_\beta} = U_{p_\beta}$$

- (H5) For every $w \in W$ and $p_\alpha, p_\beta \in P_w$,

$$\bar{C}_{p_\alpha} \geq \bar{C}_{p_\beta} \implies \text{either } \bar{Y}_{p_\alpha} \not\leq L_{p_\alpha} \text{ or } \bar{Y}_{p_\beta} \not\leq U_{p_\beta}$$

- (H6) For every $w \in W$ and $p_\alpha, p_\beta \in P_w$,

$$\bar{C}_{p_\alpha} > \bar{C}_{p_\beta} \implies \text{either } \bar{Y}_{p_\alpha} = L_{p_\alpha} \text{ or } \bar{Y}_{p_\beta} = U_{p_\beta}.$$

Remark 3.2.20

1) In a model without capacity constraints, condition (H4) collapses to (B4) of the previous subsection.

2) Again in a model without constraints condition (H5) is named in [51] as a weak vector Wardrop principle. It was introduced by Oettli in [47] to express a necessary condition for a vector variational equilibrium.

3) Conditions (H1)-(H6) given above were developed for networks with capacity constraints by Luc [37].

For networks with capacity constraints, the following notion of equilibrium introduced by Li, Teo and Yang has received a lot of attention (see [32, 33, 38] for instance): a feasible pattern flow \bar{y} is said to be a vector equilibrium if for every $j = 1, \dots, q, w \in W$ and $p_\alpha, p_\beta \in P_w$ one has implication

$$\bar{c}_{p_\alpha}^j \geq \bar{c}_{p_\beta}^j \implies \text{either } \bar{y}_{p_\alpha}^j = l_{p_\alpha}^j \text{ or } \bar{y}_{p_\beta}^j = u_{p_\beta}^j.$$

Observe that this concept of equilibrium and the ones given in (H4) and (H5) are equivalent when $q = 1$, but they are not comparable when $q > 1$ as it is shown by Examples 3.2.21 and 3.2.22. Because the implication in the definition of vector equilibrium involves individually q products, its analysis is much similar to single product multi-criteria models (see [32]). In contrast to this, equilibrium in conditions (H4) and (H5) consider collectively different kinds of products and seem to be more suitable in the models in which a certain proportion between the products to transport is to be kept (for instance, we cannot transport cows without dried grass on a long distance even if on a route the cost for cows is cheaper and the cost for dried grass is more expensive than on another route).

Example 3.2.21 Consider a network problem with only one pair of origin-destination nodes $w = (x, x')$, two criteria and two products to traverse the network with two available paths: $P_w = \{p_1, p_2\}$. Assume that $d_w^1 = 6, d_w^2 = 13, l_{p_i}^j = 2, u_{p_i}^j = 10$ for $p_i \in P_w, j = 1, 2$, and

$$\begin{array}{cccc} \bar{y}_{p_1}^1 = 2 & \bar{y}_{p_1}^2 = 3 & \bar{c}_{p_1}^1 = (20, 10)^T & \bar{c}_{12} = (15, 8)^T \\ \bar{y}_{p_2}^1 = 4 & \bar{y}_{p_2}^2 = 10 & \bar{c}_{p_2}^1 = (15, 9)^T & \bar{c}_{p_2}^2 = (10, 7)^T \end{array}$$

Since $\bar{c}_{p_1}^1 \geq \bar{c}_{p_2}^1$ and $\bar{y}_{p_1}^1 = l_{p_1}^1 = 2; \bar{c}_{p_1}^2 \geq \bar{c}_{p_2}^2$ and $\bar{y}_{p_2}^2 = u_{p_2}^2 = 10, \bar{Y}$ is a vector equilibrium. However, \bar{Y} is not a strong vector equilibrium. In fact, $\bar{Y}_{p_2} \neq U_{p_2}$ and $\bar{Y}_{p_1} \neq L_{p_1}$, although $\bar{C}_{p_1} \geq \bar{C}_{p_2}$.

Example 3.2.22 Consider a network problem with only one pair of origin-destination nodes $w = (x, x')$, two criteria and two products to traverse the network with three available paths: $P_w = \{p_1, p_2, p_3\}$. Assume that $d_w^1 = 15, d_w^2 = 20, l_{p_i}^j = 2, u_{p_i}^j = 10$ for $p_i \in P_w$ and $j = 1, 2$, and

$$\begin{array}{cccc} \bar{y}_{p_1}^1 = 3 & \bar{y}_{p_1}^2 = 6 & \bar{c}_{p_1}^1 = (12, 20)^T & \bar{c}_{p_1}^2 = (19, 10)^T \\ \bar{y}_{p_2}^1 = 3 & \bar{y}_{p_2}^2 = 9 & \bar{c}_{p_2}^1 = (20, 18)^T & \bar{c}_{p_2}^2 = (15, 12)^T \\ \bar{y}_{p_3}^1 = 9 & \bar{y}_{p_3}^2 = 5 & \bar{c}_{p_3}^1 = (18, 12)^T & \bar{c}_{p_3}^2 = (10, 19)^T \end{array}$$

Since $\bar{C}_{p_\alpha} \not\geq \bar{C}_{p_\beta}$ and $\bar{C}_{p_\alpha} \not\leq \bar{C}_{p_\beta}, \forall p_\alpha, p_\beta \in P_w$ and $L_i < \bar{Y}_{p_i} < U_{p_i}, \forall p_i \in P_w, \bar{Y}$ is a strong vector equilibrium. However, \bar{Y} is not a vector equilibrium. In fact, $\bar{c}_{p_1}^1 > \bar{c}_{p_3}^1$ but $\bar{y}_{p_2}^1 \neq l_{p_2}^1$ and $\bar{y}_{p_3}^1 \neq u_{p_3}^1$.

In a similar vein, Raciti [51] studies equilibrium for a model without capacity constraints by requiring that for every $k \in \{1, \dots, l\}, w \in W$ and $p_\alpha, p_\beta \in P_w$ one has implication

$$\bar{c}_{p_\alpha, k}^j \geq \bar{c}_{p_\beta, k}^j \implies \bar{y}_{p_\alpha}^j = 0 \text{ for all } j = 1, \dots, q.$$

In this definition not only products are considered individually, but the criteria too. So its study belongs to the category of single-product single-criterion network equilibria. Here are some relationships between the conditions of equilibrium.

Proposition 3.2.23 Let \bar{Y} be a feasible pattern flow. The following assertions hold.

- i) (H1) \implies (H4) \implies (H5).
- ii) (H4) \implies (H1) provided that for every $w \in W$, the set $C(w)$ has ideal minimal elements and that

$$d_w \not\leq \sum_{p_\beta \in P_w: C_{p_\beta} = \text{Inf}C(w)} U_{p_\beta} + \sum_{p_\alpha \in P_w: C_{p_\alpha} \neq \text{Inf}C(w)} L_{p_\alpha}.$$

Proof. The implication (H4) \Rightarrow (H5) is obvious. For the implication (H1) \Rightarrow (H4), let $\bar{C}_{p_\alpha} \geq \bar{C}_{p_\beta}$ for some $p_\alpha, p_\beta \in P_w$. Then \bar{C}_{p_α} is not ideal minimal, and $\bar{Y}_{p_\alpha} = L_{p_\alpha}$ by (H1). This shows that (H4) is satisfied. To prove (ii), we assume (H4). Let $\bar{C}_{p_\alpha} \geq \text{Inf}\bar{C}(w)$ for some $p_\alpha \in P_w$. Picking any $\bar{C}_{p_\beta} = \text{Inf}\bar{C}(w)$, we obtain $\bar{C}_{p_\alpha} \geq \bar{C}_{p_\beta}$ which implies that either $\bar{Y}_{p_\alpha} = L_{p_\alpha}$ or $\bar{Y}_{p_\beta} = U_{p_\beta}$. If $\bar{Y}_{p_\alpha} = L_{p_\alpha}$, we obtain (H1). If not, $\bar{Y}_{p_\gamma} = U_{p_\gamma}$ for all $p_\gamma \in P_w$ with $\bar{C}_{p_\gamma} = \text{Inf}\bar{C}(w)$. Consequently,

$$\begin{aligned} d_w &= \sum_{p_\alpha \in P_w} \bar{Y}_{p_\alpha} \\ &= \sum_{p_\beta \in P_w: \bar{C}_{p_\beta} = \text{Inf}\bar{C}(w)} U_{p_\beta} + \sum_{p_\alpha \in P_w: \bar{C}_\alpha \neq \text{Inf}\bar{C}(w)} \bar{Y}_{p_\alpha} \\ &\geq \sum_{p_\beta \in P_w: \bar{C}_{p_\beta} = \text{Inf}\bar{C}(w)} U_{p_\beta} + \sum_{p_\alpha \in P_w: \bar{C}_{p_\alpha} \neq \text{Inf}\bar{C}(w)} L_{p_\alpha} \end{aligned}$$

which contradicts the hypothesis. \square

Remark 3.2.24

- 1) The implication (H5) \Rightarrow (H4) is not true except for the case $q = 1$, in which the two conditions (H4) and (H5) coincide. Of course, both of them generalize Wardrop's equilibrium [64] when $q = 1$ and $l = 1$.
- 2) The results obtained in Proposition 3.2.19 and Proposition 3.2.23 of Luc [37] suggest to call a feasible flow satisfying (H1), (H4) and (H5) as an ideal equilibrium, a strong equilibrium and a weak equilibrium respectively.
- 3) Condition (H6) is a general version of weak vector equilibrium defined in Li, Teo, Yang [33] with respect to the number of considered products.

Note that a multi-product network hardly possesses ideal equilibrium flows, the concept of strong equilibrium seems to be most appropriate for multi-product networks. Weak equilibrium is particularly interesting in networks in which products are transported by bundles. For instance, machines sending from a factory to a destination are accompanied by a number of accessories. It is possible that on a path lower limits for certain accessories are reached while lower limits for other accessories are not. In such a model, strong equilibria infrequently exist and weak equilibria turn to be good substitutes.

Equilibrium in a multi-criteria traffic network without capacity constraints

In this chapter, we develop a new method to generate the set of equilibrium flows of a multi-criteria traffic network. To this end we introduce two optimization problems by using a vector version of the Heaviside Step function and the distance function to Pareto minimal elements and show that the optimal solutions of these problems are exactly the equilibria of the network. We study the objective functions by establishing their generic differentiability and local calmness at equilibrium solutions. Then we present an algorithm to generate a representative set of equilibrium solutions by using a modified Frank-Wolfe reduced gradient method and prove its convergence. We give some numerical examples to illustrate our algorithm and show its advantage over a popular method by using linear scalarization. A method of smoothing the objective functions by analytic approximations of the Heaviside Step function is also considered. Finally, we introduce the concept of robust equilibrium and obtain a formula to compute the radius of robustness together with an algorithm to find robust equilibrium flows.

The following concept of equilibrium is known as a vector version of Wardrop's famous user principle (see [24]).

Definition 4.0.25 *A feasible path flow y is said to be a vector equilibrium (respectively a weak vector equilibrium) of G if for every O/D pair $w \in W$ and for every couple of paths $p, p' \in P_w$ one has implication*

$$c_p(y) - c_{p'}(y) \geq 0 \text{ (respectively } c_p(y) - c_{p'}(y) > 0) \implies y_p = 0.$$

It is clear that every vector equilibrium is weak vector equilibrium, and the converse is not true in general. Note that the set of weak vector equilibria is closed if the vector cost functions are continuous, while it is not always the case for the set of vector equilibria (see Example 4.3.2). Further, if we denote by $C_w(y)$ the set of all vectors $c_p(y), p \in P_w$ for an O/D pair $w \in W$, then the above definition is equivalent to the implication

$$c_p(y) \notin \text{Min}(C_w(y)) \text{ (respectively } c_p(y) \notin \text{WMin}(C_w(y)) \text{)} \implies y_p = 0,$$

for every $p \in P_w$ and every $w \in W$.

4.1 Equivalent problems

A common technique to find equilibrium of a multi-criteria traffic network is to transform it to an equivalent problem the solution methods of which are already known. In this section we discuss two well-known approaches of such transformations: an approach by scalarization and an approach by variational inequalities. Then we introduce a new approach by constructing two optimization problems the solutions of which are exactly the vector equilibria of the network.

4.1.1 Scalarization

The first method for solving a multi-criteria traffic network equilibrium problem is to convert it to a single-criterion problem by scalarizing the vector cost function. Namely, let h be a real-valued function on the set $\{c_p(y) : y \in K, p \in P\}$ which satisfies a monotonicity (respectively weak monotonicity) condition: for every $w \in W, p$ and $p' \in P_w$,

$$h(c_p(y)) > h(c_{p'}(y)) \text{ if } c_p(y) \geq c_{p'}(y) \text{ (respectively } c_p(y) > c_{p'}(y)).$$

By considering the network G equipped with the scalar cost function

$$\pi_p(y) = h(c_p(y))$$

one says that a feasible path flow \bar{y} is a π -equilibrium if for every $w \in W$ and for every $p, p' \in P_w$, one has implication

$$\pi_p(\bar{y}) - \pi_{p'}(\bar{y}) > 0 \implies \bar{y}_p = 0.$$

It is clear that if h is monotone, then a π -equilibrium is a vector equilibrium and if h is weakly monotone, then a π -equilibrium is a weak vector equilibrium. The converse is not true in general. Here are some typical instances of scalarization.

1) *Linear scalarization.* In the classical bi-criteria models of [12, 14, 44, 45] the authors consider a vector cost function on arcs

$$\hat{c}_a(z) = \begin{pmatrix} t_a(z) \\ u_a(z) \end{pmatrix}$$

where $t_a(z)$ is the travel time function and $u_a(z)$ is the travel cost function on arc $a \in A$. They choose nonnegative weights λ_a^1 and λ_a^2 associated with t_a and u_a respectively on each arc $a \in A$ and define the scalarized cost function on paths as follows

$$\pi_p(y) = \sum_{a \in A} (\lambda_a^1, \lambda_a^2) \hat{c}_a(\Delta y) \delta_{ap} = \sum_{a \in A} (\lambda_a^1 t_a(\Delta y) + \lambda_a^2 u_a(\Delta y)) \delta_{ap}$$

for every flow $y \in K$ and every path $p \in P$.

Since $c_p(y) = \sum_{a \in A} \hat{c}_a(z) \delta_{ap}$, the vector cost function c_p on path p can be written as

$$c_p(y) = \sum_{a \in A} \begin{pmatrix} t_a(\Delta y) \\ u_a(\Delta y) \end{pmatrix} \delta_{ap}. \quad (4.1)$$

Assume there is some weight vector $(\alpha, \beta) \geq 0$ such that $(\lambda_a^1, \lambda_a^2) = (\alpha, \beta)$ for all $a \in A$, that is, the weights $(\lambda_a^1, \lambda_a^2)$ are common on all arcs and equal to (α, β) . Then the scalarized cost function π_p can be written as

$$\pi_p(y) = (\alpha, \beta) c_p(y).$$

The (linear) scalarizing function h defined by

$$h(c_p(y)) = (\alpha, \beta) c_p(y)$$

is monotone if $(\alpha, \beta) > 0$ and weakly monotone if $(\alpha, \beta) \geq 0$. Consequently, a π -equilibrium is a vector equilibrium or a weak vector equilibrium depending on whether $(\alpha, \beta) > 0$ or $(\alpha, \beta) \geq 0$.

It is worthwhile noting here that when the weights $(\lambda_a^1, \lambda_a^2)$ are distinct on arcs, a π -equilibrium is not necessarily a vector equilibrium or a weak vector equilibrium.

2) *Nonlinear scalarization.* As it was already said a weak vector equilibrium is not necessary a π -equilibrium when h takes a linear form. In other words, without any specific properties of the cost functions, not all weak vector equilibria of G may be obtained by solving network equilibrium problems in which the cost functions are of type $h(c_p(y)) = (\alpha, \beta)c_p(y)$ with $(\alpha, \beta) \geq 0$. To fulfill this gap nonlinear scalarizing functions are widely used in recent models ([24, 32, 37, 38]). Namely, for every path flow y and path $p \in P_w, w \in W$ we define a scalarized relative cost function $r_p(y)$ to be

$$r_p(y) = \max_{p' \in P_w} \min_{j=1, \dots, l} (c_{p,j}(y) - c_{p',j}(y))$$

where $c_{p,j}(y)$ denotes the j th component of $c_p(y)$. The function h defined by

$$h(c_p(y)) = r_p(y)$$

is weakly monotone. Indeed, let $p, p' \in P_w, w \in W$ with $c_p(y) > c_{p'}(y)$. Let $p'' \in P_w$ such that

$$h(c_{p'}(y)) = \min_{j=1, \dots, l} (c_{p',j}(y) - c_{p'',j}(y)).$$

We have

$$\begin{aligned} \min_{j=1, \dots, l} (c_{p,j}(y) - c_{p'',j}(y)) &= \min_{j=1, \dots, l} [(c_{p,j}(y) - c_{p',j}(y)) + (c_{p',j}(y) - c_{p'',j}(y))] \\ &\geq \min_{j=1, \dots, l} (c_{p,j}(y) - c_{p',j}(y)) + \min_{j=1, \dots, l} (c_{p',j}(y) - c_{p'',j}(y)) \\ &> \min_{j=1, \dots, l} (c_{p',j}(y) - c_{p'',j}(y)), \end{aligned}$$

and deduce

$$h(c_p(y)) \geq \min_{j=1, \dots, l} (c_{p,j}(y) - c_{p'',j}(y)) > h(c_{p'}(y)).$$

As before, we say that a feasible path flow \bar{y} is *r-equilibrium* if for every $w \in W$ and $p, p' \in P_w$, one has implication

$$r_p(\bar{y}) > r_{p'}(\bar{y}) \implies \bar{y}_p = 0. \quad (4.2)$$

We observe that (4.2) is equivalent to (4.3) below

$$r_p(\bar{y}) > 0 \implies \bar{y}_p = 0. \quad (4.3)$$

Indeed, since $r_p(y) \geq 0$ for all $p \in P$, implication (4.3) \implies (4.2) is clear. For the converse implication, suppose $r_p(\bar{y}) > 0$ for some $p \in P_w$. There exists some $p' \in P_w$ such that

$$r_p(\bar{y}) = \min_{j=1, \dots, l} (c_{p,j}(\bar{y}) - c_{p',j}(\bar{y})) > 0.$$

This shows that $c_p(\bar{y}) > c_{p'}(\bar{y})$ and implies $r_p(\bar{y}) > r_{p'}(\bar{y})$ because h is weakly monotone. In view of (4.2), we deduce $\bar{y}_p = 0$.

The following property demonstrates an important role of nonlinear scalarization: a feasible path flow is a weak vector equilibrium if and only if it is an *r-equilibrium*. Indeed, the function h being weakly monotone, the "if" part is clear. For the "only if" part, suppose \bar{y} is not *r-equilibrium*. By (4.3) there exist some $w \in W$ and $p' \in P_w$ such that $r_p(\bar{y}) > 0$ and $\bar{y}_p \neq 0$. Let $p' \in P_w$ be such that

$$r_p(\bar{y}) = \min_{j=1, \dots, l} (c_{p,j}(\bar{y}) - c_{p',j}(\bar{y})) > 0.$$

Then $c_p(\bar{y}) > c_{p'}(\bar{y})$, and hence \bar{y} is not a weak vector equilibrium.

3) *Augmented nonlinear scalarization.* We notice that an r -equilibrium is not necessarily a vector equilibrium. In order to obtain vector equilibria we consider a new scalarized relative cost function $R_p^\epsilon(y)$ defined by

$$R_p^\epsilon(y) = \max_{p' \in P_w} \min_{j=1, \dots, l} [c_{p,j}(y) - c_{p',j}(y) + \epsilon \sum_{j=1}^l (c_{p,j}(y) - c_{p',j}(y))]$$

for every $y \in K$ and $p \in P_w$ for some $w \in W$, where $\epsilon > 0$ is a small constant. Using an argument similar to the case of r_p , we may prove that the function h defined by

$$h(c_p(y)) = R_p^\epsilon(y)$$

is a monotone function. A feasible path flow \bar{y} is said to be R^ϵ -equilibrium if for every path $p \in P$, one has implication

$$R_p^\epsilon(\bar{y}) > 0 \implies \bar{y}_p = 0.$$

It can also be proven that a feasible path flow is a vector equilibrium if and only if there is some $\epsilon > 0$ such that it is an R^ϵ -equilibrium. Notice that the constant ϵ depends on each equilibrium flow. Therefore, in order to generate the set of vector equilibria by this approach one has to find the set of R^ϵ -equilibria for all $\epsilon > 0$.

4.1.2 Vector variational inequalities

Another approach in solving a multi-criteria traffic network equilibrium problem is to construct a suitable vector variational inequality the solutions of which are vector equilibria of the model, see [32]. We consider two vector variational inequality problems, denoted respectively (VI) and (WVI): Find $\bar{y} \in K$ such that

$$C(\bar{y})(y - \bar{y}) \not\leq 0 \text{ for all } y \in K$$

and

$$C(\bar{y})(y - \bar{y}) \not\leq 0 \text{ for all } y \in K.$$

The first variational inequality can be written as $C(\bar{y})(y - \bar{y}) \notin -\mathbb{R}_+^l \setminus \{0\}$ and the second one is written as $C(\bar{y})(y - \bar{y}) \notin \text{int}\mathbb{R}_+^l$. The following claim is also clear (see [32]): If \bar{y} solves (VI) (respectively (WVI)), then it is a vector equilibrium (respectively a weak vector equilibrium). The converse is not true, that is, a vector equilibrium (respectively weak vector equilibrium) is not necessarily a solution of (VI) (respectively (WVI)). By considering the set of the so-called elementary flows one is able to construct an equivalent vector variational inequality for the multi-criteria network equilibrium problem. Namely, let us denote by $K(\bar{y})$ the set of flows $y \in K$ such that $y - \bar{y}$ is elementary in the sense that there are $w \in W$ and $p, p' \in P_w$ such that $[y - \bar{y}]_{p''} = 0$ for $p'' \in P \setminus \{p, p'\}$ and $[y - \bar{y}]_p = -[y - \bar{y}]_{p'}$, where $[y - \bar{y}]_{p'}$ is the traffic load on path p' . It was proven in [38] that \bar{y} is a vector equilibrium if and only if it is a solution of the following vector quasi-variational inequality problem

$$C(\bar{y})(y - \bar{y}) \not\leq 0 \text{ for all } y \in K(\bar{y}).$$

A similar result is true for weak vector equilibria. Notice that finding a feasible flow satisfying the above mentioned vector variational inequality is hard and as far as we know, up to now there is no efficient method to solve it.

4.1.3 Two optimization problems

In this part we develop a new approach to solve a multi-criteria traffic equilibrium problem. Specifically we shall construct two optimization problems the solutions of which are exactly the vector equilibria of the network. The following notations will be used:

- $d[x, B]$ is the Euclidean distance from a point x to a set B in \mathbb{R}^l .
- $e \in \mathbb{R}^l$ is the vector of ones and $H_+ : \mathbb{R}^l \rightarrow \mathbb{R}^l$ is defined by

$$H_+(x) = \begin{cases} e & \text{if } x \geq 0 \\ 0 & \text{else.} \end{cases}$$

The function H_+ is vector version of the Heaviside Step function, which can also be expressed by

$$H_+(x) = \left(\prod_{i=1}^l h_+(x_i) \right) e \text{ for all } x \in \mathbb{R}^l,$$

where h_+ is the scalar Heaviside Step function, that is $h_+(t) = 1$ for $t \in \mathbb{R}_+$ and $h_+(t) = 0$ for $t < 0$. Let us define two functions on the set of feasible flows:

$$\begin{aligned} \phi(y) &:= \sum_{p \in P_w, w \in W} y_p d[c_p(y), \text{Min}(C_w(y))] \\ \psi(y) &:= \sum_{p \in P_w, w \in W} y_p \sum_{p' \in P_w} [c_p(y) - c_{p'}(y)]^T H_+[c_p(y) - c_{p'}(y)]. \end{aligned}$$

In general these functions are not continuous, but still have nice properties that we shall develop in the next section. We now show that the problems of minimizing them on the feasible set K are equivalent to the multi-criteria network equilibrium problem.

Theorem 4.1.1 *Let \bar{y} be a feasible flow. The following statements are equivalent:*

- (i) \bar{y} is a vector equilibrium.
- (ii) \bar{y} is an optimal solution of the following problem, denoted (P1):

$$\begin{aligned} &\text{minimize } \phi(y) \\ &\text{subject to } y \in K \end{aligned}$$

and the optimal value of this problem is zero.

- (iii) \bar{y} is an optimal solution of the following problem, denoted (P2):

$$\begin{aligned} &\text{minimize } \psi(y) \\ &\text{subject to } y \in K \end{aligned}$$

and the optimal value of this problem is zero.

Proof. We first prove that (i) and (ii) are equivalent. Let \bar{y} be a vector equilibrium. Since $\phi(y) \geq 0$ for every $y \in K$, it suffices to prove $\phi(\bar{y}) = 0$ in order to deduce (ii). In fact, for every $p \in P_w, w \in W$ one has either $c_p(\bar{y}) \in \text{Min}(C_w(\bar{y}))$ or there is some $p' \in P_w$ such that $c_p(\bar{y}) - c_{p'}(\bar{y}) \geq 0$. In the first case, $d[c_p(\bar{y}), \text{Min}(C_w(\bar{y}))] = 0$, and in the second case, $\bar{y}_p = 0$ by definition. Thus, the terms $\bar{y}_p d[c_p(\bar{y}), \text{Min}(C_w(\bar{y}))], p \in P$ are all equal to zero, which implies $\phi(\bar{y}) = 0$. Conversely, assume \bar{y} is an optimal solution of (P1) with $\phi(\bar{y}) = 0$. Since all terms in the sum defining ϕ are nonnegative, we have $y_p d[c_p(\bar{y}), \text{Min}(C_w(\bar{y}))] = 0$ for all $p \in P_w, w \in W$. If for some p and p' from $P_w, w \in W$ one has $c_p(\bar{y}) - c_{p'}(\bar{y}) \geq 0$, then $c_p(\bar{y}) \notin \text{Min}(C_w(\bar{y}))$. Hence $d[c_p(\bar{y}), \text{Min}(C_w(\bar{y}))] \neq 0$ and $\bar{y}_p = 0$. This proves that \bar{y} is a vector equilibrium.

Next we show equivalence between (i) and (iii). Let \bar{y} be a vector equilibrium. Since $\psi(y) \geq 0$ for every $y \in K$, as before, it suffices to prove $\psi(\bar{y}) = 0$ in order to deduce (iii). Let $p \in P_w, w \in W$. Consider the term $\sum_{p' \in P_w} \bar{y}_p [c_p(\bar{y}) - c_{p'}(\bar{y})]^T H_+[c_p(\bar{y}) - c_{p'}(\bar{y})]$, denoted b_p . If $c_p(\bar{y}) - c_{p'}(\bar{y}) \geq 0$ for some $p' \in P_w$, then by definition, $\bar{y}_p = 0$. If $c_p(\bar{y}) - c_{p'}(\bar{y}) = 0$ for some $p' \in P_w$, it is clear that the corresponding term of the above sum is zero. If $c_p(\bar{y}) - c_{p'}(\bar{y}) \not\geq 0$ for some $p' \in P_w$, then $H_+[c_p(\bar{y}) - c_{p'}(\bar{y})] = 0$. Therefore $b_p = 0$. Consequently, $\psi(\bar{y}) = 0$ as requested. Conversely, assume \bar{y} solves (P2) and $\psi(\bar{y}) = 0$. It follows that $b_p = 0$ for every $p \in P$. If for some p and p' from $P_w, w \in W$ one has $c_p(\bar{y}) - c_{p'}(\bar{y}) \geq 0$, then $[c_p(\bar{y}) - c_{p'}(\bar{y})]^T H_+[c_p(\bar{y}) - c_{p'}(\bar{y})] > 0$. Consequently, $\bar{y}_p = 0$. We deduce that \bar{y} is a vector equilibrium. \square

We note that problems (P1) and (P2) belong to the class of nonconvex problems under linear constraints. Their significance resides in the fact that the set of optimal solutions is exactly the set of vector equilibrium flows. Therefore these problems will be used to develop algorithms to generate vector equilibrium flows of the network.

Furthermore, by using the same method of proof we may establish a similar result for weak vector equilibria. Namely, a feasible \bar{y} is a weak vector equilibrium if and only if it solves each of the following problems

$$\begin{aligned} & \text{minimize } \phi^w(y) := \sum_{p \in P_w, w \in W} y_p d[c_p(y), \text{WMin}(C_w(y))] \\ & \text{subject to } y \in K \end{aligned}$$

and

$$\begin{aligned} & \text{minimize } \psi^w(y) := \sum_{p \in P_w, w \in W} y_p \sum_{p' \in P_w} [c_p(y) - c_{p'}(y)]^T H_+[c_p(y) - c_{p'}(y)] \\ & \text{subject to } y \in K \end{aligned}$$

and the optimal values are equal to zero, where H_+^w is defined by

$$H_+^w(x) = \begin{cases} e & \text{if } x > 0 \\ 0 & \text{else.} \end{cases}$$

And finally, we observe that the conclusion of Theorem 5.2.1 remains true when the function H_+ defining the objective function of (P2) is substituted by the function

$$\tilde{H}_+(x) = \left(\prod_{i=1}^l \tilde{h}_+(x_i) \right) e \text{ for all } x \in \mathbb{R}^l,$$

where $\tilde{h}_+(t) = h_+(t)$ for all $t \in \mathbb{R} \setminus \{0\}$ and $\tilde{h}_+(0) = 1/2$.

4.2 Generic differentiability and local calmness of the objective functions

The objective functions ϕ and ψ of problems (P1) and (P2) are not only nonconvex, but also not continuous as we have already noticed. This defect causes major difficulties in applying global optimization tools to solve (P1) and (P2). However, as we shall see in this section, these functions enjoy certain generic properties on continuity and differentiability. Specifically, we shall show that they are discontinuous or nondifferentiable only on a negligible subset.

We first need some preliminary lemmas in order to establish generic properties of the functions ϕ and ψ . Let b_1, \dots, b_k be vector-valued functions from \mathbb{R}^m to \mathbb{R}^l . For every $y \in \mathbb{R}^m$ and $j \in \{1, \dots, k\}$ denote

$$I(y) = \{i \in \{1, \dots, k\} : b_i(y) \in \text{Min}\{b_1(y), \dots, b_k(y)\}\}$$

$$J_j(y) = \{i \in \{1, \dots, k\} : b_j(y) \geq b_i(y)\}.$$

Thus, $y \mapsto I(y)$ and $y \mapsto J_j(y), j \in \{1, \dots, k\}$ are set-valued maps from \mathbb{R}^m to the topological space $\{1, \dots, k\}$ equipped with the discrete topology. In this case the map I is lower semi-continuous at y if there is a neighborhood U of y in \mathbb{R}^m such that $I(y') \supseteq I(y)$ for every $y' \in U$, and it is upper semi-continuous if there is a neighborhood U of y in \mathbb{R}^m such that $I(y') \subseteq I(y)$ for every $y' \in U$.

Lemma 4.2.1 *Assume that the functions b_1, \dots, b_k are continuous. The following statements hold.*

- (i) *The set-valued map $y \mapsto I(y)$ is lower semi-continuous at every point $y \in \mathbb{R}^m$ at which the vectors $b_1(y), \dots, b_k(y)$ are distinct from each other.*
- (ii) *The set-valued maps $y \mapsto J_j(y), j = 1, \dots, k$ are upper semi-continuous.*

Proof. To prove (i), let $i \in I(y)$. Since all vectors $b_j(y), j \in \{1, \dots, k\}$ are distinct, by definition one has $b_j(y) \notin b_i(y) - \mathbb{R}_+^l$ for $j \neq i$. Moreover, because b_1, \dots, b_k are continuous, there exists an open neighborhood U of y such that

$$b_j(y') \notin b_i(y') - \mathbb{R}_+^l \text{ for } j \neq i, y' \in U.$$

Hence $b_i(y') \in \text{Min}\{b_1(y'), \dots, b_k(y')\}$ for all $y' \in U$, which means $i \in I(y')$. We conclude $I(y) \subseteq I(y')$ for all $y' \in U$.

To prove (ii) we show that for any fixed index j , the map $y \mapsto J_j(y)$ is closed. Indeed, let $\{y^\nu\}_{\nu \geq 0}$ be a sequence in \mathbb{R}^m converging to y and $i_\nu \in J_j(y^\nu)$ converging to i . Then $i_\nu = i$ for ν sufficiently large, which means that $b_j(y^\nu) \geq b_i(y^\nu)$ for ν large. By continuity, when ν tends to ∞ , we obtain $b_j(y) \geq b_i(y)$, which proves that $i \in J_j(y)$. Since the image space of J_j is finite, in view of Proposition 1.4.8 of [2], the map J_j is upper semi-continuous. \square

We note that in general the map $y \mapsto I(y)$ is not upper semi-continuous and it is not lower semi-continuous without the condition that the vectors $b_1(y), \dots, b_k(y)$ are distinct from each other. Similarly the maps $y \mapsto J_j(y), j \in \{1, \dots, k\}$ are not lower semi-continuous. This observation is seen in the following example.

Example 4.2.2 *Let vector-valued functions b_1, b_2, b_3, b_4 from \mathbb{R}^3 to \mathbb{R}^2 be defined as follows*

$$b_1(y) = (y_1 + y_2, y_1 + y_3)^T,$$

$$b_2(y) = (y_2, y_2 + y_3)^T,$$

$$b_3(y) = (y_1 + y_2 + 2y_3, y_3)^T,$$

$$b_4(y) = (y_1 + 2y_2 + y_3, y_2)^T.$$

Consider the family $\{b_1(y), b_2(y), b_3(y)\}$ at the point $\bar{y} = (3, 3, 3)^T$. We have $b_1(\bar{y}) = (6, 6)^T, b_2(\bar{y}) = (3, 6)^T, b_3(\bar{y}) = (12, 3)^T$. By definition, $I(\bar{y}) = \{2, 3\}$. We claim that the map $y \mapsto I(y)$ is not upper semi-continuous at \bar{y} . Indeed, for $\nu > 1$, set $y^\nu = (3 - 1/\nu, 3 + 1/\nu, 3)^T$. Then y^ν converges to \bar{y} as ν tends to ∞ , while $I(y^\nu) = \{1, 2, 3\}$ for every $\nu > 1$.

For the family $\{b_2(y), b_3(y), b_4(y)\}$, the hypothesis of (i) of Lemma 4.2.1 does not hold at $\bar{y} = (3, 3, 3)^T$. We have $I(\bar{y}) = \{2, 3, 4\}$ and $J_3(\bar{y}) = \{3, 4\}$. However, for $y^\nu = (3, 3 + 1/\nu, 3 - 1/\nu)^T$ with $\nu > 1$, we have $I(y^\nu) = \{2, 3\}$ and $J_3(y^\nu) = \{3\}$ for all $\nu > 1$. Since y^ν converges to \bar{y} as ν tends to ∞ we conclude that the maps $y \mapsto I(y)$ and $y \mapsto J_4(y)$ are not lower semi-continuous at \bar{y} .

According to a generic theorem (Theorem 1.4.13 of [2]) the set-valued maps $y \mapsto J_j(y), j \in \{1, \dots, k\}$ and $y \mapsto I(y)$ are continuous on a dense subset of \mathbb{R}^n . Below we establish a stronger result which says that the set of discontinuity of these maps is nowhere dense.

Lemma 4.2.3 *Assume that the functions b_1, \dots, b_k are continuous. For every open set $U \subseteq \mathbb{R}^m$ there is an open subset U' such that*

- (i) $I(y) = I(y')$ for every $y, y' \in U'$
(ii) $J_j(y) = J_j(y')$ for every $y, y' \in U'$ and $j \in \{1, \dots, k\}$.

Proof. For each couple b_i and $b_j, i \neq j$, either $b_i(y) = b_j(y)$ for all $y \in U$, or there is some $y \in U$ such that $b_i(y) \neq b_j(y)$, in which case, due to continuity, there is a small neighborhood U_1 of y in U such that $b_i(y') \neq b_j(y')$ for all $y' \in U_1$. Applying this argument to all possible couples from the family $\{b_1, \dots, b_k\}$ we may assume the existence of an open subset U_2 in U and an index set $I \subseteq \{1, \dots, k\}$ such that the vectors $b_i(y'), i \in I$ are all distinct from each other and for every $i \in \{1, \dots, k\}$ there is some $j \in I$ such that $b_i(y') = b_j(y'), y' \in U_2$. Let $y \in U_2$. In view of Lemma 4.2.1 one find some neighborhood U_3 of y in U_2 such that $I(y) \subseteq I(y')$ for all $y' \in U_3$ with $I(y)$ and $I(y')$ being subsets of I . It follows from the definition of I that the latter statement is true for the family b_1, \dots, b_k . Let us now consider the set of all index sets $I(y') \subseteq \{1, \dots, k\}, y' \in U_3$. We find an index set $I(y')$ for some $y' \in U_3$ which is maximal with respect to inclusion. Apply again Lemma 4.2.1 to this y' we may find a smaller neighborhood $U' \subseteq U_3$ such that $I(y') \subseteq I(y'')$ for all $y'' \in U'$. But $I(y')$ was chosen to be maximal, we have $I(y') = I(y'')$ for all $y'' \in U'$ as requested.

To prove (ii), let us consider a point $y \in U$. In view of Lemma 4.2.1 (ii) for each $j \in \{1, \dots, k\}$ there exists an open set U_1 of y in U such that

$$J_j(y') \subseteq J_j(y) \text{ for all } y' \in U_1.$$

For $j = 1$, consider the family of index set $J_1(y'), y' \in U_1$ and let $y^1 \in U_1$ be a point at which $J_1(y^1)$ is minimal with respect to inclusion. Then there is an open set U_2 of y^1 in U_1 such that

$$J_1(y') \subseteq J_1(y^1) \text{ for all } y' \in U_2.$$

Because of the choice of y^1 , the above inclusion is equality. The same argument is applied to $J_2(\cdot)$ on U_2 . We obtain an open set $U_3 \subseteq U_2$ such that

$$\begin{aligned} J_1(y') &= J_1(y'') \\ J_2(y') &= J_2(y'') \text{ for all } y', y'' \in U_3. \end{aligned}$$

Continuing this process we arrive at an open set $U' \subseteq U$ in which

$$J_j(y') = J_j(y'') \text{ for all } y', y'' \in U' \text{ and } j = 1, \dots, k$$

as requested. \square

We recall that the Hausdorff distance between two bounded closed sets B and B' in \mathbb{R}^l is defined by

$$H(B, B') = \min\{t \geq 0 : B \subseteq B' + t\mathbb{B}, B' \subseteq B + t\mathbb{B}\} = \max\{\max_{x \in B} d(x, B'), \max_{x' \in B'} d(x', B)\},$$

where \mathbb{B} denotes the unit ball centered at 0 in \mathbb{R}^l . Now we are in position to present and prove the main result of this section.

Theorem 4.2.4 *Assume that the vector cost functions $c_{p_i}, i = 1, \dots, m$ are continuous (respectively locally Lipschitz or differentiable). Then every open set in \mathbb{R}^m contains an open subset where the objective functions ϕ and ψ of problems (P1) and (P2) are continuous (respectively locally Lipschitz or differentiable).*

Proof. Let U be an open set in \mathbb{R}^m . Let $w \in W$. In view of Lemma 4.2.3 there is an open subset $U_w \subseteq U$ and an index set $I \subseteq \{i \in \{1, \dots, m\} : p_i \in P_w\}$ such that

$$\text{Min}(C_w(y)) = \{c_{p_i}(y) : i \in I\} \text{ for all } y \in U_w.$$

Then for $y, y' \in U_w$ we have

$$\begin{aligned} H[\text{Min}(C_w(y)), \text{Min}(C_w(y'))] &\leq \max\{\max_{i \in I} d(c_{p_i}(y), \text{Min}(C_w(y'))), \\ &\quad \max_{i \in I} d(c_{p_i}(y'), \text{Min}(C_w(y)))\} \\ &\leq \max_{i \in I} \|c_{p_i}(y) - c_{p_i}(y')\|. \end{aligned}$$

If the functions $c_{p_i}, p_i \in P$ are continuous, then the set-valued map $y \mapsto \text{Min}(C_w(y))$ is also continuous on U_w with respect to the Hausdorff distance. In view of Theorem 1.4.16 [2] we deduce that the functions $d[c_{p_i}(y), \text{Min}(C_w(y))], p_i \in P_w$ are continuous on U_w . By choosing another O/D pair w' from W and applying the same argument as above on the open set $U_{w'}$ we obtain an open set $U' \subseteq U$ such that all functions $d[c_{p_i}(y), \text{Min}(C_w(y))], p_i \in P_w, w \in W$ are continuous, which implies the same property for ϕ on U' .

If the functions $c_{p_i}, p_i \in P$ are locally Lipschitz, then the set-valued map $y \mapsto \text{Min}(C_w(y))$ is also locally Lipschitz on U_w . The method of proof of Theorem 1.4.16 [2] can be applied to show that the functions $d[c_{p_i}(y), \text{Min}(C_w(y))], p_i \in P_w, w \in W$ are locally Lipschitz on U_w , and then one obtains an open set $U' \subseteq U$ on which ϕ is locally Lipschitz.

Assume now the functions $c_{p_i}, p_i \in P$ are differentiable. We consider the family C_w on the open set U_w . Then for every $p_i \in P_w$ and $y \in U_w$ one has

$$d[c_{p_i}(y), \text{Min}(C_w(y))] = \min\{d[c_{p_i}(y), c_{p_{i'}}(y)] : i' \in I\}.$$

If $c_{p_i}(y) \in \text{Min}(C_w(y))$ for all $y \in U_w$, then $d[c_{p_i}(y), \text{Min}(C_w(y))] = 0$ on U_w , and so this function is differentiable on U_w . If $c_{p_i}(y) \notin \text{Min}(C_w(y))$ for all $y \in U_w$, we denote by $I'(y)$ the set of indices $i' \in I$ such that the above distance is attained. We claim that the set-valued map $y \mapsto I'(y)$ is upper semi-continuous. Indeed, let $y \in U_w$ and let $\{y_\nu\}_{\nu \geq 0}$ be a sequence of elements in U_w converging to y . Let $i'_\nu \in I'(y_\nu)$. Since $I'(y_\nu) \subseteq I$ and I is finite, we may assume without loss of generality that $i'_\nu = i_0$ for some $i_0 \in I$ and for all $\nu \geq 0$. We have

$$d[c_{p_i}(y_\nu), c_{p_{i_0}}(y_\nu)] \leq d[c_{p_i}(y_\nu), c_{p_{i'}}(y_\nu)] \text{ for all } i' \in I.$$

By passing to the limit as ν tends to ∞ , we obtain

$$d[c_{p_i}(y), c_{p_{i_0}}(y)] \leq d[c_{p_i}(y), c_{p_{i'}}(y)] \text{ for all } i' \in I$$

which proves that $i_0 \in I'(y)$, and hence the map $I'(y)$ is upper semi-continuous. Let $y_0 \in U_w$ be a point such that $I'(y_0)$ is minimal with respect to inclusion among the index sets $I'(y), y \in U_w$. Then, there is an open neighborhood U_1 of y_0 in U_w such that $I'(y') = I'(y_0)$ for all $y' \in U_1$. By choosing any $i' \in I'(y_0)$ we obtain $d[c_{p_i}(y), \text{Min}(C_w(y))] = d[c_{p_i}(y), c_{p_{i'}}(y)] > 0$ for all $y \in U_1$. Hence $d[c_{p_i}(y), \text{Min}(C_w(y))]$ is differentiable on U_1 . Apply this argument to other paths and other O/D pairs to obtain an open set $U' \subseteq U$ where ϕ is differentiable.

Now we consider ψ . First we prove that for $p, p' \in P_w$ there is an open set $U' \subseteq U$ such that the function $H_+[c_p(y) - c_{p'}(y)]$ is constant on U' . Indeed, if $c_p(y) - c_{p'}(y) \geq 0$ for every $y \in U$, then by definition $H_+[c_p(y') - c_{p'}(y')] = e$ on U . If $c_p(y) - c_{p'}(y) \not\geq 0$ for some y , then it is clear that there is some neighborhood U' of y in U such that

$$c_p(y') - c_{p'}(y') \not\geq 0 \text{ for all } y' \in U'.$$

By definition $H_+[c_p(y') - c_{p'}(y')] = 0$ on U' . Applying the above argument to all $p, p' \in P_w$ and $w \in W$, we find an open set $U' \subseteq U$ on which the functions $H_+[c_p(y) - c_{p'}(y)]$ are constant. Hence the function ψ is continuous (respectively Lipschitz continuous or differentiable) on U' whenever the functions $c_{p_i}, i = 1, \dots, m$ are continuous (respectively Lipschitz continuous or differentiable). \square

For a path p_i in P let us denote $w(i)$ the O/D pair for which the set $P_{w(i)}$ contains this path. Thus, $w(i) = w(i')$ if the paths p_i and $p_{i'}$ connect the same O/D pair. We will also adopt a convention that for two vectors $b, b' \in \mathbb{R}^l$, $(b - b')/\|b - b'\| = 0$ if $b = b'$.

Theorem 4.2.5 *Assume that the vector cost functions $c_{p_i}, i = 1, \dots, m$ are differentiable. Then for every point y outside of some nowhere dense subset and for every path p_i , there exists a path $p_{\nu(i)}$ from $P_{w(i)}$ such that*

- (i) $c_{p_{\nu(i)}}(y) \in \text{Min}C_{w(i)}(y)$
- (ii) $d[c_{p_i}(y), \text{Min}C_{w(i)}(y)] = \|c_{p_i}(y) - c_{p_{\nu(i)}}(y)\|$
- (iii) ϕ is differentiable at y and its gradient is computed by

$$\nabla\phi(y) = \begin{pmatrix} \|c_{p_1}(y) - c_{p_{\nu(1)}}(y)\| \\ \dots \\ \|c_{p_m}(y) - c_{p_{\nu(m)}}(y)\| \end{pmatrix} + \sum_{i=1}^m y_{p_i} \begin{pmatrix} \frac{c_{p_i}(y) - c_{p_{\nu(i)}}(y)}{\|c_{p_i}(y) - c_{p_{\nu(i)}}(y)\|} \left(\frac{\partial c_{p_i}}{\partial y_{p_1}}(y) - \frac{\partial c_{p_{\nu(i)}}}{\partial y_{p_1}}(y) \right) \\ \dots \\ \frac{c_{p_i}(y) - c_{p_{\nu(i)}}(y)}{\|c_{p_i}(y) - c_{p_{\nu(i)}}(y)\|} \left(\frac{\partial c_{p_i}}{\partial y_{p_m}}(y) - \frac{\partial c_{p_{\nu(i)}}}{\partial y_{p_m}}(y) \right) \end{pmatrix}$$

Proof. Let U be any open set in K . According to the proof of Theorem 4.2.4 there are an open set $U' \subseteq U$ and index sets $I_{w(i)}, i = 1, \dots, m$ such that

$$d[c_{p_i}(y), \text{Min}C_{w(i)}(y)] = \|c_{p_i}(y) - c_{p_{i'}}(y)\|$$

for all $y \in U', i' \in I_{w(i)}, i \in \{1, \dots, m\}$. Choose $\nu(i)$ to be any i' from $I_{w(i)}$ for $i = 1, \dots, m$, we have

$$\phi(y) = \sum_{i=1}^m y_{p_i} \|c_{p_i}(y) - c_{p_{\nu(i)}}(y)\|$$

for all $y \in U'$. If $i \in I_{w(i)}$, then $\|c_{p_i}(y) - c_{p_{\nu(i)}}(y)\| = 0$ for all $y \in U'$. If $i \notin I_{w(i)}$, then $\|c_{p_i}(y) - c_{p_{\nu(i)}}(y)\| > 0$ for all $y \in U'$. In any case every term in the sum of the function ϕ is differentiable on U' . A direct calculation produces the formula for the gradient of ϕ given in (iii). \square

In the next theorem we set $\sum_{j \in J} (c_{p_i}(y) - c_{p_j}(y)) = 0$ if the index set J is empty and keep the notation $I_{w(i)}$ for the set of all indices j such that $p_j \in P_{w(i)}$ satisfying $d[c_{p_i}(y), \text{Min}C_{w(i)}(y)] = \|c_{p_i}(y) - c_{p_j}(y)\|$.

Theorem 4.2.6 *Assume that the vector cost functions $c_{p_i}, i = 1, \dots, m$ are differentiable. Then for every point y outside of some nowhere dense subset and for every path p_i , there exists a subset $J_i(y) \subseteq I_{w(i)}$ such that*

- (i) $c_{p_i}(y) \geq c_{p_j}(y)$ for every $j \in J_i(y)$
- (ii) $\psi(y) = \sum_{i=1}^m y_{p_i} \langle \sum_{j \in J_i(y)} (c_{p_i}(y) - c_{p_j}(y)), e \rangle$
- (iii) ψ is differentiable at y and its gradient is computed by

$$\nabla\psi(y) = \begin{pmatrix} \sum_{j \in J_1(y)} \langle c_{p_1}(y) - c_{p_j}(y), e \rangle \\ \dots \\ \sum_{j \in J_m(y)} \langle c_{p_m}(y) - c_{p_j}(y), e \rangle \end{pmatrix} + \sum_{i=1}^m y_{p_i} \begin{pmatrix} \sum_{j \in J_i(y)} \langle \frac{\partial c_{p_i}}{\partial y_{p_1}}(y) - \frac{\partial c_{p_j}}{\partial y_{p_1}}(y), e \rangle \\ \dots \\ \sum_{j \in J_i(y)} \langle \frac{\partial c_{p_i}}{\partial y_{p_m}}(y) - \frac{\partial c_{p_j}}{\partial y_{p_m}}(y), e \rangle \end{pmatrix}$$

Proof. We apply an argument similar to the proof of Theorem 4.2.5. Given any open set U , there are an open set $U' \subseteq U$ and index sets J_1, \dots, J_m such that for $y \in U'$, $i \in \{1, \dots, m\}$ and $p_j \in P_{w(i)}$ one has $c_{p_i}(y) - c_{p_j}(y) \geq 0$ if and only if $j \in J_i$. We deduce that

$$\psi(y) = \sum_{i=1}^m y_{p_i} \sum_{j \in J_i} \langle c_{p_i}(y) - c_{p_j}(y), e \rangle$$

for all $y \in U'$. Consequently, ψ is differentiable on U' . A direct calculation gives us the formula (iii) for the gradient of ψ . \square

To establish some more continuity properties of the objective functions we recall that a function f on K is said to be locally calm at $\bar{y} \in K$ if there are some constants $\delta > 0$ and $\kappa > 0$ such that $|f(y) - f(\bar{y})| \leq \kappa \|y - \bar{y}\|$ for all $y \in K$ with $\|y - \bar{y}\| \leq \delta$ (see [15]).

Proposition 4.2.7 *Assume that the vector cost functions c_{p_1}, \dots, c_{p_m} are continuous. Then the functions ϕ and ψ are continuous at every vector equilibrium. If in addition c_{p_1}, \dots, c_{p_m} are locally calm at a vector equilibrium, then ϕ and ψ are also locally calm there.*

Proof. Let $\bar{y} \in K$ be a vector equilibrium of G . Assume that c_{p_1}, \dots, c_{p_m} are continuous on K and locally calm at \bar{y} . The case where these functions are merely continuous is proven by a similar argument. We wish to show that there exist some $\delta > 0$ and $\kappa > 0$ such that $|\phi(y) - \phi(\bar{y})| \leq \kappa \|y - \bar{y}\|$ and $|\psi(y) - \psi(\bar{y})| \leq \kappa \|y - \bar{y}\|$ for every $y \in K$ with $\|y - \bar{y}\| \leq \delta$. Because ϕ and ψ take nonnegative values and as \bar{y} is equilibrium, by Theorem ??, $\phi(\bar{y}) = \psi(\bar{y}) = 0$, the above inequalities are equivalent to

$$\phi(y) \leq \kappa \|y - \bar{y}\| \tag{4.4}$$

and

$$\psi(y) \leq \kappa \|y - \bar{y}\|. \tag{4.5}$$

We establish (4.4) first. Recall that for $i \in \{1, \dots, m\}$, $w(i)$ denotes the O/D pair connected by the path p_i and $P_{w(i)}$ denotes the set of all paths connecting this O/D pair. We make also use of the following notations

$$\begin{aligned} I_0 &= \{i \in \{1, \dots, m\} : \bar{y}_{p_i} = 0\} \\ I_+ &= \{i \in \{1, \dots, m\} : \bar{y}_{p_i} > 0\} \\ D &= \max\{d_w : w \in W\} \\ T &= \max\{\|c_{p_i}(y)\| : y \in K, i = 1, \dots, m\}. \end{aligned}$$

The following inequalities are clear

$$y_{p_i} \leq \|y\| \leq D \tag{4.6}$$

$$d[c_{p_i}(y), \text{Min}(C_{w(i)}(y))] \leq 2T \tag{4.7}$$

for every $y \in K$ and $p_i \in P$. Consider the terms

$$f_i(y) := y_{p_i} d[c_{p_i}(y), \text{Min}(C_{w(i)}(y))], i = 1, \dots, m.$$

If $i \in I_0$, then (4.6) and (4.7) yield

$$|f_i(y) - f_i(\bar{y})| = f_i(y) \leq 2T y_{p_i} \leq 2T \|y - \bar{y}\|. \tag{4.8}$$

If $i \in I_+$, then we have $d[c_{p_i}(\bar{y}), \text{Min}(C_{w(i)}(\bar{y}))] = 0$. Consider the family of functions

$$Q := \{c_{p_j} \in C_{w(i)} : c_{p_j}(\bar{y}) = c_{p_i}(\bar{y})\}.$$

We have

$$(C_{w(i)}(\bar{y}) \setminus Q(\bar{y})) \cap (Q(\bar{y}) - \mathbb{R}_+^l) = \emptyset.$$

Because the set $C_{w(i)}(\bar{y}) \setminus Q(\bar{y})$ is finite and the set $Q(\bar{y}) - \mathbb{R}_+^l$ is closed, there exists some $\gamma > 0$ such that

$$[(C_{w(i)}(\bar{y}) + \gamma\mathbb{B}) \setminus (Q(\bar{y}) + \gamma\mathbb{B})] \cap (Q(\bar{y}) + \gamma\mathbb{B} - \mathbb{R}_+^l) = \emptyset.$$

By continuity there exists $\delta_i > 0$ such that

$$C_{w(i)}(y) \subseteq C_{w(i)}(\bar{y}) + \gamma\mathbb{B}$$

$$Q(y) \subseteq Q(\bar{y}) + \gamma\mathbb{B}$$

for every $y \in K$ with $\|y - \bar{y}\| \leq \delta_i$. We deduce for such y and for every $c_{p_j} \in Q$ that

$$[C_{w(i)}(y) \setminus Q(y)] \cap (c_{p_j}(y) - \mathbb{R}_+^l) = \emptyset. \quad (4.9)$$

Consider the set $\text{Min}(Q(y))$. It is nonempty because $Q(y)$ is finite. We claim that

$$\text{Min}(Q(y)) \subseteq \text{Min}(C_{w(i)}(y)). \quad (4.10)$$

Indeed, let $a \in \text{Min}(Q(y))$. For each $p_j \in P_{w(i)}$, if $c_{p_j}(y) \notin Q(y)$, then by (4.9), one has

$$c_{p_j}(y) \notin a - \mathbb{R}_+^l. \quad (4.11)$$

If $c_{p_j}(y) \in Q(y) \setminus \{a\}$, then (4.11) is true because a is an efficient element of $Q(y)$. By this, $a \in \text{Min}(C_{w(i)}(y))$ and (4.10) follows. Furthermore, by the calmness hypothesis there are some constants $\delta' > 0$ and $\kappa' > 0$ such that

$$\|c_{p_j}(y) - c_{p_j}(\bar{y})\| \leq \kappa' \|y - \bar{y}\| \quad (4.12)$$

for every $y \in K$ with $\|y - \bar{y}\| \leq \delta'$ and $j = 1, \dots, m$. By using (4.10) we obtain the following estimation

$$\begin{aligned} d[c_{p_i}(y), \text{Min}(C_{w(i)}(y))] &\leq d[c_{p_i}(y), \text{Min}(Q(y))] \\ &\leq \max_{c_{p_j} \in Q} \|c_{p_i}(y) - c_{p_j}(y)\| \\ &\leq \max_{c_{p_j} \in Q} \|c_{p_i}(y) - c_{p_i}(\bar{y}) + c_{p_j}(\bar{y}) - c_{p_j}(y)\| \\ &\leq 2\kappa' \|y - \bar{y}\|. \end{aligned}$$

This and (4.6) imply

$$|f_i(y) - f_i(\bar{y})| = f_i(y) \leq 2D\kappa' \|y - \bar{y}\| \quad (4.13)$$

for all $y \in K$ with $\|y - \bar{y}\| \leq \min(\delta_i, \delta')$. By setting $\delta = \min\{\min(\delta_i, \delta') : i \in I_+\}$ and $\kappa = 2T|I_0| + 2D\kappa'|I_+|$ we obtain (4.4) from (4.8) and (4.13).

To prove (4.5), we proceed in a similar manner. Consider the terms

$$g_i(y) := y_{p_i} \sum_{p_j \in P_{w(i)}} [c_{p_i}(y) - c_{p_j}(y)]^T H_+ [c_{p_i}(y) - c_{p_j}(y)], \quad i = 1, \dots, m.$$

Observe first that

$$0 \leq [c_{p_i}(y) - c_{p_j}(y)]^T H_+ [c_{p_i}(y) - c_{p_j}(y)] \leq 2lT,$$

for all $p_i, p_j \in P$ and $y \in K$. Therefore, if $i \in I_0$, then

$$|g_i(y) - g_i(\bar{y})| = g_i(y) \leq 2mlTy_{p_i} \leq 2mlT\|y - \bar{y}\|. \quad (4.14)$$

If $i \in I_+$, we have

$$\sum_{p_j \in P_{w(i)}} [c_{p_i}(\bar{y}) - c_{p_j}(\bar{y})]^T H_+ [c_{p_i}(\bar{y}) - c_{p_j}(\bar{y})] = 0,$$

which implies that for $p_j \in P_{w(i)} \setminus \{p_i\}$, either $c_{p_i}(\bar{y}) \not\leq c_{p_j}(\bar{y})$ or $c_{p_i}(\bar{y}) = c_{p_j}(\bar{y})$. In the first case, due to the continuity hypothesis, there exists some $\delta_i'' > 0$ such that $c_{p_i}(y) \not\leq c_{p_j}(y)$ for all $y \in K$ with $\|y - \bar{y}\| \leq \delta_i''$. In the second case, by the calmness hypothesis (4.12), we have

$$\begin{aligned} \|c_{p_i}(y) - c_{p_j}(y)\| &= \|c_{p_i}(y) - c_{p_i}(\bar{y}) + c_{p_j}(\bar{y}) - c_{p_j}(y)\| \\ &\leq 2\kappa'\|y - \bar{y}\| \end{aligned}$$

for all $y \in K$ with $\|y - \bar{y}\| \leq \delta'$. Set $\gamma_i = \min\{\delta', \delta_i''\}$. We deduce

$$|g_i(y) - g_i(\bar{y})| = g_i(y) \leq lD \sum_{p_j \in P_{w(i)}} \|c_{p_i}(y) - c_{p_j}(y)\| \leq 2lmD\kappa'\|y - \bar{y}\| \quad (4.15)$$

for $y \in K$ with $\|y - \bar{y}\| \leq \gamma_i$. It remains to choose $\delta = \min\{\gamma_i : i \in I_+\}$ and $\kappa = 2mlT|I_0| + 2mlD\kappa'|I_+|$ to obtain (4.5) from (4.14) and (4.15). The proof is complete. \square

4.3 Generating vector equilibrium flows

In this section we propose an algorithm based on Theorem 4.1.1 to generate a subset of vector equilibrium flows of the network we described in Section 2. This algorithm, denoted (A), consists of two procedures. The first procedure is to create a net of feasible flows with which the second procedure will start. The second procedure is aimed at solving problem (P2) which is equivalent to the network equilibrium problem as stated in Theorem 4.1.1 by starting from an initial point from the net obtained by the first procedure. Even if the vector cost functions are linear or differentiable, problem (P2) belongs to the class of nonconvex global optimization problems and is hard to solve. The main difficulty in solving this problem is caused by the fact that the index sets J_i in the definition of the function ψ (Theorem 4.2.6) change from point to point. To overcome this we modify Frank-Wolfe's reduced gradient method to find descent direction at each iteration in order to construct a decreasing sequence of feasible values.

4.3.1 Description of the algorithm (A)

Assume that W consists of r elements w_1, \dots, w_r in the network and for each pair w_i there are $|P_{w_i}|$ paths connecting its origin to its destination. We denote also $I_j = \{i \in \{1, \dots, m\} : p_i \in P_{w_j}\}$.

Step 0 (initialization). Choose a positive integer q and a tolerance level $\epsilon \geq 0$.

Procedure A1.

Step 1. Set $\delta_j = d_{w_j}/(q|P_{w_j}|)$, $j = 1, \dots, r$.

Step 2. Choose $(k_1, \dots, k_m)^T \in \mathbb{N}^m$ satisfying

$$\sum_{i \in I_j} k_i = q|P_{w_j}|, \quad j = 1, \dots, r.$$

Step 3. Store $y = (y_1, \dots, y_m)^T$ in Y^0 where

$$y_i = k_i \delta_j \quad \text{for } i \in I_j, j = 1, \dots, r$$

and return to Step 2 for other vectors (k_1, \dots, k_m) unless no one left.

Procedure A2.

Step 4. Choose a feasible flow y^0 from Y^0 to start. Set $k = 0$, $u^{k-1} = y^k$, $\alpha_{k-1} = \infty$, $Y^0 = Y^0 \setminus \{y^0\}$ and $E_q^e = \emptyset$.

Step 5. Compute $J_i(y^k) = \{i' \in \{1, \dots, m\} : p_{i'} \in P_{w(i)}, c_{p_i}(y^k) - c_{p_{i'}}(y^k) \geq 0\}$ for every $i \in \{1, \dots, m\}$. Set

$$\psi_k(y) := \sum_{i=1}^m y_{p_i} \sum_{i' \in J_i(y^k)} \langle c_{p_i}(y) - c_{p_{i'}}(y), e \rangle.$$

Compute $\psi_k(y^k)$.

If $\psi_k(y^k) \leq \epsilon$, store y^k in E_q^e and return to Step 4 until no element of Y^0 left.

Otherwise go to the next step.

Step 6. If $|\psi_k(y^k) - \alpha_{k-1}| \leq \epsilon$, go to Step 4 to choose another feasible solution from Y^0 to restart the procedure.

If $\psi_k(y^k) < \alpha_{k-1} - \epsilon$, set $\alpha_k = \psi_k(y^k)$ and go to Step 7.

If $\psi_k(y^k) > \alpha_{k-1} + \epsilon$, replace $y^k = y^{k-1} + (y^k - y^{k-1})/2$ and return to Step 5.

Step 7. Compute $\nabla \psi_k(y^k)$. Solve (P_k)

$$\begin{aligned} & \text{minimize} && u^T \nabla \psi_k(y^k) \\ & \text{subject to} && u \in K \\ & && |u_i - y_i^0| \leq \delta_{w(i)}, i = 1, \dots, m. \end{aligned}$$

Let u^k be an optimal solution.

If $|\psi_k(y^k) - \psi_k(u^k)| \leq \epsilon$, go to Step 4 to choose another feasible solution from Y^0 to restart the procedure until no element of Y^0 left.

Otherwise, set $y^{k+1} = u^k$. Update $k = k + 1$ and return to Step 5.

Some comments on the implementation of the algorithm are in order.

1) By using the notation $J_i(y)$ defined in Lemma 4.2.1 for the system of vector-valued functions c_{p_1}, \dots, c_{p_m} we have equality

$$\psi(y^k) = \sum_{i=1}^m y_{p_i}^k \sum_{i' \in J_i(y^k)} \langle c_{p_i}(y^k) - c_{p_{i'}}(y^k), e \rangle = \psi_k(y^k).$$

Note, however, that the functions ψ and ψ_k may differ from each other around y^k .

2) If the vector cost functions $c_{p_i}, i \in \{1, \dots, m\}$ are differentiable, then the function ψ_k is differentiable and its gradient is given in Theorem 4.2.6.

3) If $\psi_k(y^k) = 0$, then y^k is a vector equilibrium. If $\epsilon > 0$ and $|\psi_k(y^k)| \leq \epsilon$, we call y^k a

vector ϵ -equilibrium.

4) Problem (P_k) is considered as a linearized problem of (P2) at y^k . Therefore, when $\psi_k(u^k) = \psi_k(y^k)$ or $\psi(u^k) = \psi(y^k)$, the current solution y^k is called a *stationary point* of (P2), which includes also the case $\nabla\psi_k(y^k) = 0$. It is a local optimal solution of (P2) if in addition ψ is locally convex, but not a global optimal solution whenever the network has equilibrium because $\psi(y^k) = \psi_k(y^k) \neq 0$. When $|\psi_k(u^k) - \psi_k(y^k)| \leq \epsilon$ with $\epsilon > 0$, we call y^k an ϵ -stationary point of (P2). We notice that $\psi(y^k) = \psi_k(y^k)$, but $\psi(u^k)$ may differ from $\psi_k(u^k)$.

4.3.2 Convergence of the algorithm

We have already noticed that the objective function ψ is not continuous, and in general the limit of a sequence of vector ϵ -equilibrium flows is not necessarily a vector ϵ -equilibrium. However, the continuity of ψ at vector equilibria (Proposition 4.2.7) allows us to cover all vector equilibrium flows by the output E_q^ϵ . We recall that the outer limit of a sequence $\{A_q\}_{q \geq 1}$ of sets in \mathbb{R}^l , denoted $\limsup_{q \rightarrow \infty} A_q$, consists of all cluster points of sequences $\{a_q\}_{q \geq 1}$ with $a_q \in A_q$ for all $q \geq 1$. The sets of vector equilibrium flows and weak vector equilibrium flows are denoted respectively E and WE .

Theorem 4.3.1 *Assume the vector cost functions $c_{p_i}, i = 1, \dots, m$ are differentiable. For a fixed $\epsilon > 0$, the algorithm terminates after a finite number of iterations and the output E_q^ϵ contains vector ϵ -equilibrium flows. Moreover, the following inclusions hold true*

$$E \subseteq \bigcap_{\epsilon > 0} \limsup_{q \rightarrow \infty} E_q^\epsilon \subseteq WE.$$

Proof. By definition the output E_q^ϵ collects vector ϵ -equilibrium flows obtained at Step 5 of Procedure A2. Moreover, because the set Y^0 generated by Procedure A1 is finite, to prove the first part of the theorem it suffices to show that for a given $y^0 \in Y^0$ Procedure A2 terminates after a finite number of iterations. Assume this procedure does not terminate at iteration $k \geq 1$. We claim that

$$\psi_{i+1}(y^{i+1}) \leq \psi_i(y^i) - \epsilon \text{ for } i = 0, \dots, k-1. \quad (4.16)$$

Indeed, let us consider the case of y^0 and y^1 in details when $k \geq 1$. Because the procedure does not terminate, we have $\epsilon < \psi_0(y^0) < \infty - \epsilon$ and go to Step 7. By the same reason we have

$$\psi_0(y^0) - \psi_0(y^1) > \epsilon \quad (4.17)$$

and either of the following conditions is true

$$\psi_1(y^1) < \psi_0(y^0) - \epsilon \quad (4.18)$$

$$\psi_1(y^1) > \psi_0(y^0) + \epsilon. \quad (4.19)$$

If (4.19) holds, then according to our algorithm y^1 is replaced by $y^0 + (y^1 - y^0)/2$ until (4.18) is satisfied. Indeed, if (4.18) does not hold, then after some cycles y^1 is so close to y^0 that by Lemma 4.2.1 we may assume that $J_i(y^0) \supseteq J_i(y^1)$ for every $i = 1, \dots, m$. It follows that

$$\psi_0(y) - \psi_1(y) = \sum_{i=1}^m y_{p_i} \sum_{i' \in J_i(y^0) \setminus J_i(y^1)} \langle c_{p_i}(y) - c_{p_{i'}}(y), e \rangle \geq 0 \text{ for all } y \in K.$$

Combining this with (4.17) we deduce

$$\psi_0(y^0) > \psi_0(y^1) + \epsilon \geq \psi_1(y^1) + \epsilon$$

which contradicts (4.19). Thus, (4.18) is true. By applying the same argument to y^2, \dots, y^k we obtain (4.16), and by observing $\psi(y^i) = \psi_i(y^i)$ for $i = 0, \dots, k$ we deduce the following inequalities

$$\psi(y^k) < \psi(y^{k-1}) - \epsilon < \dots < \psi(y^0) - k\epsilon.$$

It is clear that for k sufficiently large, the procedure must be over because ψ takes nonnegative values only.

We now turn to the second part of the theorem. Let \bar{y} be a vector equilibrium. We know $\psi(\bar{y}) = 0$. By Proposition 4.2.7 there is some $\delta > 0$ sufficiently small such that $\psi(y) < \epsilon$ for every y with $\|y - \bar{y}\|_\infty < \delta$ (we are in a finite dimensional space, therefore the norm $\|\cdot\|_\infty$ is equivalent to the Euclidean norm $\|\cdot\|$). Choose q sufficiently large so that $\delta_{w(i)} < \delta/2$ for all $i = 1, \dots, m$. By the construction of Y^0 , there is $y^0 \in Y^0$ such that $\|y^0 - \bar{y}\|_\infty < \delta$. It follows that $\psi(y^0) < \epsilon$ which implies that y^0 is an ϵ -equilibrium and collected in E_q^ϵ . By this \bar{y} belongs to the outer limit of E_q^ϵ when q tends to infinity and for very fixed $\epsilon > 0$. This proves the first inclusion. To prove the second inclusion let \bar{y} be any element of the intersection of the outer limits of E_q^ϵ and suppose to the contrary that it is not a weak vector equilibrium. There exist some paths p_i and $p_j \in P_{w(i)}$ such that $c_{p_i}(\bar{y}) - c_{p_j}(\bar{y}) > 0$ and $\bar{y}_{p_i} > 0$. By continuity, there exists $\delta > 0$ and $t > 0$ such that $c_{p_i}(y) - c_{p_j}(y) > t$ and $y_{p_i} > t$ for all $y \in K$ satisfying $\|y - \bar{y}\| < \delta$. We have then $\psi(y) > t^2$ for all such y , and so y cannot belong to E_q^ϵ once $\epsilon < t^2$. This completes the proof. \square

We note that the inclusions in the above theorem are generally strict as it is shown by the two examples below. Actually in the first example we prove that the set of vector equilibrium flows is not closed, hence the first inclusion is strict, for the outer limit is a closed set. In the second example we present a network in which there is a weak vector equilibrium that is not a vector equilibrium. At this equilibrium the value of the function ψ is strictly positive, hence whenever ψ is continuous at it, there is no ϵ -equilibrium nearby with ϵ sufficiently small.

Example 4.3.2 Consider a network equilibrium problem with one pair of origin-destination nodes $w = (x, x')$, two criteria: travel time and travel cost, two available paths: $P_w = \{p_1, p_2\}$ with the travel demand $d_w = 10$. Assume that travel time and travel cost functions on the paths are given as follows

$$\begin{aligned} c_{p_1,1}(y) &= 5y_{p_1} + 3y_{p_2} & c_{p_2,1}(y) &= y_{p_1} + 3y_{p_2} + 9 \\ c_{p_1,2}(y) &= y_{p_1} + 2y_{p_2} & c_{p_2,2}(y) &= 2y_{p_1} + y_{p_2} + 6 \end{aligned}$$

Let $\nu \in \mathbb{N} \setminus \{0\}$. Define $y^\nu = (2 - 1/\nu, 8 + 1/\nu)^T$. It is a vector equilibrium because

$$c_{p_1}(y^\nu) - c_{p_2}(y^\nu) = \begin{pmatrix} 34 - 2/\nu \\ 18 + 1/\nu \end{pmatrix} - \begin{pmatrix} 35 + 2/\nu \\ 18 - 1/\nu \end{pmatrix} = \begin{pmatrix} -1 - 4/\nu \\ 2/\nu \end{pmatrix} \notin \mathbb{R}_+^2.$$

When ν tends to ∞ , y^ν converges to the feasible flow $\bar{y} = (2, 8)^T$. However, \bar{y} is not a vector equilibrium because $c_{p_2}(\bar{y}) = (35, 18)^T \geq c_{p_1}(\bar{y}) = (34, 18)^T$ while $\bar{y}_{p_2} = 8 \neq 0$.

Example 4.3.3 We consider the network with one pair of origin-destination nodes as described in Example 4.3.2. The travel demand is $d_w = 10$. The travel time and travel cost are given as below

$$\begin{aligned} c_{p_1,1}(y) &= 8y_{p_1} + 6y_{p_2} & c_{p_2,1}(y) &= 3y_{p_1} + y_{p_2} \\ c_{p_1,2}(y) &= 7y_{p_1} + y_{p_2} & c_{p_2,2}(y) &= 7y_{p_1} + y_{p_2} \end{aligned}$$

We observe that in this model every feasible flow is a weak vector equilibrium because $c_{p_1,1}(y) > c_{p_2,1}(y)$ and $c_{p_1,2}(y) = c_{p_2,2}(y)$ for every $y \geq 0$. Moreover, the function ψ is given by

$$\psi(y) = y_{p_1}(c_{p_1,1}(y) - c_{p_2,1}(y)) = y_{p_1}(5y_{p_1} + 5y_{p_2}) = 50y_{p_1}.$$

Consider the flow $\bar{y} = (1, 9)^T$ and choose $\epsilon \in (0, 1)$. Then every feasible flow $y = (y_{p_1}, y_{p_2})^T$ satisfying $|y_{p_1} - \bar{y}_{p_1}| \leq 1/2$ has $\psi(y) \geq 25$. Consequently, y cannot be an ϵ -equilibrium.

4.3.3 Numerical examples

In this section we present some numerical examples to illustrate the algorithm described in the preceding section. We use the MATLAB Optimization Toolbox to compute a representative set of vector equilibria and vector ϵ -equilibria for networks in which the cost functions are linear or nonlinear and the number of criteria is at least two. We also compare our method with the method by scalarization in each example. For the readers' convenience, let us briefly recall the scalarization method of [44] to find a vector equilibrium. It is to underline that [44] gave a method to find one vector equilibrium, but not the entire set of equilibria or a representative part of it. Without loss of generality we assume $l = 2$ and choose some $\alpha \in (0, 1)$. As we already discussed in Section 4.1, the scalarizing function $h(c_p(y)) = (\alpha, 1 - \alpha)c_p(y)$ is monotone, which ensures that every equilibrium of the network equipped with the scalarized cost functions $\pi_p(y) = h(c_p(y))$ is a vector equilibrium. In order to find an equilibrium of the scalarized network the author of [44] solves the following variational inequality problem: Find $\bar{y} \in K$ such that

$$\langle F(\bar{y}), y - \bar{y} \rangle \geq 0 \text{ for all } y \in K,$$

where $F(y) = ((\alpha, 1 - \alpha)c_{p_1}(y), \dots, (\alpha, 1 - \alpha)c_{p_m}(y))^T$. This later problem is solved by a modified projection method which is known to be convergent when F is Lipschitz continuous and monotone on K in the sense that

$$\langle F(y') - F(y), y' - y \rangle \geq 0, \text{ for all } y, y' \in K.$$

In general, for a weight vector $(\alpha, 1 - \alpha)$ the modified projection method yields one solution of the variational inequality problem which is also a vector equilibrium of the network. Therefore, in order to generate a set of vector equilibria by this method we use a finite family of equally distributed weight vectors in our examples. The number of weight vectors is chosen to be equal to the number of initial points of our method so that the comparison of computing time has sense.

Apparently the scalarization method has an advantage over our method that the function F has nice structure when the vector cost functions are linear or differentiable and one may hope to utilize a lot of existing optimization algorithms to obtain a solution of the variational inequality problem, while the function ψ used in our method is even not continuous. However, there are at least three drawbacks of the scalarization method. First, there exist vector equilibria that are not solutions of the variational inequality problem whatever weight vector $(\alpha, 1 - \alpha)$ be chosen. This is seen in Example 4.3.7. Hence there may be a large portion of vector equilibria that cannot be generated by the scalarization method. Second, even when the weight vectors are chosen to be equally spaced, there is no guarantee that the set of solutions of the variational inequality problems associated with these weight vectors is well distributed among the set of vector equilibria. Third, the modified projection method and some recent improvements [6, 26, 41, 56, 62] converge only under a certain monotonicity or generalized monotonicity property, which generally is not satisfied by network cost functions. This is illustrated in Example 4.3.6.

Example 4.3.4 Consider a network problem with one pair of origin-destination nodes $w = (x, x')$, two criteria: travel time and travel cost, two available paths: $P_w = \{p_1, p_2\}$ with the

travel demand $d_w = 30$. Assume that travel time and travel cost functions on the paths are linear and are given as follows

$$\begin{aligned} c_{p_1,1}(y) &= y_{p_1} + 2y_{p_2} & c_{p_2,1}(y) &= y_{p_1} + 6y_{p_2} \\ c_{p_1,2}(y) &= 6y_{p_1} + 2y_{p_2} & c_{p_2,2}(y) &= 6y_{p_1} + 8y_{p_2}. \end{aligned}$$

We tested our program for the zero tolerance $\epsilon = 0$ and $q = 2$ which yields 5 initial points. The results are displayed in the table below.

Initial point	Numbers of iterations	Vector equilibrium	CPU Time (seconds)
$(0, 30)^T$	3	—	0.256515
$(7.5, 22.5)^T$	4	—	
$(15, 15)^T$	3	—	
$(22.5, 7.5)^T$	100	—	
$(30, 0)^T$	1	$(30, 0)^T$	

We remark that starting from the first four initial points our algorithm finds no vector equilibrium because after some iterations either the value $\psi_k(y^k)$ is almost the same as α_{k-1} at Step 6 or the value $\psi_k(y^k)$ is almost the same as $\psi_k(u^k)$ at Step 7, which enforces us to go to Step 4 to choose another initial point to restart. Moreover, when the tolerance is strictly positive, the program runs much faster. For instance with $\epsilon = 10^{-4}$, the algorithm finds two vector ϵ -equilibria in 0.069872 seconds, one of which is $(30, 0)^T$ and the other is very near to it.

To apply the scalarization method of [44] we solve the following scalarized variational inequality problem: Find $\bar{y} \in K$ such that

$$\langle F(\bar{y})^T, y - \bar{y} \rangle \geq 0, \quad \forall y \in K,$$

where $F(y) = (\alpha(y_1 + 2y_2) + (1 - \alpha)(6y_1 + 2y_2), \alpha(3y_1 + 6y_2) + (1 - \alpha)(9y_1 + 8y_2))$ and $\alpha \in (0, 1)$. Since F is monotone and Lipschitz continuous on K , the modified projection method is convergent. For any weight vector $(\alpha, 1 - \alpha)$, it yields exactly the unique solution $(30, 0)^T$ of the problem. The CPU time to obtain the vector equilibrium by this method and the one by our method are almost the same.

In the following example, we shall see that the method by scalarization is slower in comparison with our method and it produces a not very well distributed set of equilibria.

Example 4.3.5 Consider the network described in Example 4.3.4 with one pair of origin-destination nodes, two paths and two criteria. We assume $d_w = 10$ and the travel time and travel cost functions on the paths p_1 and p_2 are given as follows

$$\begin{aligned} c_{p_1,1}(y) &= 3y_{p_1} + y_{p_2} & c_{p_2,1}(y) &= y_{p_1} + y_{p_2} \\ c_{p_1,2}(y) &= 5y_{p_1} + 3y_{p_2} & c_{p_2,2}(y) &= 3y_{p_1} + 5y_{p_2}. \end{aligned}$$

With $q = 10$ we have 21 initial points. Using the algorithm (A) we obtain 9 vector equilibria in 0.293389 seconds.

The scalarization method of [44] consists of solving the following variational inequality problem: Find $\bar{y} \in K$ such that

$$\langle F(\bar{y})^T, y - \bar{y} \rangle \geq 0, \quad \forall y \in K,$$

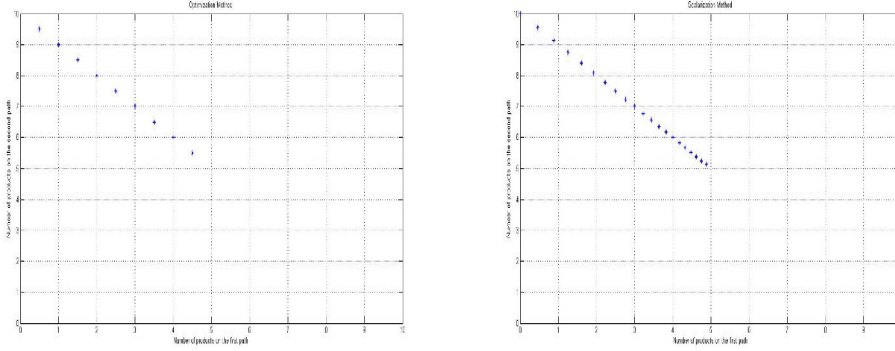
where

$$F(y)^T = \left(\begin{array}{l} \alpha(3y_{p_1} + y_{p_2}) + (1 - \alpha)(5y_{p_1} + 3y_{p_2}) \\ \alpha(y_{p_1} + y_{p_2}) + (1 - \alpha)(3y_{p_1} + 5y_{p_2}) \end{array} \right) \text{ with } \alpha \in (0, 1).$$

Since the function F is monotone and Lipschitz continuous, the modified projection method to solve this variational inequality problem is applied and produces a vector equilibrium for

each weight vector $(\alpha, 1 - \alpha)$ with $\alpha \in (0, 1)$. We performed this method for the initial point $(3, 7)^T$ and 21 weight vectors from the family $\left\{ \left(\frac{1}{21}, \frac{20}{21} \right), \left(\frac{2}{21}, \frac{19}{21} \right), \dots, \left(\frac{19}{21}, \frac{2}{21} \right), \left(\frac{20}{21}, \frac{1}{21} \right) \right\}$ and obtained 21 vector equilibria in 11.497309 seconds.

It can be seen that the set of all vector equilibria is the segment between $(0, 10)^T$ and $(5, 5)$. The vector equilibrium sets obtained by the two approaches are graphically presented below. The set obtained by our method is uniformly distributed, while the set obtained by the scalarization method is getting more condensed as we are approaching the point $(5, 5)^T$.

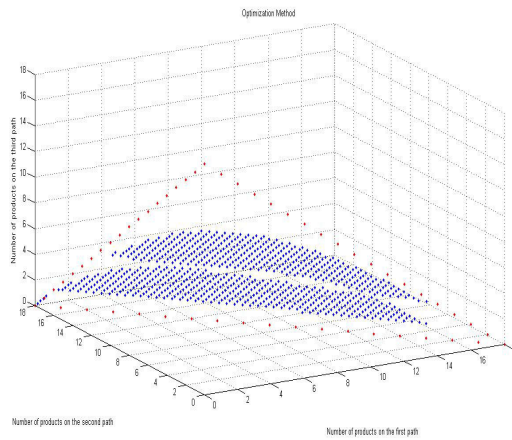


In the next example the travel time and travel cost functions are nonlinear. The scalarization method of [44] fails in finding even one equilibrium because the function defining the corresponding variational inequality is not monotone, while our algorithm works quite well and gives a satisfactory result.

Example 4.3.6 Consider a network problem with only one pair of origin-destination nodes $w = (x, x')$, two criteria: travel time and travel cost, three available paths: $P_w = \{p_1, p_2, p_3\}$ with the travel demand $d_w = 18$. Assume that travel time and travel cost functions on the paths are given as follows

$$\begin{aligned}
 c_{p_1,1}(y) &= y_{p_1}^2 + y_{p_2}^2 + y_{p_3}^3 & c_{p_2,1}(y) &= 8y_{p_1}y_{p_2} + y_{p_2}^2 \\
 c_{p_1,2}(y) &= 2y_{p_1} + 5y_{p_2} + 3y_{p_3} & c_{p_2,2}(y) &= y_{p_2} + 10y_{p_3} \\
 c_{p_3,1}(y) &= y_{p_1} + y_{p_2}^2 + y_{p_3}^3 & c_{p_3,2}(y) &= 10y_{p_3}^3.
 \end{aligned}$$

With $\epsilon = 0$ and $q = 20$ we have 1888 initial points. Using the algorithm (A) we obtained 1017 vector equilibria in 67.938183 seconds which are presented in the next figure.



The variational inequality problem of the scalarization method of [44] is the following: Find $\bar{y} \in K$ such that

$$\langle F(\bar{y})^T, y - \bar{y} \rangle \geq 0, \quad \forall y \in K,$$

where

$$F(y)^T = \left(\begin{array}{c} \alpha (y_{p_1}^2 + y_{p_2}^2 + y_{p_3}^3) + (1 - \alpha)(2y_{p_1} + 5y_{p_2} + 3y_{p_3}) \\ \alpha (8y_{p_1}y_{p_2} + y_{p_2}^2) + (1 - \alpha)(y_{p_2} + 10y_{p_3}) \\ \alpha (y_{p_1} + y_{p_2}^2 + y_{p_3}^3) + (1 - \alpha)10y_{p_3}^3 \end{array} \right) \text{ with } \alpha \in (0, 1).$$

It is clear that this function is Lipschitz, but not monotone on K . We run a program to solve it by starting from the point $(5, 5, 8)^T$ and taking weight vectors from the family

$$\left\{ \left(\frac{1}{1888}, \frac{1887}{1888} \right), \left(\frac{2}{1888}, \frac{1886}{1888} \right), \dots, \left(\frac{1886}{1888}, \frac{2}{1888} \right), \left(\frac{1887}{1888}, \frac{1}{1888} \right) \right\}.$$

However, after 549.238187 seconds no solution is found. In this example the modified projection algorithm does not converge.

We have also tested our program for a number of networks with more than three paths connecting an O/D pair. Our experience is that when the cost functions are linear both the scalarization method and our method work well with advantage in computing time of our method, and when the cost functions are not linear, in most cases the scalarization method is not convergent. For instance in one test with 6 paths we used 456 initial points and found 2 equilibrium flows in about two minutes, and in another test with 7 paths we used 6182 initial points and found 33 equilibrium flows in about 25 minutes. In both tests the scalarization method is not convergent and yields no equilibrium.

As we already mentioned at the beginning of this section, not every vector equilibrium of a multi-criteria network can be obtained by linear scalarization. Here is an example.

Example 4.3.7 In this example we consider a network problem with one pair of origin-destination nodes, two criteria, three available paths: $P_w = \{p_1, p_2, p_3\}$ and the travel demand $d_w = 7$. The travel time and travel cost functions on the paths are given as follows

$$\begin{aligned} c_{p_1,1}(y) &= 4y_{p_1} + 3 & c_{p_2,1}(y) &= 3y_{p_1} + 1 \\ c_{p_1,2}(y) &= 3y_{p_1} + y_{p_2} + y_{p_3} + 2 & c_{p_2,2}(y) &= 3y_{p_1} + 2y_{p_2} + y_{p_3} + 5 \\ & & c_{p_3,1}(y) &= 4y_{p_1} + 5 \\ & & c_{p_3,2}(y) &= 3y_{p_1} + y_{p_2} + y_{p_3} + 1. \end{aligned}$$

Let us consider the feasible flow $y^* = (3, 2, 2)^T$. It is clear that it is a vector equilibrium. We claim that whatever a vector weight $(\alpha, 1 - \alpha)$ be chosen, this flow cannot be an equilibrium of the network equipped with the scalarized cost functions $(\alpha, 1 - \alpha)c_i$, where $c_{p_1}(y) = (c_{p_1,1}(y), c_{p_1,2}(y))^T$, $c_{p_2}(y) = (c_{p_2,1}(y), c_{p_2,2}(y))^T$ and $c_{p_3}(y) = (c_{p_3,1}(y), c_{p_3,2}(y))^T$. In fact, let $\alpha \in (0, 1)$. We know that a feasible flow is an equilibrium of the network with the scalar cost functions $(\alpha, 1 - \alpha)c_{p_i}$, $p_i \in P_w$ if and only if it solves the following variational inequality problem

$$(\alpha, 1 - \alpha)C(\bar{y})(y - \bar{y})^T \geq 0, \quad \forall y \in K, \quad (4.20)$$

where

$$C(\bar{y}) = \left(\begin{array}{ccc} 4\bar{y}_{p_1} + 3 & 3\bar{y}_{p_1} + 1 & 4\bar{y}_{p_1} + 5 \\ 3\bar{y}_{p_1} + \bar{y}_{p_2} + \bar{y}_{p_3} + 2 & 3\bar{y}_{p_1} + 2\bar{y}_{p_2} + \bar{y}_{p_3} + 5 & 3\bar{y}_{p_1} + \bar{y}_{p_2} + \bar{y}_{p_3} + 1 \end{array} \right).$$

By setting $\bar{y} = y^* = (3, 2, 2)^T \in K$, inequality (4.20) is equivalent to

$$15(y_{p_1} - 3) + (20 - 10\alpha)(y_{p_2} - 2) + (14 + 3\alpha)(y_{p_3} - 2) \geq 0, \quad \forall y \in K.$$

It is easy to see that this latter inequality cannot hold for $\alpha \in (0, 1)$, for instance with $y = (4, 2, 1)^T$ one has $\alpha \leq 1/3$ in contradiction with $\alpha \geq 5/7$ when $y = (0, 0, 7)^T$. Thus, for any $\alpha \in (0, 1)$, y^* cannot be a solution of (4.20).

4.3.4 Smoothing the objective function

The function \tilde{H}_+ we defined in Section 4 admits the following analytic approximations

$$\tilde{H}_\nu(x) = \left(\prod_{i=1}^l \frac{1 + \tanh(\nu x_i)}{2} \right) e \quad \text{for } \nu \geq 1,$$

which produce also smooth approximations of the objective function ψ when the cost functions are smooth:

$$\psi_\nu(y) := \sum_{p \in P_w, w \in W} y_p \sum_{p' \in P_w} [c_p(y) - c_{p'}(y)]^T \tilde{H}_\nu [c_p(y) - c_{p'}(y)].$$

The corresponding optimization problems, denoted $(P2_\nu)$, are given below

$$\begin{aligned} & \text{minimize } \psi_\nu(y) \\ & \text{subject to } y \in K. \end{aligned}$$

Note that unlike ψ , the function ψ_ν may take negative values on K and an optimal solution of $(P2_\nu)$ is not necessarily a weak vector equilibrium.

Example 4.3.8 We consider the case $\nu = 1$. Examples for $\nu > 1$ are constructed in a similar manner. Suppose we have one pair w of origin-destination nodes, two paths p_1 and p_2 joining them, two criteria and the demand $d_w = 10$. Let $c_{p_2}(y) = (c_{p_2,1}(y), c_{p_2,2}(y))^T$ be a continuous vector cost function on p_2 . We define a vector cost function on p_1 to be $c_{p_1}(y) = c_{p_2}(y) + \alpha(y)(p_1, 1)^T$, where $\alpha(y)$ is the function given by

$$\alpha(y) = \begin{cases} 0.4 & \text{if } 0.3 \leq y_{p_1} \leq 10 \\ -0.4y_{p_1} + 1.6 & \text{if } 0 \leq y_{p_1} < 0.3. \end{cases}$$

Then for every flow $y = (y_{p_1}, y_{p_2})^T$ with $y_{p_1} + y_{p_2} = 10$ we have

$$\begin{aligned} \psi_1(y) &= 2\alpha(y)y_{p_1} \left(\frac{1 + \tanh(\alpha(y))}{2} \right)^2 - 2\alpha(y)y_{p_2} \left(\frac{1 + \tanh(-\alpha(y))}{2} \right)^2 \\ &= 2\alpha(y)y_{p_1} \left(\left(\frac{1 + \tanh(\alpha(y))}{2} \right)^2 + \left(\frac{1 + \tanh(-\alpha(y))}{2} \right)^2 \right) \\ &\quad - 20\alpha(y) \left(\frac{1 + \tanh(-\alpha(y))}{2} \right)^2. \end{aligned}$$

It is clear that $\psi_1(0.3, 9.7) < \psi_1(0, 10) < 0$, which proves that the optimal value of $(P2_1)$ is strictly negative and at every optimal solution $y = (y_{p_1}, y_{p_2})^T$ one has $y_{p_1} > 0$, while $c_{p_1}(y) > c_{p_2}(y)$. Hence optimal solutions of $(P2_1)$ cannot be weak vector equilibria.

Theorem 4.3.9 Assume that the cost functions c_{p_1}, \dots, c_{p_m} are continuous. If for every $\nu \geq 1, y^\nu \in K$ is an optimal solution of $(P2_\nu)$, then every cluster point \bar{y} of the sequence $\{y^\nu\}_{\nu \geq 1}$ is a weak vector equilibrium.

Proof. We keep the notations D and T defined in the proof of Proposition 4.2.7 and conduct the proof by contradiction. Suppose that \bar{y} is not a weak vector equilibrium. There exist some $w \in W$ and $p_i, p_j \in P_w$ such that

$$\bar{y}_{p_i} > 0 \text{ and } c_{p_i}(\bar{y}) - c_{p_j}(\bar{y}) > 0. \quad (4.21)$$

Claim 1. There exists some $\delta > 0$ and $\nu_0 \geq 1$ such that

$$\psi_\nu(y^\nu) \geq \delta \text{ for all } \nu \geq \nu_0. \quad (4.22)$$

To prove this claim let us fix a small $\epsilon > 0$ and consider any two paths $p, p' \in P_w$ for $w \in W$. Denote

$$\begin{aligned} \gamma_\nu(t) &:= \frac{1 + \tanh(\nu t)}{2} \\ \beta_{p,p'}(y^\nu) &:= y_p^\nu \left(\prod_{r=1}^l \gamma_\nu(c_{p,r}(y^\nu) - c_{p',r}(y^\nu)) \right) \langle c_p(y^\nu) - c_{p'}(y^\nu), e \rangle. \end{aligned}$$

We distinguish three possible cases concerning the components $c_{p,r}(\bar{y}) - c_{p',r}(\bar{y})$, $r = 1, \dots, l$ of the vector $c_p(\bar{y}) - c_{p'}(\bar{y})$.

Case 1: $c_{p,r}(\bar{y}) - c_{p',r}(\bar{y}) > 0$. By continuity there exists some $\nu_1 \geq 1$ such that

$$\begin{aligned} c_{p,r}(y^\nu) - c_{p',r}(y^\nu) &\geq \frac{1}{2} (c_{p,r}(\bar{y}) - c_{p',r}(\bar{y})), \\ \gamma_\nu(c_{p,r}(y^\nu) - c_{p',r}(y^\nu)) &\geq \frac{1}{4} \end{aligned} \quad (4.23)$$

for all $\nu \geq \nu_1$. If in addition $\bar{y}_p > 0$ and $c_p(\bar{y}) - c_{p'}(\bar{y}) > 0$, then there is $\nu'_1 \geq \nu_1$ such that

$$\begin{aligned} y_p^\nu &\geq \frac{1}{2} \bar{y}_p, \\ \beta_{p,p'}(y^\nu) &\geq \frac{1}{4^{l+1}} \bar{y}_p \langle c_p(\bar{y}) - c_{p'}(\bar{y}), e \rangle \end{aligned} \quad (4.24)$$

for every $\nu \geq \nu'_1$.

Case 2: $c_{p,r}(\bar{y}) - c_{p',r}(\bar{y}) = 0$. Again, by continuity there exists some $\nu_2 \geq 1$ such that

$$|c_{p,r}(y^\nu) - c_{p',r}(y^\nu)| \leq \epsilon \text{ for } \nu \geq \nu_2. \quad (4.25)$$

In particular, if $c_p(\bar{y}) - c_{p'}(\bar{y}) \geq 0$ and $c_p(\bar{y}) - c_{p'}(\bar{y}) \not\equiv 0$, then there exists $\nu'_2 \geq \nu_2$ such that

$$\beta_{p,p'}(y^\nu) \geq -Dl\epsilon \text{ for } \nu \geq \nu'_2. \quad (4.26)$$

Case 3: $c_{p,r}(\bar{y}) - c_{p',r}(\bar{y}) < 0$. As in the first case, one may find some $\nu_3 \geq 1$ such that

$$\begin{aligned} c_p(y^\nu) - c_{p'}(y^\nu) &\leq \frac{1}{2} (c_{p,r}(\bar{y}) - c_{p',r}(\bar{y})), \\ \gamma_\nu(c_{p,r}(y^\nu) - c_{p',r}(y^\nu)) &\leq \epsilon \text{ for } \nu \geq \nu_3. \end{aligned}$$

We deduce that if $c_p(\bar{y}) - c_{p'}(\bar{y}) \not\leq 0$, then there exists some $\nu'_3 \geq \nu_3$ such that

$$\beta_{p,p'}(y^\nu) \geq -2lDT\epsilon \text{ for } \nu \geq \nu'_3. \quad (4.27)$$

Let us now evaluate $\psi_\nu(y^\nu)$ by using (4.24-4.27). We have

$$\psi_\nu(y^\nu) = \sum_{p \in P_w, w \in W} \sum_{p' \in P_w} \beta_{p,p'}(y^\nu).$$

We decompose this sum into three sums \sum^1 , \sum^2 and \sum^3 . The first sum \sum^1 collects all terms $\beta_{p,p'}(y^\nu)$ with $p, p' \in P_w$ for some $w \in W$ such that $c_p(\bar{y}) - c_{p'}(\bar{y}) > 0$. The second sum \sum^2 collects all terms $\beta_{p,p'}(y^\nu)$ with $p, p' \in P_w$ for some $w \in W$ such that $c_p(\bar{y}) - c_{p'}(\bar{y}) \geq 0$, but $c_p(\bar{y}) - c_{p'}(\bar{y}) \not\equiv 0$, and the last sum \sum^3 contains all terms $\beta_{p,p'}(y^\nu)$ with $p, p' \in P_w$ for some $w \in W$ such that $c_p(\bar{y}) - c_{p'}(\bar{y}) \not\equiv 0$.

We may choose ν'_1, ν'_2 and ν'_3 depending on ϵ so that the relations (4.24-4.27) hold true for all components of the vectors $c_p(\bar{y}) - c_{p'}(\bar{y}), p, p' \in P_w$ and $w \in W$. It follows from the hypothesis (4.21) that the sum \sum^1 contains $\beta_{p_i, p_j}(y^\nu)$ and the other terms of this sum are positive for ν large. Moreover, in view of (4.24), there exists $\nu_4 \geq \nu'_1$ such that

$$\sum^1 \geq \frac{1}{4^{l+1}} \bar{y}_{p_i} \langle c_{p_i}(\bar{y}) - c_{p_j}(\bar{y}), e \rangle \quad \text{for } \nu \geq \nu_4. \quad (4.28)$$

Furthermore, because the sum \sum^2 contains at most m terms $\beta_{p,p'}(y^\nu)$ in which at most l components of $c_p(\bar{y}) - c_{p'}(\bar{y})$ are equal to zero and the others are strictly positive, in view of (4.26), there exists $\nu_5 \geq \nu'_2$ such that

$$\sum^2 \geq -mDl\epsilon \quad \text{for } \nu \geq \nu_5. \quad (4.29)$$

As to the sum \sum^3 we apply (4.27) to find $\nu_6 \geq \nu'_3$ such that

$$\sum^3 \geq -mlDT\epsilon \quad \text{for } \nu \geq \nu_6. \quad (4.30)$$

We set $\nu_0 = \max\{\nu_4, \nu_5, \nu_6\}$ and deduce from (4.28), (4.29) and (4.30) that

$$\psi_\nu(y^\nu) \geq \frac{1}{4^{l+1}} \bar{y}_{p_i} \langle c_{p_i}(\bar{y}) - c_{p_j}(\bar{y}), e \rangle - 3mlD(T+1)\epsilon \quad \text{for } \nu \geq \nu_0.$$

It remains to choose

$$\epsilon = \frac{\bar{y}_{p_i} \langle c_{p_i}(\bar{y}) - c_{p_j}(\bar{y}), e \rangle}{4^{l+2} 3mlD(T+1)}$$

to obtain q_0 and then set

$$\delta = \frac{1}{4^{l+2}} \bar{y}_{p_i} \langle c_{p_i}(\bar{y}) - c_{p_j}(\bar{y}), e \rangle$$

to satisfy Claim 1.

Claim 2. Let $y^* \in K$ be a vector equilibrium. Then $\lim_{\psi \rightarrow \infty} \psi(y^*) = 0$. Indeed, since y^* is a vector equilibrium, we have $y_p^* > 0, p \in P_w, w \in W$ only when either $c_p(\bar{y}) - c_{p'}(\bar{y}) = 0$ or $c_p(\bar{y}) - c_{p'}(\bar{y}) \not\equiv 0$ for every $p' \in P_w$. In the first case,

$$y_p^* [c_p(y^*) - c_{p'}(y^*)]^T \tilde{H}_\nu [c_p(y^*) - c_{p'}(y^*)] = 0. \quad (4.31)$$

In the second case, there is some $r \in \{1, \dots, l\}$ such that $c_{p,r}(y^*) - c_{p',r}(y^*) < 0$. Because

$$\lim_{\nu \rightarrow \infty} \gamma_\nu (c_{p,r}(y^*) - c_{p',r}(y^*)) = 0,$$

we deduce

$$\lim_{\nu \rightarrow \infty} y_p^* (c_p(y^*) - c_{p'}(y^*))^T \tilde{H}_\nu [c_p(y^*) - c_{p'}(y^*)] = 0. \quad (4.32)$$

It follows from (4.31) and (4.32) that $\lim_{\nu \rightarrow \infty} \psi_\nu(y^*) = 0$ as requested.

To complete the proof we choose a vector equilibrium $y^* \in K$, which evidently exists. By hypothesis $\psi_\nu(y^\nu) \leq \psi_\nu(y^*)$ for every $\nu \geq 1$. This implies

$$\liminf_{\nu \rightarrow \infty} \psi_\nu(y^\nu) \leq \lim_{\nu \rightarrow \infty} \psi_\nu(y^*).$$

and, in view of Claims 1 and 2, we arrive at a contradiction that $\delta \leq 0$. The proof is complete. \square

We have used the Matlab Global Optimization ToolBox to find optimal solutions of $(P2_\nu)$, $\nu = 1, \dots, 10$ for Examples 4.3.5 and 4.3.6. The numerical results are encountered below.

For Example 4.3.5 we chose three initial points $(0, 10)^T$, $(5, 5)^T$ and $(10, 0)^T$ and obtained corresponding optimal solutions, which are all vector equilibrium.

Initial point	$(0, 10)^T$	$(5, 5)^T$	$(10, 0)^T$
$\nu = 1$	$(3.3621, 6.6379)^T$	$(3.3624, 6.6376)^T$	$(3.3617, 6.6383)^T$
$\nu = 2$	$(2.4865, 7.5135)^T$	$(2.5128, 7.4872)^T$	$(3.6852, 6.3148)^T$
$\nu = 3$	$(1.7666, 8.2334)^T$	$(2.5127, 7.4873)^T$	$(3.9860, 6.0140)^T$
$\nu = 4$	$(1.4403, 8.5597)^T$	$(2.5127, 7.4873)^T$	$(4.1482, 5.8518)^T$
$\nu = 5$	$(1.1060, 8.8940)^T$	$(2.5126, 7.4874)^T$	$(4.2404, 5.7596)^T$
$\nu = 6$	$(1.0053, 8.9947)^T$	$(2.5137, 7.4863)^T$	$(4.2915, 5.7085)^T$
$\nu = 7$	$(1.1353, 8.8647)^T$	$(2.5135, 7.4865)^T$	$(4.3150, 5.6850)^T$
$\nu = 8$	$(0.7096, 9.2904)^T$	$(2.5134, 7.4866)^T$	$(4.3184, 5.6816)^T$
$\nu = 9$	$(0.6348, 9.3652)^T$	$(2.5133, 7.4867)^T$	$(4.3057, 5.6943)^T$
$\nu = 10$	$(0.6347, 9.3653)^T$	$(2.5132, 7.4868)^T$	$(4.2789, 5.7211)^T$

For Example 4.3.6 we chose three initial points $(12, 6, 0)^T$, $(0, 12, 6)^T$ and $(6, 6, 6)^T$. By using the first initial point, for $\nu = 1$, the optimal solution is $(7.0158, 2.9657, 8.0185)^T$ and for every $\nu = 2, \dots, 10$ we arrive at the same optimal solution $(7.0178, 2.9657, 8.0165)^T$. With the second and the third initial points the toolbox produced the same optimal solutions for every $\nu = 1, \dots, 10$:

$$(0.9675, 13.9921, 3.0404)^T \text{ and } (4.5458, 4.5046, 8.9496)^T.$$

Fortunately, they are all vector equilibria. This, however, is not true in general as the next example shows.

Example 4.3.10 Consider a network problem with one pair of origin-destination nodes, two criteria, six available paths and the travel demand $d_w = 100$. The travel time and travel cost functions on the paths are given as follows

$$\begin{aligned} c_{p1,1}(y) &= 2y_{p1}^2 + 7y_{p2}^2 & c_{p2,1}(y) &= 8y_{p1}y_{p2} + y_{p2}^2 & c_{p3,1}(y) &= y_{p1} + y_{p2}^2 + y_{p3}^2 \\ c_{p1,2}(y) &= 2y_{p1} + 5y_{p2} + 3y_3 & c_{p2,2}(y) &= 3y_{p2} + 10y_{p3}^2 & c_{p3,2}(y) &= y_{p3} + 10y_{p3}^3 \\ c_{p4,1}(y) &= y_{p1}^3 + y_{p4}y_5 & c_{p5,1}(y) &= y_{p2} + y_{p5} + y_{p6} & c_{p6,1}(y) &= y_{p3} + y_{p6} \\ c_{p4,2}(y) &= 2y_{p2} + y_{p4}^2 & c_{p5,2}(y) &= y_{p1}y_{p5} + y_{p3}y_{p6} + 10 & c_{p6,2}(y) &= y_{p1} + y_{p6}^2 \end{aligned}$$

With $q = 1$ we have 462 initial points to start with in finding optimal solutions of $(P2_\nu)$. For $\nu = 10$ the Global Optimization Toolbox produced 403 local optimal solutions and 59 global optimal solutions. Direct checking proves that there are 206 solutions which are vector equilibria. Meanwhile using the algorithm (A) for the problem (P2) with the same initial points we obtained one vector equilibrium $(37.5, 4.1667, 4.1667, 8.3333, 16.6667, 29.1667)^T$.

4.4 Robust equilibrium

In this section we are interested in vector equilibrium flows that are not affected by small perturbation of the cost functions. We shall write (G, C) to indicate the network G equipped

with the vector cost functions $c_p(y), p \in P$. Let \bar{y} be a feasible path flow. We say that it is a *robust vector equilibrium* if there is some $\epsilon > 0$ such that it is a vector equilibrium of the network (G, \tilde{C}) for any perturbed cost functions \tilde{c}_p satisfying $\|c_p(\bar{y}) - \tilde{c}_p(\bar{y})\| \leq \epsilon, p \in P$. In other words, \bar{y} is a robust vector equilibrium if there is $\epsilon > 0$ such that for every O/D pair $w \in W$, for every couple of paths $p, p' \in P_w$ and functions $\tilde{c}_p, \tilde{c}_{p'}$ with $\|c_p(\bar{y}) - \tilde{c}_p(\bar{y})\| \leq \epsilon$ and $\|\tilde{c}_{p'}(\bar{y}) - c_{p'}(\bar{y})\| \leq \epsilon$ one has implication

$$\tilde{c}_p(\bar{y}) - \tilde{c}_{p'}(\bar{y}) \geq 0 \implies \bar{y}_p = 0.$$

The biggest value of ϵ for which the above implication holds is called *radius of robustness* at \bar{y} . In robust optimization one considers a parametric maximization problem and defines a robust feasible solution as a solution that is feasible for all instances of parameter. A robust feasible solution is optimal if it maximizes the worst value of the objective function over the robust feasible set (see [4]). The concept of robust equilibrium we defined above can be considered as a particular version of robust solutions in optimization when the constraint set does not depend on the parameter and all instances of the objective function realize its maximum at the solution under consideration.

It is clear from the definition that a robust vector equilibrium is a vector equilibrium, but not all vector equilibria are robust. Moreover, as we know a multi-criteria network with continuous cost functions always has a vector equilibrium, but this is not true for robust equilibrium. For instance, the unique vector equilibrium $\bar{y} = (30, 0)^T$ of the network described in Example 4.3.4 is not robust. Indeed, a small perturbation of c_{p_1} by

$$c_1^\epsilon(y) = \begin{pmatrix} y_{p_1} + 2y_{p_2} + \epsilon \\ 6y_{p_1} + 2y_{p_2} \end{pmatrix} \text{ with } \epsilon > 0$$

makes \bar{y} non-equilibrium.

We shall make use of the following notation: for $w \in W$ and $y \in K$, $I_w(y)$ denotes the set of indices i such that $p_i \in P_w$ and $c_{p_i}(y) \in \text{Min}(C_w(y))$ and define a function ρ on K to be

$$\rho(y) := \sum_{p \in P_w, w \in W} y_p (d[c_p(y), \text{Min}(C_w(y))]) + \sum_{i \in I_w(y), p_i \neq p} \chi_{\{0\}}(\|c_p(y) - c_{p_i}(y)\|)$$

where $\chi_{\{0\}}$ is the characteristic function of $\{0\}$, that is $\chi_{\{0\}}(t) = 0$ if $t \neq 0$ and $\chi_{\{0\}}(t) = 1$ if $t = 0$.

Theorem 4.4.1 *Let $\bar{y} \in K$ be a vector equilibrium of G . The following statement are equivalent.*

- (i) \bar{y} is robust.
- (ii) \bar{y} is an optimal solution of the following optimization problem, denoted (P'_1)

$$\begin{aligned} & \text{minimize } \rho(y) \\ & \text{subject to } y \in K, \end{aligned}$$

and the optimal value of this problem is equal to zero.

- (iii) There exists an $\epsilon > 0$ such that for every $w \in W, p \in P_w$ with $\bar{y}_p > 0$, one has

$$(\tilde{c}_p(\bar{y}) - \mathbb{R}_+^l) \cap (\tilde{C}_w(\bar{y}) \setminus \{\tilde{c}_p(\bar{y})\}) = \emptyset$$

for all $\tilde{c}_{p_i}, p_i \in P_w$ satisfying $\|\tilde{c}_{p_i}(\bar{y}) - c_{p_i}(\bar{y})\| \leq \epsilon$.

Proof. To prove implication (i) \implies (ii), let $\bar{y} \in K$ be a robust vector equilibrium. We know by Theorem 4.1.1 that $\phi(\bar{y}) = 0$. Suppose to the contrary that $\rho(\bar{y}) > 0$. There are some $w \in W$ and $p \in P_w$ such that

$$\bar{y}_p \left(d[c_p(\bar{y}), \text{Min}(C_w(\bar{y}))] + \sum_{i \in I_w(\bar{y}), p_i \neq p} \chi_{\{0\}}(\|c_p(\bar{y}) - c_{p_i}(\bar{y})\|) \right) > 0.$$

In particular, $\bar{y}_p > 0$ and because $d[c_p(\bar{y}), \text{Min}(C_w(\bar{y}))] = 0$, there is some $i \in I_w(\bar{y})$ such that $p_i \neq p$ and $c_p(\bar{y}) = c_{p_i}(\bar{y})$. For every $\epsilon > 0$ define

$$\tilde{c}_{p_j} = \begin{cases} c_{p_j} & \text{if } p_j \neq p \\ c_p + (\epsilon/l)e & \text{if } p_j = p. \end{cases}$$

It is clear that $\|\tilde{c}_{p_j} - c_{p_j}\| \leq \epsilon, p_j \in P$ and \bar{y} is not a vector equilibrium of (G, \tilde{C}) because $\tilde{c}_p(\bar{y}) > \tilde{c}_{p_i}(\bar{y})$ while $\bar{y}_p > 0$. For implication (ii) \Rightarrow (iii) assume $\rho(\bar{y}) = 0$ and $\bar{y}_p > 0$ for some $p \in P_w, w \in W$. We deduce that $c_p(\bar{y}) \in \text{Min}(C_w(\bar{y}))$ and $\sum_{i \in I_w(\bar{y}), p_i \neq p} \chi_{\{0\}}(\|c_p(\bar{y}) - c_{p_i}(\bar{y})\|) = 0$.

In particular,

$$c_{p_i}(\bar{y}) \notin c_p(\bar{y}) - \mathbb{R}_+^l \text{ for all } p_i \in P_w, p_i \neq p. \quad (4.33)$$

Then we can find $\epsilon_p > 0$ such that

$$\tilde{c}_{p_i}(\bar{y}) \notin \tilde{c}_p(\bar{y}) - \mathbb{R}_+^l \quad (4.34)$$

for all $p_i \in P_w, p_i \neq p$, and $\|\tilde{c}_{p_i}(\bar{y}) - c_{p_i}(\bar{y})\| \leq \epsilon_p$ and $\|\tilde{c}_p(\bar{y}) - c_p(\bar{y})\| \leq \epsilon_p$. It remains to choose $\epsilon = \min\{\epsilon_p : p \in P, \bar{y}_p > 0\}$ to obtain (iii).

In order to prove the implication (iii) \Rightarrow (i), we observe that for the perturbed cost functions satisfying (iii), $\tilde{c}_p(\bar{y}) \in \text{Min}\tilde{C}_w(\bar{y})$ for every $p \in P_w$ with $\bar{y}_p > 0$. By this, \bar{y} is a vector equilibrium of the network (G, \tilde{C}) . Consequently, \bar{y} is a robust vector equilibrium. \square

Corollary 4.4.2 *Assume that \bar{y} is a vector equilibrium and for every $w \in W$ the elements $c_{p_i}(\bar{y}), i \in I_w(\bar{y})$ are all distinct from each other. Then \bar{y} is a robust vector equilibrium.*

Proof. Under the hypothesis of the corollary, one has

$$\sum_{i \in I_w(\bar{y}), p_i \neq p} \chi_{\{0\}}(\|c_p(\bar{y}) - c_{p_i}(\bar{y})\|) = 0$$

for every $p \in P$. Therefore, $\rho(\bar{y}) = 0$ and by Theorem 4.4.1, \bar{y} is robust. \square

Let \bar{y} be a given feasible flow of G . We denote by $\mathcal{C}(\bar{y})$ the set of $l \times m$ -matrices of continuous cost functions on K such that \bar{y} is a vector equilibrium of $(G, C), C \in \mathcal{C}(\bar{y})$. It is clear that $\mathcal{C}(\bar{y})$ is nonempty, for it contains all constant functions. We consider the subset $\mathcal{C}_R(\bar{y})$ consisting of all $C \in \mathcal{C}(\bar{y})$ such that \bar{y} is a robust vector equilibrium of (G, C) . It is clear that $\mathcal{C}_R(\bar{y})$ is a proper subset of $\mathcal{C}(\bar{y})$, for \bar{y} is not robust of (G, C) when C is a constant cost function and G has at least two paths joining an O/D pair. We shall see that $\mathcal{C}_R(\bar{y})$ is dense in $\mathcal{C}(\bar{y})$.

Lemma 4.4.3 *Let $q^1, \dots, q^m \in \mathbb{R}^l$ with $q^1 = \dots = q^k \in \text{Min}\{q^1, \dots, q^m\}$ for some $k \leq m$ and $q^k \neq q^j$ for $j > k$. Then for every $\epsilon > 0$ there exists $\tilde{q}^1, \dots, \tilde{q}^k$ such that*

- (i) $\|\tilde{q}^i - q^i\| \leq \epsilon$ for $i = 1, \dots, k$
- (ii) $\tilde{q}^i - \tilde{q}^j \notin \mathbb{R}_+^l \cup (-\mathbb{R}_+^l), i, j \in \{1, \dots, k\}, i \neq j$
- (iii) $\text{Min}\{\tilde{q}^1, \dots, \tilde{q}^k, q^{k+1}, \dots, q^m\} = \{\tilde{q}^1, \dots, \tilde{q}^k\} \cup ((\text{Min}\{q^1, \dots, q^m\}) \setminus \{q^1, \dots, q^k\})$.

Proof. Let $\epsilon > 0$ be given. As before $e^i \in \mathbb{R}^l$ denotes the unit i th vector and e the vector of ones. Since $q^1 = \dots = q^k \neq q^j$ for $j > k$, there exists a positive $\epsilon' \leq \epsilon/(3l)$ such that

$$q^i - \epsilon' e \notin q^j - \mathbb{R}_+^l \text{ for all } q^j \in \text{Min}\{q^1, \dots, q^m\}, j > k, i = 1, \dots, k. \quad (4.35)$$

Define

$$\tilde{q}^i = q^i - \epsilon' e + \epsilon' \left(\frac{i}{k} e^1 + \left(1 - \frac{i}{k}\right) e^l \right), \quad i = 1, \dots, k.$$

Then

$$\|\tilde{q}^i - q^i\| = \epsilon' \| -e + \frac{i}{k} e^1 + \left(1 - \frac{i}{k}\right) e^l \| \leq 3l\epsilon' \leq \epsilon,$$

which prove (i). Moreover for $i, j \in \{1, \dots, k\}, i \neq j$ one has

$$\tilde{q}^i - \tilde{q}^j = \epsilon' \left(\frac{i-j}{k} e^1 + \frac{j-i}{k} e^l \right) \notin \mathbb{R}_+^l \cup (-\mathbb{R}_+^l)$$

which is (ii). Finally, since $q^i \in \text{Min}\{q^1, \dots, q^m\}$ we have $q^j \notin q^i - \mathbb{R}_+^l$ and hence $q^j \notin \tilde{q}^i - \mathbb{R}_+^l$ for all $j > k$. This and (ii) prove that $\tilde{q}^i \in \text{Min}\{\tilde{q}^1, \dots, \tilde{q}^k, q^{k+1}, \dots, q^m\}, i = 1, \dots, k$. For $j > k$ such that $q^j \in \text{Min}\{q^1, \dots, q^m\}$ one observes that

$$q^j - \mathbb{R}_+^l \subseteq q^j + \epsilon' \left(\frac{i}{k} e^1 + \left(1 - \frac{i}{k}\right) e^l \right) - \mathbb{R}_+^l$$

and deduces from (4.35) that $\tilde{q}^i \notin q^j - \mathbb{R}_+^l$, for $i \in \{1, \dots, l\}$. By this q^j belongs to the set $\text{Min}\{\tilde{q}^1, \dots, \tilde{q}^k, q^{k+1}, \dots, q^m\}$.

For $j > k$ such that $q^j \notin \text{Min}\{q^1, \dots, q^m\}$ we find some $q^i \in \text{Min}\{q^1, \dots, q^m\}$ such that $q^i \leq q^j$. If $i > k$, then it is clear that $q^j \notin \text{Min}\{\tilde{q}^1, \dots, \tilde{q}^k, q^{k+1}, \dots, q^m\}$. If $i \leq k$, then we have $\tilde{q}^i \leq q^i \leq q^j$, which again implies that $q^j \notin \text{Min}\{\tilde{q}^1, \dots, \tilde{q}^k, q^{k+1}, \dots, q^m\}$. By this (iii) follows. \square

Corollary 4.4.4 *Let \bar{y} be a feasible flow of G . Then the set $\mathcal{C}_R(\bar{y})$ is open and dense in $\mathcal{C}(\bar{y})$.*

Proof. The fact that $\mathcal{C}_R(\bar{y})$ is open is immediate from the definition of robust vector equilibrium. To prove the density let $\epsilon > 0$ and assume \bar{y} is a vector equilibrium of (G, C) for some $C \in \mathcal{C}(\bar{y})$, but it is not robust. In view of Theorem 4.4.1 there exist some $w \in W$ and $p_i \in P_w$ such that $\bar{y}_{p_i} > 0$ and the set $\{j \in I_w(\bar{y}) : c_{p_i}(\bar{y}) = c_{p_j}(\bar{y}) \in \text{Min}C_w(\bar{y})\}$ consists of at least two elements. Applying Lemma 4.4.3 to the set $C_w(\bar{y})$ we find cost functions $\tilde{c}_p, p \in P$ such that $\|\tilde{c}_p(\bar{y}) - c_p(\bar{y})\| \leq \epsilon$ for all $p \in P$, all elements of $\text{Min}C_w(\bar{y})$ are distinct from each other and $I_w(\bar{y})$ is unchanged for $\tilde{C}(\bar{y})$. We deduce that $d[\tilde{c}_p(\bar{y}), \text{Min}\tilde{C}(\bar{y})] = 0$ and $\chi_{\{0\}}(\|\tilde{c}_{p_i}(\bar{y}) - \tilde{c}_{p_j}(\bar{y})\|) = 0$ for $j \neq i$ and $j \in I_w(\bar{y})$. This argument applied to all path on which the flow \bar{y} has a nonzero component, we deduce

$$\tilde{\rho}(y) = \sum_{p \in P_w, w \in W} \bar{y}_p \left(d[\tilde{c}_p(\bar{y}), \text{Min}\tilde{C}_w(\bar{y})] + \sum_{i \in I_w(\bar{y}), p_i \neq p} \chi_{\{0\}}(\|\tilde{c}_p(\bar{y}) - \tilde{c}_{p_i}(\bar{y})\|) \right) = 0.$$

which, in view of Theorem 4.4.1, implies that \bar{y} is robust for (G, \tilde{C}) . \square

When a robust vector equilibrium $\bar{y} \in K$ is given, we wish to know how far we can perturb the cost function C so that \bar{y} remains equilibrium for the perturbed costs. In other words we wish to find the radius of robustness at this equilibrium. For a feasible flow $\bar{y} \in K$ we denote

$$I_w^+(\bar{y}) = \{i \in I_w(\bar{y}) : \bar{y}_{p_i} > 0\}.$$

Corollary 4.4.5 *Let $\bar{y} \in K$ be a robust vector equilibrium. Then the radius of robustness at \bar{y} is computed by*

$$r(\bar{y}) = \frac{\sqrt{l}}{2} \min_{w \in W, i \in I_w^+(\bar{y})} \min_{p' \in P_w \setminus \{p_i\}} \|(c_{p'}(\bar{y}) - c_{p_i}(\bar{y}))^+\|$$

where $(c_{p'}(\bar{y}) - c_{p_i}(\bar{y}))^+$ denotes the positive part of the vector $c_{p'}(\bar{y}) - c_{p_i}(\bar{y})$.

Proof. It follows from the proof of Theorem 4.4.1 that \bar{y} is a vector equilibrium of (G, \tilde{C}) as soon as (4.34) is true which is equivalent to

$$(\tilde{c}_{p_i}(\bar{y}) - \tilde{c}_p(\bar{y}))^+ \neq 0,$$

where $p \in P_w$ with $\bar{y}_p > 0$ and $p_i \in P_w$. Let $\epsilon < r(\bar{y})$ and $\|\tilde{c}_{p_i}(\bar{y}) - c_{p_i}(\bar{y})\| \leq \epsilon$, $\|\tilde{c}_p(\bar{y}) - c_p(\bar{y})\| \leq \epsilon$, one has

$$(\tilde{c}_{p_i}(\bar{y}) - \tilde{c}_p(\bar{y}))^+ \geq [(c_{p_i}(\bar{y}) - \epsilon e) - (c_p(\bar{y}) + \epsilon e)]^+ \geq 0,$$

proving that \bar{y} is a vector equilibrium of (G, \tilde{C}) .

Let $\epsilon > r(\bar{y})$ and let

$$r(\bar{y}) = \frac{\sqrt{l}}{2} \|(c_{p'}(\bar{y}) - c_{p_i}(\bar{y}))^+\|$$

for some $i \in I_w^+(\bar{y})$, $p' \in P_w \setminus \{p_i\}$. Define a perturbed cost \tilde{C} by

$$\tilde{c}_p(\bar{y}) = \begin{cases} c_p(\bar{y}) & \text{if } p \neq p', p \neq p_i \\ c_{p_i}(\bar{y}) + (\epsilon/\sqrt{l})e & \text{if } p = p_i \\ c_{p'}(\bar{y}) - (\epsilon/\sqrt{l})e & \text{if } p = p'. \end{cases}$$

Then $\|\tilde{c}_p(\bar{y}) - c_p(\bar{y})\| \leq \epsilon$ for all $p \in P$ and $\tilde{c}_{p'}(\bar{y}) - \tilde{c}_{p_i}(\bar{y}) = c_{p'}(\bar{y}) - c_{p_i}(\bar{y}) - 2\epsilon \leq 0$. Consequently $\tilde{c}_{p_i}(\bar{y}) \notin \text{Min}\tilde{C}_w(\bar{y})$. This and the fact that $\bar{y}_{p_i} > 0$ implies that $\hat{\phi}(\bar{y}) > 0$. By Theorem 4.1.1, \bar{y} is not a vector equilibrium of (G, \tilde{C}) . \square

We complete this section by presenting an algorithm to find robust vector equilibria among the vector equilibria obtained by the algorithm (A) of Section 5 and to compute the radius of robustness at it.

Description of the algorithm

We denote E_q the set of vector equilibria, $S \times R = \{(\bar{y}, r(\bar{y})) : \bar{y} \text{ is robust equilibrium and } r(\bar{y}) \text{ is radius of robustness corresponding with robust equilibrium}\}$.

Step 1. Choose a vector equilibrium \bar{y}^1 from E_q to start. Set $k = 1$, $S = \emptyset$ and $R = \emptyset$.

Step 2. Determine $I_w(\bar{y}^k)$.

Calculate

$$\rho(\bar{y}^k) = \sum_{p \in P_w, w \in W} \bar{y}_p^k \sum_{i \in I_w(\bar{y}^k), p_i \neq p} \chi_{\{0\}}(\|c_p(\bar{y}^k) - c_{p_i}(\bar{y}^k)\|).$$

If $\rho(\bar{y}^k) \neq 0$, then return to Step 1 and choose another vector equilibrium from E_q to restart the algorithm until no element of E_q left.

If $\rho(\bar{y}^k) = 0$, then determine $I_w^+(\bar{y}^k)$ and calculate

$$r(\bar{y}^k) = \frac{\sqrt{l}}{2} \min_{w \in W, i \in I_w^+(\bar{y}^k)} \min_{p' \in P_w \setminus \{p_i\}} \|(c_{p'}(\bar{y}^k) - c_{p_i}(\bar{y}^k))^+\|.$$

Store $(\bar{y}^k, r(\bar{y}^k))$ in $S \times R$ and go to Step 1 until no element of E_q left.

We tested this algorithm for Examples 4.3.4 and 4.3.5. In Example 4.3.4 the unique vector equilibrium $(30, 0)^T$ is not robust. Therefore, $S = \emptyset$ and $R = \emptyset$. In Example 4.3.5, the obtained robust vector equilibria and radius of robustness are given in the table below.

Robust vector equilibria	Radius of robustness	Robust vector equilibria	Radius of robustness
$(0.5, 9.5)^T$	0.7071	$(3, 7)^T$	4.2426
$(1, 9)^T$	1.4142	$(3.5, 6.5)^T$	4.2426
$(1.5, 8.5)^T$	2.1213	$(4, 6)^T$	2.8284
$(2, 8)^T$	2.8284	$(4.5, 5.5)^T$	1.4142
$(2.5, 7.5)^T$	3.5355		

Equilibrium in a multi-criteria traffic network with capacity constraints

The purpose of this chapter is to study a single-product multi-criteria traffic network with capacity constraints. We construct an optimization problem the solutions of which are exactly the set of equilibria of the model. We establish some important generic continuity and differentiability properties of the objective function and give the formula to calculate the gradient of the objective function. Then we apply the algorithm proposed in Chapter 4 with some modifications in order to obtain a subset of optimal solutions which are equilibria of our model. Numerical examples are also presented to illustrate our approach.

5.1 Single-product multi-criteria traffic network with capacity constraints

We consider a traffic network $G = [N, A, W]$ that consists of a set of nodes N , a set of n directed arcs or links $A = \{a_1, \dots, a_n\}$ and a set W of r origin/destination (O/D for short) pairs of nodes $w = (x, x')$ with $x, x' \in N$ such that there is a path from x to x' . For a pair of nodes $w = (x, x')$, the set of available paths from the origin x to the destination x' is denoted by P_w , and the set of all available paths of the network is denoted by

$$P = \{p_1, \dots, p_m\} = \cup_{w \in W} P_w.$$

As in the preceding chapter, y_p denotes the traffic flow on path $p \in P$. We assume throughout this chapter that the demand d_w of the traffic flow for each O/D pair $w \in W$ is fixed and there are capacity constraints on each path of the network. Namely, for every $p \in P$ we have two nonnegative numbers $l_p \in \mathbb{R}$ and $u_p \in \mathbb{R}$ with $l_p < u_p$ that represent the lower and the upper capacity constraint on the path p . A path flow y is said to be feasible if it satisfies the capacity constraints and the conservation of flows equations:

$$l_p \leq y_p \leq u_p \quad \forall p \in P; \quad (5.1)$$

$$\sum_{p \in P_w} y_p = d_w \quad \forall w \in W. \quad (5.2)$$

The set of all feasible path flows is denoted K . Assume further that a vector cost function \hat{c}_a is given on each arc $a \in A$, depending on the traffic arc flow z and taking values in a finite dimensional space \mathbb{R}^l with $l \geq 2$. Then the vector cost function c_p on path p depends on the path flow y and is computed by

$$c_p(y) = \sum_{a \in A} \hat{c}_a(z) \delta_{ap}, \quad (5.3)$$

as we already knew in the model without capacity constraints in Chapter 4. We recall that $C(y)$ denotes the $l \times m$ -matrix, the columns of which are $c_p, p \in P$ and $\hat{C}(z)$ denotes the $l \times n$ -matrix, the columns of which are $\hat{c}_a, a \in A$. Then these matrices are linked by the formula

$$C(y) = \hat{C}(z)\Delta,$$

with Δ the incident matrix. In this chapter we study the concept of equilibrium introduced in [24] which is a vector version of Wardrop's famous user principle.

Definition 5.1.1 *A feasible path flow y is said to be a vector equilibrium (respectively a weak vector equilibrium) of G if for every O/D pair $w \in W$ and for every couple of paths $p, p' \in P_w$ one has implication*

$$c_p(y) - c_{p'}(y) \geq 0 \text{ (respectively } c_p(y) - c_{p'}(y) > 0) \implies \text{either } y_p = l_p \text{ or } y_{p'} = u_{p'}.$$

It is clear that every vector equilibrium is weak vector equilibrium, and the converse is not true in general. When $l_p = 0$ and $u_p = +\infty$, that is, there are no capacity constraints, the aforementioned definition coincides with the one given in Definition 4.0.25.

5.2 Equivalent optimization problem

In this section we construct an optimization problem the optimal solutions of which are equilibria of the traffic network with capacity constraints we described above. For every feasible flow y define

$$\psi(y) := \sum_{p, p' \in P_w, w \in W} (y_p - l_p)(u_{p'} - y_{p'})[c_p(y) - c_{p'}(y)]^T H_+[c_p(y) - c_{p'}(y)].$$

where the function H_+ is a vector version of the Heaviside Step function, which was given in Chapter 4. Since in the case of network without capacity constraints the upper capacity constraint is set equal to $+\infty$, the function ψ introduced in Chapter 4 cannot be obtained from this function. However, most of properties of that function we established in Chapter 4 remain true. The following theorem is important for the further developments.

Theorem 5.2.1 *Let \bar{y} be a feasible flow. The following statements are equivalent:*

- (i) \bar{y} is a vector equilibrium.
- (ii) \bar{y} is an optimal solution of the following problem, denoted (P):

$$\begin{aligned} & \text{minimize } \psi(y) \\ & \text{subject to } y \in K \end{aligned}$$

and the optimal value of this problem is zero.

Proof. Let \bar{y} be a vector equilibrium. Since $\psi(y) \geq 0$ for every $y \in K$, as before, it suffices to prove $\psi(\bar{y}) = 0$ in order to deduce (ii). Let $p \in P_w, w \in W$. Consider the term $(\bar{y}_p - l_p)(u_{p'} - \bar{y}_{p'})[c_p(\bar{y}) - c_{p'}(\bar{y})]^T H_+[c_p(\bar{y}) - c_{p'}(\bar{y})]$, denoted $b_{p,p'}$. If $c_p(\bar{y}) - c_{p'}(\bar{y}) \geq 0$ for some $p' \in P_w$, then by definition, either $\bar{y}_p = l_p$ or $\bar{y}_{p'} = u_{p'}$. If $c_p(\bar{y}) - c_{p'}(\bar{y}) = 0$ for some $p' \in P_w$, it is clear that the corresponding term of the above sum is zero. If $c_p(\bar{y}) - c_{p'}(\bar{y}) \not\geq 0$ for some $p' \in P_w$, then $H_+[c_p(\bar{y}) - c_{p'}(\bar{y})] = 0$. Therefore $b_{p,p'} = 0$. Consequently, $\psi(\bar{y}) = 0$ as requested. Conversely, assume \bar{y} solves (P) and $\psi(\bar{y}) = 0$. It follows that $b_{p,p'} = 0$ for every $p \in P$. If for some p and p' from $P_w, w \in W$ one has $c_p(\bar{y}) - c_{p'}(\bar{y}) \geq 0$, then $[c_p(\bar{y}) - c_{p'}(\bar{y})]^T H_+[c_p(\bar{y}) - c_{p'}(\bar{y})] > 0$. Consequently, either $\bar{y}_p = l_p$ or $\bar{y}_{p'} = u_{p'}$. We

deduce that \bar{y} is a vector equilibrium. \square

By using the same method of proof we may establish a similar result for weak vector equilibria. For this purpose let us define

$$\psi^w(y) := \sum_{p,p' \in P_w, w \in W} (y_p - l_p)(u_{p'} - y_{p'})[c_p(y) - c_{p'}(y)]^T H_+^w[c_p(y) - c_{p'}(y)]$$

for every $y \in K$, and consider the optimization problem (P_w) :

$$\begin{aligned} & \text{minimize } \psi^w(y) \\ & \text{subject to } y \in K, \end{aligned}$$

where H_+^w is defined by

$$H_+^w(x) = \begin{cases} e & \text{if } x > 0 \\ 0 & \text{else.} \end{cases}$$

Theorem 5.2.2 *Let \bar{y} be a feasible flow. Then \bar{y} is a weak vector equilibrium if and only if it solves the problem (P_w) with the optimal value equal to zero.*

Proof. Apply the method of proof of Theorem 5.2.1. \square

Finally we observe that the conclusion of Theorem 5.2.1 remains true when the function H_+ defining the objective function of (P) is substituted by the function

$$\tilde{H}_+(x) = \left(\prod_{i=1}^l \tilde{h}_+(x_i) \right) e \text{ for all } x \in \mathbb{R}^l,$$

where $\tilde{h}_+(t) = h_+(t)$ for all $t \in \mathbb{R} \setminus \{0\}$ and $\tilde{h}_+(0) = 1/2$.

5.3 Generic differentiability and local calmness of the objective function

We apply the methods of proof of Chapter 4 to establish some properties of the function ψ .

Theorem 5.3.1 *Assume that the vector cost functions $c_{p_i}, i = 1, \dots, m$ are continuous (respectively locally Lipschitz or differentiable). Then every open set in \mathbb{R}^m contains an open subset where the objective function ψ of problem (P) is continuous (respectively locally Lipschitz or differentiable).*

Proof. The general term of this ψ is obtained from the general term of the function ψ of Chapter 4 by substituting y_p by $(y_p - l_p)(u_p - y_p)$ which is continuous (or locally Lipschitz/differentiable). Therefore the proof of Theorem 4.2.4 applies. \square

Theorem 5.3.2 *Assume that the vector cost functions $c_{p_i}, i = 1, \dots, m$ are differentiable. Then for every point y outside of some nowhere dense subset and for every path p_i , there exists a subset $J_i(y) \subseteq I_{w(i)}$ such that*

- (i) $c_{p_i}(y) \geq c_{p_j}(y)$ for every $j \in J_i(y)$
- (ii) $\psi(y) = \sum_{i=1}^m (y_{p_i} - l_{p_i}) \left\langle \sum_{j \in J_i(y)} (u_{p_j} - y_{p_j}) (c_{p_i}(y) - c_{p_j}(y)), e \right\rangle$

(iii) ψ is differentiable at y and its gradient's components are computed by

$$\begin{aligned} \frac{\partial \psi(y)}{\partial y_k} &= \sum_{i=1}^m (y_{p_i} - l_{p_i}) \sum_{j \in J_i(y)} (u_{p_j} - y_{p_j}) \left\langle \frac{\partial c_{p_i}(y)}{\partial y_k} - \frac{\partial c_{p_j}(y)}{\partial y_k}, e \right\rangle \\ &+ \sum_{\substack{i=1 \\ i \neq k}}^m (l_{p_i} - y_{p_i}) \langle c_{p_i}(y) - c_{p_k}(y), e \rangle \\ &+ \sum_{j \in J_k(y)} (u_{p_j} - y_{p_j}) \langle c_{p_k}(y) - c_{p_j}(y), e \rangle \end{aligned}$$

for $k = 1, \dots, m$.

Proof. The same proof of Theorem 4.2.6 goes through for (i) and (ii). The formula given in (iii) is obtained by a direct calculation. \square

We notice that in general the objective function of the problem (P) is not continuous. Again we are able to prove its calmness at a point that is a vector equilibrium.

Proposition 5.3.3 *Assume that the vector cost functions c_{p_1}, \dots, c_{p_m} are continuous. Then the function ψ is continuous at every vector equilibrium. If in addition c_{p_1}, \dots, c_{p_m} are locally calm at a vector equilibrium, then ψ is also locally calm there.*

Proof. Let $\bar{y} \in K$ be a vector equilibrium of G . Assume that c_{p_1}, \dots, c_{p_m} are continuous on K and locally calm at \bar{y} . The case where the function ψ is merely continuous is proven by a similar argument. We wish to show that there exist some $\delta > 0$ and $\kappa > 0$ such that $|\psi(y) - \psi(\bar{y})| \leq \kappa \|y - \bar{y}\|$ for every $y \in K$ with $\|y - \bar{y}\| \leq \delta$. Because ψ takes nonnegative values and as \bar{y} is equilibrium, by Theorem 5.2.1, $\psi(\bar{y}) = 0$, the above inequality is equivalent to

$$\psi(y) \leq \kappa \|y - \bar{y}\|. \quad (5.4)$$

Recall that for $i \in \{1, \dots, m\}$, $w(i)$ denotes the O/D pair connected by the path p_i and $P_{w(i)}$ denotes the set of all paths connecting this O/D pair. We make also use of the following notations

$$\begin{aligned} I_0 &= \{i \in \{1, \dots, m\} : \bar{y}_{p_i} = l_{p_i}\} \\ I_+ &= \{i \in \{1, \dots, m\} : l_{p_i} < \bar{y}_{p_i} \leq u_{p_i}\} \\ D &= \max\{d_w : w \in W\} \\ T &= \max\{\|c_{p_i}(y)\| : y \in K, i = 1, \dots, m\} \\ U &= \max\{\|u_{p_i}(y)\| : y \in K, i = 1, \dots, m\}. \end{aligned}$$

The following inequality is clear

$$y_{p_i} \leq \|y\| \leq D \quad (5.5)$$

for every $y \in K$ and $p_i \in P$. Consider the terms

$$g_i(y) := (y_{p_i} - l_{p_i}) \sum_{p_j \in P_{w(i)}} (u_{p_j} - y_{p_j}) [c_{p_i}(y) - c_{p_j}(y)]^T H_+ [c_{p_i}(y) - c_{p_j}(y)]$$

for $i = 1, \dots, m$.

Observe first that

$$0 \leq (u_{p_j} - y_{p_j}) [c_{p_i}(y) - c_{p_j}(y)]^T H_+ [c_{p_i}(y) - c_{p_j}(y)] \leq 2lTU,$$

for all $p_i, p_j \in P$ and $y \in K$. Therefore, if $i \in I_0$, then

$$|g_i(y) - g_i(\bar{y})| = g_i(y) \leq 2mlTU(y_{p_i} - l_{p_i}) \leq 2mlTU \|y - \bar{y}\|. \quad (5.6)$$

If $i \in I_+$, we have

$$\sum_{p_j \in P_{w(i)}} (u_{p_j} - \bar{y}_{p_j}) [c_{p_i}(\bar{y}) - c_{p_j}(\bar{y})]^T H_+ [c_{p_i}(\bar{y}) - c_{p_j}(\bar{y})] = 0,$$

which implies that for $p_j \in P_{w(i)} \setminus \{p_i\}$, either $\bar{y}_{p_j} = u_{p_j}$ or $c_{p_i}(\bar{y}) \not\leq c_{p_j}(\bar{y})$ or $c_{p_i}(\bar{y}) = c_{p_j}(\bar{y})$. In the first case, there exists some $p_i \in P_{w(i)}$ such that $c_{p_i}(\bar{y}) \geq c_{p_j}(\bar{y})$. Due to the continuity hypothesis, there exists some $\delta'_i > 0$ such that $c_{p_i}(y) \geq c_{p_j}(y)$ for all $y \in K$ with $\|y - \bar{y}\| \leq \delta'_i$. In the second case, also due to the continuity hypothesis, there exists some $\delta''_i > 0$ such that $c_{p_i}(y) \not\leq c_{p_j}(y)$ for all $y \in K$ with $\|y - \bar{y}\| \leq \delta''_i$. In the third case, by the calmness hypothesis, there are some constants $\delta' > 0$ and $\kappa' > 0$ such that

$$\|c_{p_j}(y) - c_{p_j}(\bar{y})\| \leq \kappa' \|y - \bar{y}\| \quad (5.7)$$

for every $y \in K$ with $\|y - \bar{y}\| \leq \delta'$ and $j = 1, \dots, m$. We have

$$\begin{aligned} \|c_{p_i}(y) - c_{p_j}(y)\| &= \|c_{p_i}(y) - c_{p_i}(\bar{y}) + c_{p_i}(\bar{y}) - c_{p_j}(\bar{y}) + c_{p_j}(\bar{y}) - c_{p_j}(y)\| \\ &\leq 2\kappa' \|y - \bar{y}\| \end{aligned}$$

for all $y \in K$ with $\|y - \bar{y}\| \leq \delta'$. Set $\gamma_i = \min\{\delta', \delta'_i, \delta''_i\}$. We deduce

$$|g_i(y) - g_i(\bar{y})| = g_i(y) \leq lDU \sum_{p_j \in P_{w_i}} \|c_{p_i}(y) - c_{p_j}(y)\| \leq 2lmDU\kappa' \|y - \bar{y}\| \quad (5.8)$$

for $y \in K$ with $\|y - \bar{y}\| \leq \gamma_i$. It remains to choose $\delta = \min\{\gamma_i : i \in I_+\}$ and $\kappa = 2mlTU|I_0| + 2mlDU\kappa'|I_+|$ to obtain (5.4) from (5.6) and (5.8). The proof is complete. \square

5.4 Generating vector equilibrium flows

In this section we present an algorithm based on the one in Chapter 4 with some modifications to obtain a subset of vector equilibria of the network. We also present some numerical examples to illustrate the algorithm.

5.4.1 Description of the algorithm

Assume that W consists of r elements w_1, \dots, w_r in the network and for each pair w_i there are $|P_{w_i}|$ paths connecting its origin to its destination. We denote also $I_j = \{i \in \{1, \dots, m\} : p_i \in P_{w_j}\}$.

Step 0 (initialization). Choose a positive integer q and a tolerance level $\epsilon \geq 0$.

Procedure A1.

Step 1. Enter $L = (l_p)_{p \in P}$ and $U = (u_p)_{p \in P}$. Set $\delta_j = d_{w_j} / (q|P_{w_j}|)$, $j = 1, \dots, r$.

Step 2. Choose $(k_1, \dots, k_m)^T \in \mathbb{N}^m$ satisfying

$$\sum_{i \in I_j} k_i = q|P_{w_j}|, \text{ and } l_{p_i} \leq k_i \cdot \delta_j \leq u_{p_i} \text{ for } i \in I_j; j = 1, \dots, r.$$

Step 3. Store $y = (y_1, \dots, y_m)^T$ in Y^0 where

$$y_i = k_i \delta_j \text{ for } i \in I_j, j = 1, \dots, r$$

and return to Step 2 for other vectors (k_1, \dots, k_m) unless no one left.

Procedure A2.

Step 4. Choose a feasible flow y^0 from Y^0 to start. Set $k = 0$, $u^{k-1} = y^k$, $\alpha_{k-1} = \infty$, $Y^0 = Y^0 \setminus \{y^0\}$ and $E_q^\epsilon = \emptyset$.

Step 5. Compute $J_i(y^k) = \{i' \in \{1, \dots, m\} : p_{i'} \in P_{w(i)}, c_{p_i}(y^k) - c_{p_{i'}}(y^k) \geq 0\}$ for every $i \in \{1, \dots, m\}$. Set

$$\psi_k(y) := \sum_{i=1}^m (y_{p_i} - l_{p_i}) \sum_{i' \in J_i(y^k)} (u_{p_{i'}} - y_{p_{i'}}) \langle c_{p_i}(y) - c_{p_{i'}}(y), e \rangle.$$

Compute $\psi_k(y^k)$.

If $\psi_k(y^k) \leq \epsilon$, store y^k in E_q^ϵ and return to Step 4 until no element of Y^0 left.

Otherwise go to the next step.

Step 6. If $|\psi_k(y^k) - \alpha_{k-1}| \leq \epsilon$, go to Step 4 to choose another feasible solution from Y^0 to restart the procedure.

If $\psi_k(y^k) < \alpha_{k-1} - \epsilon$, set $\alpha_k = \psi_k(y^k)$ and go to Step 7.

If $\psi_k(y^k) > \alpha_{k-1} + \epsilon$, replace $y^k = y^{k-1} + (y^k - y^{k-1})/2$ and return to Step 5.

Step 7. Compute $\nabla \psi_k(y^k)$. Solve (P_k)

$$\begin{aligned} & \text{minimize} && u^T \nabla \psi_k(y^k) \\ & \text{subject to} && u \in K \\ & && |u_i - y_i^0| \leq \delta_{w(i)}, i = 1, \dots, m. \end{aligned}$$

Let u^k be an optimal solution.

If $|\psi_k(y^k) - \psi_k(u^k)| \leq \epsilon$, go to Step 4 to choose another feasible solution from Y^0 to restart the procedure until no element of Y^0 left.

Otherwise, set $y^{k+1} = u^k$. Update $k = k + 1$ and return to Step 5.

5.4.2 Numerical examples

The examples below are modified versions of Examples 4.3.4-4.3.6 of Chapter 4 to which we add lower and upper capacity constraints. They are coded and computed on Matlab Version 2014a.

Example 5.4.1 Consider a network problem with one pair of origin-destination nodes $w = (x, x')$, two criteria: travel time and travel cost, two available paths: $P_w = \{p_1, p_2\}$ with the travel demand $d_w = 30$. Assume that

$$\begin{aligned} l_{p_1} &= 1 & l_{p_2} &= 1 & u_{p_1} &= 20 & u_{p_2} &= 25 \\ c_{p_1,1}(y) &= y_{p_1} + 2y_{p_2} & c_{p_2,1}(y) &= y_{p_1} + 6y_{p_2} \\ c_{p_1,2}(y) &= 6y_{p_1} + 2y_{p_2} & c_{p_2,2}(y) &= 6y_{p_1} + 8y_{p_2}. \end{aligned}$$

We tested our program for the zero tolerance $\epsilon = 0$ and $q = 3$ which yields 4 feasible initial points. The results are displayed in the table below.

Initial point	Numbers of iterations	Vector equilibrium	CPU Time (seconds)
$(5, 25)^T$	100	—	0.285410
$(10, 20)^T$	100	—	
$(15, 15)^T$	3	$(20, 10)^T$	
$(20, 10)^T$	1	$(20, 10)^T$	

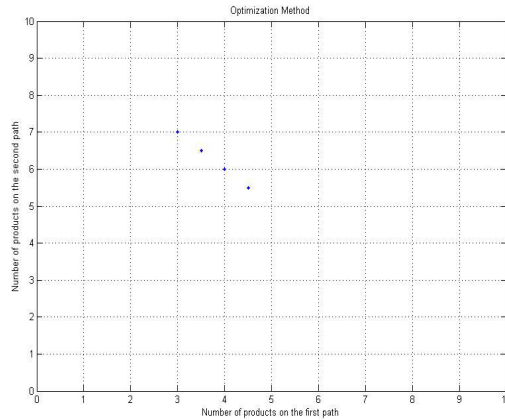
We remark that starting from the first two initial points our algorithm finds no vector equilibrium because after some iterations either the value $\psi_k(y^k)$ is almost the same as α_{k-1} at Step 6 or the value $\psi_k(y^k)$ is almost the same as $\psi_k(u^k)$ at Step 7, which enforces us to go to Step 4 to choose another initial point to restart. Moreover, when the tolerance is strictly positive, the program runs much faster. For instance with $\epsilon = 10^{-4}$, the algorithm finds two vector ϵ -equilibria in 0.053571 seconds, one of which is $(20, 10)^T$ and the other is very near to it.

Example 5.4.2 Consider the network described in Example 5.4.1 with one pair of origin-destination nodes, two paths and two criteria with the travel demand $d_w = 1$. Assume that

$$l_{p_1} = 1 \quad l_{p_2} = 2 \quad u_{p_1} = 8 \quad u_{p_2} = 7$$

$$\begin{aligned} c_{p_1,1}(y) &= 3y_{p_1} + y_{p_2} & c_{p_2,1}(y) &= y_{p_1} + y_{p_2} \\ c_{p_1,2}(y) &= 5y_{p_1} + 3y_{p_2} & c_{p_2,2}(y) &= 3y_{p_1} + 5y_{p_2}. \end{aligned}$$

With $q = 10$ we have 11 feasible initial points, we obtain 5 vector equilibria in 0.814601 seconds. It can be seen that the set of all vector equilibria is the segment between $(3, 7)^T$ and $(5, 5)$. The vector equilibrium set is graphically presented below.



In the next example the travel time and travel cost functions are nonlinear.

Example 5.4.3 Consider a network problem with only one pair of origin-destination nodes $w = (x, x')$, two criteria: travel time and travel cost, three available paths: $P_w = \{p_1, p_2, p_3\}$ with the travel demand $d_w = 18$. Assume that

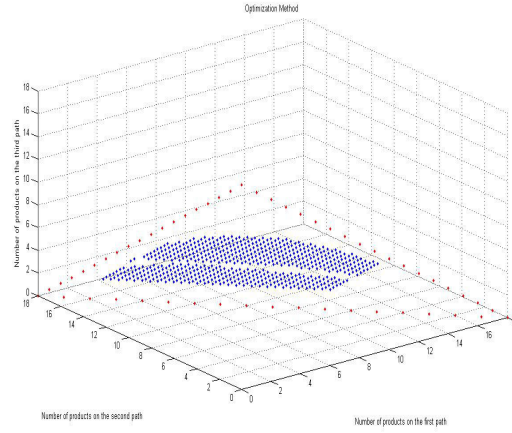
$$l_{p_1} = 1 \quad l_{p_2} = 0 \quad l_{p_3} = 3 \quad u_{p_1} = 10 \quad u_{p_2} = 15 \quad u_{p_3} = 12$$

$$\begin{aligned} c_{p_1,1}(y) &= y_{p_1}^2 + y_{p_2}^2 + y_{p_3}^3 & c_{p_2,1}(y) &= 8y_{p_1}y_{p_2} + y_{p_2}^2 \\ c_{p_1,2}(y) &= 2y_{p_1} + 5y_{p_2} + 3y_{p_3} & c_{p_2,2}(y) &= y_{p_2} + 10y_{p_3} \end{aligned}$$

$$c_{p_3,1}(y) = y_{p_1} + y_{p_2}^2 + y_{p_3}^3$$

$$c_{p_3,2}(y) = 10y_{p_3}^3.$$

With $\epsilon = 0$ and $q = 20$ we have 839 feasible initial points. Using the algorithm we obtained 659 vector equilibria in 18.809593 seconds which are presented in the next figure.



Equilibrium in a multi-product multi-criteria traffic network with capacity constraints

The purpose of this chapter is to study a multi-product, multi-criteria network with capacity constraints in which all products and all criteria are simultaneously considered. We establish existence conditions for strong vector equilibrium and a relationship between strong vector equilibrium and Pareto efficient elements of the value set of the vector cost function. The main attention is paid to constructing equivalent variational inequality problems with the help of particular classes of increasing functions. An algorithm is proposed to solve multi-criteria network equilibrium problems and numerical examples are presented to illustrate our approach.

Let us now recall some principal concepts of equilibrium in a multi-product multi-criteria network.

Definition 6.0.4 [37] *A feasible flow Y is said to be a strong vector equilibrium of the network G if for every $w \in W$ and $p_\alpha, p_\beta \in P_w$ one has implication*

$$C_{p_\alpha}(Y) \geq C_{p_\beta}(Y) \Rightarrow \text{either } Y_{p_\alpha} = L_{p_\alpha} \text{ or } Y_{p_\beta} = U_{p_\beta}.$$

It is said to be a weak vector equilibrium if for every $w \in W$ and $p_\alpha, p_\beta \in P_w$ one has implication

$$C_{p_\alpha}(Y) \geq C_{p_\beta}(Y) \Rightarrow \text{either } Y_{p_\alpha} \not\leq L_{p_\alpha} \text{ or } Y_{p_\beta} \not\leq U_{p_\beta}.$$

The aforementioned implications were already given in Chapter 3 in which we gave a number of comments on their relations with other concepts of equilibrium.

6.1 Existence conditions

It is known that in a single-criterion traffic model, Wardrop's equilibrium exists under rather mild conditions (the continuity of the cost function and the compactness of the feasible set for instance). This, however, is not true in the case of multi-criteria networks. A simple example below gives a network without strong vector equilibrium flows in which the cost function is linear.

Example 6.1.1 *Consider a network problem with only one pair of origin-destination nodes $w = (x, x')$, two criteria and two products to traverse the network with two available paths: $P_w = \{p_1, p_2\}$. Assume that $d_w^1 = y_{p_1}^1 + y_{p_2}^1 = 10$, $d_w^2 = y_{p_1}^2 + y_{p_2}^2 = 10$ and*

$$l_{p_1}^1 = l_{p_1}^2 = 1 \quad l_{p_2}^1 = 3.5 \quad l_{p_2}^2 = 2 \quad u_{p_1}^1 = u_{p_1}^2 = 10 \quad u_{p_2}^1 = u_{p_2}^2 = 12$$

$$\begin{aligned} c_{p_1}^1(Y) &= (y_{p_1}^1, y_{p_1}^1)^T & c_{p_1}^2(Y) &= (y_{p_1}^2, y_{p_1}^2)^T \\ c_{p_2}^1(Y) &= (3y_{p_2}^1, 2y_{p_2}^1)^T & c_{p_2}^2(Y) &= (3y_{p_2}^2, 2y_{p_2}^2)^T \end{aligned}$$

There does not exist any feasible flow Y satisfying the following condition: $C_{p_\alpha}(Y) \geq C_{p_\beta}(Y)$ implies either $Y_{p_\alpha} = L_{p_\alpha}$ or $Y_{p_\beta} = U_{p_\beta}$ for $p_\alpha, p_\beta \in P_w$. That means this problem does not have any strong vector equilibrium.

To establish existence conditions for strong vector equilibrium flows we need the concept of elementary flows [38], which we suggest to call elementary flow variations in reference to the well known concept of elementary variations in the theory of optimal control. A path flow Y of the network G is said to be an elementary flow variation if there are some origin-destination pair $w \in W$ and paths $p_\alpha, p_\beta \in P_w$ such that

$$\begin{aligned} Y_{p_\alpha} &= -Y_{p_\beta} \\ Y_{p_i} &= 0 \text{ for } p_i \in P_w \setminus \{p_\alpha, p_\beta\}. \end{aligned}$$

From now on, let us fix a path flow \bar{Y} , write \bar{C} and $\bar{c}_{p_i}^j$ instead of $C(\bar{Y})$ and $c_{p_i}^j(\bar{Y})$ if no misunderstanding occurs. We use also upper index r to indicate the r th component of each vector $c_{p_i}^j$ and set

$$K_+(\bar{Y}) = \{Y \in K : Y - \bar{Y} \text{ is elementary with } (Y - \bar{Y})_{p_\alpha} \geq 0 \text{ for some } p_\alpha\} \cup \{\bar{Y}\}.$$

We shall need also the following result (Theorem 5.4 [37]): If a feasible flow \bar{Y} is a strong vector equilibrium flow, then it satisfies the following vector variational inequality

$$\left(\begin{array}{ccc} \sum_{i=1}^m \bar{c}_{p_i,1}^1(Y_{p_i} - \bar{Y}_{p_i}) & \cdots & \sum_{i=1}^m \bar{c}_{p_i,1}^q(Y_{p_i} - \bar{Y}_{p_i}) \\ \vdots & \ddots & \vdots \\ \sum_{i=1}^m \bar{c}_{p_i,l}^1(Y_{p_i} - \bar{Y}_{p_i}) & \cdots & \sum_{i=1}^m \bar{c}_{p_i,l}^q(Y_{p_i} - \bar{Y}_{p_i}) \end{array} \right) \notin (-\mathbb{R}_+^q)^{l \times q} \setminus \{0\} \quad (6.1)$$

for all $Y \in K_+(\bar{Y})$. Conversely, if \bar{Y} is a solution of the above variational inequality and satisfies the following condition: For every origin-destination pair $w \in W$ and for every couple of paths $p_\alpha, p_\beta \in P_w$ one has implication

$$L_{p_\alpha} \leq Y_{p_\alpha} \text{ and } Y_{p_\beta} \leq U_{p_\beta} \Rightarrow l_{p_\alpha}^k < y_{p_\alpha}^k \text{ and } y_{p_\beta}^k < u_{p_\beta}^k \text{ for some } k \in \{1, \dots, q\}, \quad (6.2)$$

then \bar{Y} is a strong vector equilibrium flow. We notice that condition (6.2) holds trivially when the network is without capacity constraints, or more generally when the upper bound on each path is sufficiently large with regard to the demand. For instance, it is the case when $d_w^j < u_{p_i}^j$ for every $p_i \in P_w, j \in \{1, \dots, q\}$.

Theorem 6.1.2 *Assume that the feasible set K is nonempty. The network G admits a weak vector equilibrium flow if the cost function $C(\cdot)$ is continuous with respect to the variables $y_{p_1}^j, \dots, y_{p_m}^j$ for some $j \in \{1, \dots, q\}$. Moreover, if $C(\cdot)$ is continuous, then the vector variational inequality (6.1) has a solution, which is also a strong vector equilibrium when it satisfies condition (6.2).*

Proof. Without loss of generality we may assume $j = 1$. We fix a feasible flow \bar{y} and consider the network G with a single product $j = 1$ and a single-criterion cost function on path i : $f_i(z) = \sum_{r=1}^l \sum_{s=1}^q c_{p_i,r}^s(z, \bar{y}^2, \dots, \bar{y}^q)$. The feasible set K_1 for this model consists of flows $z = (z_{p_1}^1, \dots, z_{p_m}^1)^T$ satisfying the constraints: $l_{p_i}^1 \leq z_{p_i}^1 \leq u_{p_i}^1$ for $p_i \in P = \{p_1, \dots, p_m\}$ and $\sum_{p_i \in P_w} z_{p_i}^1 = d_w^1$ for every $w \in W$. By hypothesis the cost function $f = (f_1, \dots, f_m)$

is continuous, and therefore the network admits an equilibrium \bar{z} in the classical sense: $f_\alpha(\bar{z}) > f_\beta(\bar{z}) \Rightarrow$ either $\bar{z}_{p_\alpha}^1 = l_{p_\alpha}^1$ or $\bar{z}_{p_\beta}^1 = u_{p_\beta}^1$. We claim that the flow $\bar{Z} = (\bar{z}, \bar{y}^2, \dots, \bar{y}^q)$ is a weak vector equilibrium flow of the original network. Indeed, it is clear that \bar{Z} belongs to K . If for some $w \in W$, $p_\alpha, p_\beta \in P_w$ one has $C_{p_\alpha}(\bar{Z}) \geq C_{p_\beta}(\bar{Z})$, then all components of $C_{p_\alpha}(\bar{Z})$ are bigger than or equal to the corresponding components of $C_{p_\beta}(\bar{Z})$, and at least one inequality is strict, which yields $f_\alpha(\bar{z}) = \sum_{r=1}^l \sum_{s=1}^q c_{p_\alpha, r}^s(\bar{z}, \bar{y}^2, \dots, \bar{y}^q) > \sum_{r=1}^l \sum_{s=1}^q c_{p_\beta, r}^s(\bar{z}, \bar{y}^2, \dots, \bar{y}^q) = f_\beta(\bar{z})$. It follows that either $\bar{z}_{p_\alpha}^1 = l_{p_\alpha}^1$ or $\bar{z}_{p_\beta}^1 = u_{p_\beta}^1$, which implies either $\bar{Z}_{p_\alpha} \not\prec L_{p_\alpha}$ or $\bar{Z} \not\prec U_{p_\beta}$. This shows that \bar{Z} is a weak vector equilibrium.

To prove the second part of the theorem we consider the following variational inequality problem: Find $\bar{z} \in \mathbb{R}^{mq}$ such that $\langle \phi(\bar{z}), z - \bar{z} \rangle \geq 0$ for every $z \in Q$, where $z \in Q$ if and only if $z = (y_{p_1}^1, \dots, y_{p_1}^q, \dots, y_{p_m}^1, \dots, y_{p_m}^q)$ for some $Y \in K$, and $\phi(z) = (\phi_1(z), \dots, \phi_1(z), \dots, \phi_m(z), \dots, \phi_m(z)) \in \mathbb{R}^{mq}$ for $z \in Q$, in which

$\phi_i(z) = \sum_{j=1}^q \sum_{r=1}^l c_{p_i, r}^j(Y)$ with Y generating z as above. Since K is a compact set and ϕ is continuous by hypothesis, this variational inequality admits a solution. Let \bar{y} be a feasible flow corresponding to this solution. Then one can check that it is a solution to a vector variational inequality (6.1). Due to the condition (6.2) we may apply Theorem 5.4 of [37] to conclude that \bar{y} is a strong vector equilibrium flow of the original network. \square

We notice also that if the condition (6.2) is not satisfied, then a solution of (6.1) is not necessarily a strong vector equilibrium. This is seen by the following example.

Example 6.1.3 Consider a network problem with only one pair of origin-destination nodes $w = (x, x')$, two criteria and two products to traverse in the network with two available paths: $P_w = \{p_1, p_2\}$. Assume that $d_{w_1} = 4$, $d_w^2 = 20$, $l_{p_i}^j = 2$, $u_{p_i}^j = 10$ for $p_i, p_j \in P_w$ and

$$\begin{array}{llll} \bar{y}_{p_1}^1 = 2 & \bar{y}_{p_1}^2 = 10 & \bar{c}_{p_1}^1 = (15, 35)^T & \bar{c}_{p_1}^2 = (25, 20)^T \\ \bar{y}_{p_2}^1 = 2 & \bar{y}_{p_2}^2 = 10 & \bar{c}_{p_2}^1 = (10, 20)^T & \bar{c}_{p_2}^2 = (15, 12)^T \end{array}$$

Then, \bar{Y} does not satisfy the condition (8). In fact $\bar{Y}_{p_1} \geq L_{p_1}$, $\bar{Y}_{p_2} \leq U_{p_2}$, and there is not any $j \in \{1, 2\}$ such that $\bar{y}_{p_1}^j > l_{p_1}^j$ and $\bar{y}_{p_2}^j < u_{p_2}^j$. And for all $Y \in K_+(\bar{Y})$ one has

$$\begin{array}{ll} \left(\begin{array}{cc} \bar{c}_{p_1,1}^1(Y_{p_1} - \bar{Y}_{p_1}) + \bar{c}_{p_2,1}^1(Y_{p_2} - \bar{Y}_{p_2}) & \bar{c}_{p_1,1}^2(Y_{p_1} - \bar{Y}_{p_1}) + \bar{c}_{p_2,1}^2(Y_{p_2} - \bar{Y}_{p_2}) \\ \bar{c}_{p_1,2}^1(Y_{p_1} - \bar{Y}_{p_1}) + \bar{c}_{p_2,2}^1(Y_{p_2} - \bar{Y}_{p_2}) & \bar{c}_{p_1,2}^2(Y_{p_1} - \bar{Y}_{p_1}) + \bar{c}_{p_2,2}^2(Y_{p_2} - \bar{Y}_{p_2}) \end{array} \right) \\ = \left(\begin{array}{cc} 5(y_{p_1}^1 - 2) & 10(y_{p_1}^1 - 2) \\ 5(y_{p_1}^2 - 10) & 10(y_{p_1}^2 - 10) \\ 15(y_{p_1}^1 - 2) & 8(y_{p_1}^1 - 2) \\ 15(y_{p_1}^2 - 10) & 8(y_{p_1}^2 - 10) \end{array} \right) \notin (-\mathbb{R}_+^2)^{2 \times 2} \setminus \{0\} \end{array}$$

which shows that \bar{Y} is a solution of the problem (6.1).

However, \bar{Y} is not a strong vector equilibrium as $\bar{C}_{p_1} = (\bar{c}_{p_1}^1 \ \bar{c}_{p_1}^2) \geq \bar{C}_{p_2} = (\bar{c}_{p_2}^1 \ \bar{c}_{p_2}^2)$ but $\bar{Y}_{p_1} \neq L_{p_1}$ and $\bar{Y}_{p_2} \neq U_{p_2}$.

6.2 Equivalent problems

6.2.1 Equilibrium with respect to a family of increasing functions

Let \mathcal{F} be a family of real functions on $\mathbb{R}^{l \times q}$. We say that a feasible flow \bar{Y} is \mathcal{F} -equilibrium if for every $w \in W$ and $p_\alpha, p_\beta \in P_w$, one has

$$f(\bar{C}_{p_\alpha}) > f(\bar{C}_{p_\beta}), \quad \forall f \in \mathcal{F} \Rightarrow \text{either } \bar{Y}_{p_\alpha} = L_{p_\alpha} \text{ or } \bar{Y}_{p_\beta} = U_{p_\beta}. \quad (6.3)$$

The next result is a version of Theorem 4.2 of [37] for strong vector equilibrium.

Lemma 6.2.1 *Let a feasible flow \bar{Y} be given. If the family \mathcal{F} consists of increasing functions on $\{\bar{C}_{p_i}, p_i \in P\}$, then every \mathcal{F} -equilibrium is a strong vector equilibrium. Conversely, if \mathcal{F} satisfies the following condition: For every $w \in W$ and $p_i, p_{i'} \in P_w$ one has implication*

$$f(\bar{C}_{p_i}) > f(\bar{C}_{p_{i'}}) \quad \forall f \in \mathcal{F} \Rightarrow \bar{C}_{p_i} \geq \bar{C}_{p_{i'}}, \quad (6.4)$$

then every strong vector equilibrium is an \mathcal{F} -equilibrium.

Proof. Assume that the feasible flow \bar{Y} is an \mathcal{F} -equilibrium. If for some $w \in W$ and $p_\alpha, p_\beta \in P_w$, we have $\bar{C}_{p_\alpha} \geq \bar{C}_{p_\beta}$, then for every increasing function $f \in \mathcal{F}$, we obtain $f(\bar{C}_{p_\alpha}) > f(\bar{C}_{p_\beta})$. By definition, we have either $\bar{Y}_{p_\alpha} = L_{p_\alpha}$ or $\bar{Y}_{p_\beta} = U_{p_\beta}$. Hence, \bar{Y} is a strong vector equilibrium.

Conversely, let \bar{Y} be a strong vector equilibrium. If for some $w \in W$ and $p_\alpha, p_\beta \in P_w$, we have $f(\bar{C}_{p_\alpha}) > f(\bar{C}_{p_\beta})$ for all $f \in \mathcal{F}$, then by the assumption, it implies that $\bar{C}_{p_\alpha} \geq \bar{C}_{p_\beta}$. We deduce that either $\bar{Y}_{p_\alpha} = L_{p_\alpha}$ or $\bar{Y}_{p_\beta} = U_{p_\beta}$ which shows that \bar{Y} is an \mathcal{F} -equilibrium. \square

We notice that if the family \mathcal{F} does not satisfy (6.4), then a strong vector equilibrium is not necessarily an \mathcal{F} -equilibrium as it is shown by the example below.

Example 6.2.2 *Consider a network problem with only one pair of origin-destination nodes $w = (x, x')$, two criteria and two products to traverse the network with three available paths: $P_w = \{p_1, p_2, p_3\}$. Assume that $d_w^1 = 10, d_w^2 = 18, l_{p_i}^j = 2, u_{p_i}^j = 10$ for $p_i \in P_w$ and $j = 1, 2$, and*

$$\begin{array}{cccc} \bar{y}_{p_1}^1 = 3 & \bar{y}_{p_1}^2 = 7 & \bar{c}_{p_1}^1 = (2, 2)^T & \bar{c}_{p_1}^2 = (2, 6)^T \\ \bar{y}_{p_2}^1 = 2 & \bar{y}_{p_2}^2 = 9 & \bar{c}_{p_2}^1 = (7, 5)^T & \bar{c}_{p_2}^2 = (6, 2)^T \\ \bar{y}_{p_3}^1 = 5 & \bar{y}_{p_3}^2 = 2 & \bar{c}_{p_3}^1 = (5, 5)^T & \bar{c}_{p_3}^2 = (8, 2)^T \end{array}$$

and consider the family G consisting of one increasing function $G_{\bar{C}_{p_1}}^\epsilon$ for an $\epsilon > 0$ fixed. More precisely,

$$G_{\bar{C}_{p_1}}^\epsilon(\bar{C}_{p_i}) = \max_{\substack{k=1,2 \\ n=1,2}} \left\{ \epsilon \sum_{s=1}^2 \sum_{t=1}^2 (\bar{c}_{p_i,s}^t - \bar{c}_{p_1,s}^t) + \bar{c}_{p_i,k}^n - \bar{c}_{p_1,k}^n \right\}, p_i \in P_w.$$

Then the condition (6.4) does not hold, that is $G_{\bar{C}_{p_1}}^\epsilon(\bar{C}_{p_3}) = 8\epsilon + 6 > G_{\bar{C}_{p_1}}^\epsilon(\bar{C}_{p_2}) = 8\epsilon + 5$ and $\bar{C}_{p_3} \not\geq \bar{C}_{p_2}$. Obviously, \bar{Y} is a strong vector equilibrium, but not a G -equilibrium.

We now show that families of increasing functions satisfying the condition (6.4) do exist.

Lemma 6.2.3 *Let a feasible flow \bar{Y} be given. There exists $\epsilon_0 > 0$ such that for every $\epsilon \in (0, \epsilon_0)$, each of the families below of increasing functions satisfies the condition (6.4): $\mathcal{F}_1 = \{D_{\bar{C}_{p_i}}^\epsilon : p_i \in P\}$, $\mathcal{F}_2 = \{G_{\bar{C}_{p_i}}^\epsilon : p_i \in P\}$, $\mathcal{F}_3 = \{d_{\bar{C}_{p_i}}^\epsilon : p_i \in P\}$ and $\mathcal{F}_4 = \{g_{\bar{C}_{p_i}}^\epsilon : p_i \in P\}$.*

Proof. Consider the pairs (p_i, p_j) with $p_i, p_j \in P$ and define $\epsilon_{ij} = +\infty$ if $\bar{C}_{p_i} \geq \bar{C}_{p_j}$ and $\epsilon_{ij} = \epsilon(\bar{C}_{p_i}, \bar{C}_{p_j})$ if $\bar{C}_{p_i} \not\geq \bar{C}_{p_j}$, where $\epsilon(\bar{C}_{p_i}, \bar{C}_{p_j})$ is obtained by Lemma 2.0.15. Then set $\epsilon_0 = \min \{\epsilon_{ij} : p_i, p_j \in P\}$. We show that for all $\epsilon \in (0, \epsilon_0)$, \mathcal{F}_1 satisfies the condition (6.4). Indeed, let $p_j, p_{j'} \in P$ be such that $D_{\bar{C}_{p_i}}^\epsilon(\bar{C}_{p_j}) > D_{\bar{C}_{p_i}}^\epsilon(\bar{C}_{p_{j'}})$ for every $p_i \in P$. If $\bar{C}_{p_j} \not\geq \bar{C}_{p_{j'}}$, then for $p_i \equiv p_j$ we have, in view of Lemma 2.0.15, $0 = D_{\bar{C}_{p_j}}^\epsilon(\bar{C}_{p_j}) > D_{\bar{C}_{p_j}}^\epsilon(\bar{C}_{p_{j'}}) \geq 0$, which

is impossible.

The proof for \mathcal{F}_2 is similar with use of Lemma 2.0.15 for $G_{\overline{C}_{p_i}}^\epsilon$. The same argument is applied to the families \mathcal{F}_3 and \mathcal{F}_4 with the help of Lemma 2.0.14 instead of Lemma 2.0.15. \square

Corollary 6.2.4 *Let a feasible flow \overline{Y} be given. Then it is a strong vector equilibrium if and only if it is an \mathcal{F}_i -equilibrium for some $i \in \{1, \dots, 4\}$.*

Proof. It follows from Lemmas 6.2.1 and 6.2.3. \square

6.2.2 Efficiency

We wish to express strong vector equilibrium flows in terms of efficient points of the value set of the criteria function. This result is an improvement of Theorem 6.1 of [38] and will be used in the algorithm of Section 6.3. For $w \in W$, let us denote

$$\begin{aligned} L_w &:= \{p_i \in P_w \text{ such that } \overline{Y}_{p_i} = L_{p_i}\} \\ U_w &:= \{p_i \in P_w \text{ such that } \overline{Y}_{p_i} = U_{p_i}\} \\ E_w &:= \{p_i \in P_w \text{ such that } L_{p_i} \leq \overline{Y}_{p_i} \leq U_{p_i}\} \end{aligned}$$

and for an index set I , $\overline{C}_I = \{\overline{C}_i, i \in I\}$.

Theorem 6.2.5 *Let \overline{Y} be a feasible flow. It is a strong vector equilibrium if and only if for every $w \in W$ the following conditions hold*

- (i) $(\overline{C}_{L_w} + \mathbb{R}_+^{l \times q} \setminus \{0\}) \cap (\overline{C}_{U_w} \cup \overline{C}_{E_w}) = \emptyset$;
- (ii) $(\overline{C}_{U_w} - \mathbb{R}_+^{l \times q} \setminus \{0\}) \cap (\overline{C}_{L_w} \cup \overline{C}_{E_w}) = \emptyset$;
- (iii) \overline{C}_{E_w} is self-maximal.

Proof. We prove first the "if" part. Assume that \overline{Y} satisfies the three conditions of the theorem. Let $w \in W$ and $p_\alpha, p_\beta \in P_w$ such that $\overline{C}_{p_\alpha} \geq \overline{C}_{p_\beta}$. If $p_\alpha \in L_w$ or $p_\beta \in U_w$, we are done. If not, we consider four possible cases:

- a) $p_\alpha \in U_w$ and $p_\beta \in E_w$. This implies \overline{C}_{p_β} belongs to the intersection of $(\overline{C}_{U_w} - \mathbb{R}_+^{l \times q} \setminus \{0\}) \cap \overline{C}_{E_w}$, contradicting (ii).
- b) $p_\alpha \in E_w$ and $p_\beta \in E_w$. This case is impossible because \overline{C}_{E_w} is self-maximal.
- c) $p_\alpha \in U_w$ and $p_\beta \in L_w$. Then \overline{C}_{p_α} belongs to $(\overline{C}_{L_w} + \mathbb{R}_+^{l \times q} \setminus \{0\}) \cap \overline{C}_{U_w}$, contradicting (i).
- d) $p_\alpha \in E_w$ and $p_\beta \in L_w$. Similarly, \overline{C}_{p_α} belongs to $(\overline{C}_{L_w} + \mathbb{R}_+^{l \times q} \setminus \{0\}) \cap \overline{C}_{E_w}$, contradicting (i) too.

For the converse, let \overline{Y} be a strong vector equilibrium of the network. If (i) does not hold, then one can find $p_\alpha \in L_w$ and $p_\beta \in E_w \cup U_w$ such that $\overline{C}_{p_\alpha} \leq \overline{C}_{p_\beta}$. The latter inequality implies either $p_\alpha \in U_w$ or $p_\beta \in L_w$ which is not the case. Similarly, if (ii) does not hold, then one can find $p_\alpha \in U_w$ and $p_\beta \in E_w \cup L_w$ such that $\overline{C}_{p_\alpha} \geq \overline{C}_{p_\beta}$, which leads to the same contradiction. Finally, if (iii) does not hold, then one has $\overline{C}_{p_\alpha} \geq \overline{C}_{p_\beta}$ for some $p_\alpha, p_\beta \in E_w$. This is in contradiction with the definition of strong vector equilibrium. \square

It is worthwhile noticing that conditions (i) and (ii) can be substituted respectively by

$$\begin{aligned} (i') & (\text{Min}(\overline{C}_{L_w}) + \mathbb{R}_+^{l \times q} \setminus \{0\}) \cap (\overline{C}_{U_w} \cup \overline{C}_{E_w}) = \emptyset; \\ (ii') & (\text{Max}(\overline{C}_{U_w}) - \mathbb{R}_+^{l \times q} \setminus \{0\}) \cap (\overline{C}_{L_w} \cup \overline{C}_{E_w}) = \emptyset. \end{aligned}$$

Indeed, as the sets \overline{C}_{L_w} and \overline{C}_{U_w} are finite, one has $\overline{C}_{L_w} + \mathbb{R}_+^{l \times q} \setminus \{0\} = \text{Min}(\overline{C}_{L_w}) + \mathbb{R}_+^{l \times q} \setminus \{0\}$ and $\overline{C}_{U_w} - \mathbb{R}_+^{l \times q} \setminus \{0\} = \text{Max}(\overline{C}_{U_w}) - \mathbb{R}_+^{l \times q} \setminus \{0\}$. We also observe that applying Theorem 6.2.5 with (i') and (ii') above to each product j , we deduce Theorem 6.1 of [38] (Remark that in Theorem 6.1 of [38], $\mathbb{R}_+^l \setminus \{0\}$ should be instead of \mathbb{R}_+^l).

6.2.3 Variational inequality problems

Scalarization is a method widely used in the theory of multiple criteria decision making and in vector optimization. It has already been developed in the existing literature [24, 33, 31] and [38]. However, all variational inequality problems obtained in these works produce only solutions which are weak equilibrium flows. In this section we employ augmented increasing functions to construct variational inequality problems which are equivalent to the problem of finding strong vector equilibrium flows. For a given feasible flow $\bar{Y} \in K$, we shall make use of the following notations:

$$\begin{aligned}\Gamma &:= \{(p_i, w, j) : j \in \{1, \dots, q\}, w \in W, p_i \in P_w\} \\ A_w &:= \{p_\alpha \in P_w \text{ such that } \bar{Y}_{p_\alpha} \geq L_{p_\alpha}\} \\ B_w &:= \{p_\beta \in P_w \text{ such that } \bar{Y}_{p_\beta} \leq U_{p_\beta}\}\end{aligned}$$

Let h be a real function of the variables (a, b) for $a, b \in \{C_{p_i}(Y) : p_i \in P, Y \in K\}$. Consider the following variational inequality problem: Find \bar{Y} such that

$$\sum_{(p_i, w, j) \in \Gamma} \left(\min_{p_\alpha \in A_w} h(\bar{C}_{p_\alpha}, \bar{C}_{p_i}) \right) (y_{p_i}^j - \bar{y}_{p_i}^j) \geq 0 \text{ for every } Y \in K. \quad (6.5)$$

We wish to prove that this problem is equivalent to the network equilibrium problem under our consideration in the sense that a feasible flow \bar{Y} is a strong vector equilibrium if and only if it is a solution to (6.5) under a suitable hypothesis.

Lemma 6.2.6 *If the function h is increasing with respect to the second variable and if the condition (6.2) is satisfied, then every solution of the variational problem (6.5) is a strong vector equilibrium. Conversely, if the function h satisfies the condition: $a \not\geq b \Leftrightarrow h(a, b) \geq 0$, then every strong vector equilibrium is a solution of the variational problem (6.5).*

Proof. Suppose that \bar{Y} is not a strong vector equilibrium. Then there exists $w_0 \in W$ and $p_{\alpha_0}, p_{\beta_0} \in P_{w_0}$ such that

$$\bar{C}_{p_{\alpha_0}} \geq \bar{C}_{p_{\beta_0}}, \bar{y}_{p_{\alpha_0}} \geq L_{p_{\alpha_0}}, \bar{y}_{p_{\beta_0}} \leq U_{p_{\beta_0}}.$$

By (6.2), there exists $j_0 \in \{1, \dots, q\}$ such that $l_{p_{\alpha_0}}^{j_0} < \bar{y}_{p_{\alpha_0}}^{j_0}$ and $\bar{y}_{p_{\beta_0}}^{j_0} < u_{p_{\beta_0}}^{j_0}$. Construct a flow Y as follows

$$y_{p_i}^j = \begin{cases} \bar{y}_{p_i}^j & \text{if } p_i \notin \{p_{\alpha_0}, p_{\beta_0}\}, j \in \{1, \dots, q\}, \text{ or } p_i \in \{p_{\alpha_0}, p_{\beta_0}\}, j \in \{1, \dots, q\} \setminus \{j_0\} \\ \bar{y}_{p_i}^j - \delta & \text{if } p_i = p_{\alpha_0}, j = j_0 \\ \bar{y}_{p_i}^j + \delta & \text{if } p_i = p_{\beta_0}, j = j_0 \end{cases}$$

where $0 < \delta \leq \min\{\bar{y}_{p_{\alpha_0}}^{j_0} - l_{p_{\alpha_0}}^{j_0}, u_{p_{\beta_0}}^{j_0} - \bar{y}_{p_{\beta_0}}^{j_0}\}$. It is clear that Y is a feasible flow. Moreover,

$$\begin{aligned}\sum_{(p_i, w, j) \in \Gamma} \left(\min_{p_\alpha \in A_w} h(\bar{C}_{p_\alpha}, \bar{C}_{p_i}) \right) (y_{p_i}^j - \bar{y}_{p_i}^j) &= \sum_{p_i} \left(\min_{p_\alpha \in A_{w_0}} h(\bar{C}_{p_\alpha}, \bar{C}_{p_i}) \right) (y_{p_i}^{j_0} - \bar{y}_{p_i}^{j_0}) \\ &= \left(\min_{p_\alpha \in A_{w_0}} h(\bar{C}_{p_\alpha}, \bar{C}_{p_{\alpha_0}}) \right) (y_{p_{\alpha_0}}^{j_0} - \bar{y}_{p_{\alpha_0}}^{j_0}) + \left(\min_{p_\alpha \in A_{w_0}} h(\bar{C}_{p_\alpha}, \bar{C}_{p_{\beta_0}}) \right) (y_{p_{\beta_0}}^{j_0} - \bar{y}_{p_{\beta_0}}^{j_0}) \\ &= \delta \left(\min_{p_\alpha \in A_{w_0}} h(\bar{C}_{p_\alpha}, \bar{C}_{p_{\beta_0}}) - \min_{p_\alpha \in A_{w_0}} h(\bar{C}_{p_\alpha}, \bar{C}_{p_{\alpha_0}}) \right) \\ &\leq \delta \min_{p_\alpha \in A_{w_0}} [h(\bar{C}_{p_\alpha}, \bar{C}_{p_{\beta_0}}) - h(\bar{C}_{p_\alpha}, \bar{C}_{p_{\alpha_0}})] \\ &< 0,\end{aligned}$$

in which the last inequality follows from the fact that $h(\overline{C}_{p_\alpha}, \cdot)$ is increasing with respect to the second variable for every fixed \overline{C}_{p_α} . This shows that \overline{y} is not a solution of (6.5).

For the converse, suppose that \overline{Y} is a strong vector equilibrium. It follows that $\overline{C}_{p_\alpha} \not\geq \overline{C}_{p_\beta}$ for all $p_\alpha \in A_w, p_\beta \in B_w$. According to our assumption, we have $h(\overline{C}_{p_\alpha}, \overline{C}_{p_\beta}) \geq 0$ for every $p_\alpha \in A_w$ and $p_\beta \in B_w$. Let Y be a feasible flow. Set $t_w := \min_{p_\alpha \in A_w, p_\beta \in B_w} h(\overline{C}_{p_\alpha}, \overline{C}_{p_\beta}) \geq 0$,

and for each path $p_i \in P_w$ consider three possible values of the difference $\min_{p_\alpha \in A_w} h(\overline{C}_{p_\alpha}, \overline{C}_{p_i}) - t_w$.

Case 1: $\min_{p_\alpha \in A_w} h(\overline{C}_{p_\alpha}, \overline{C}_{p_i}) - t_w < 0$. Then $p_i \notin B_w$ and $y_{p_i}^j - \overline{y}_{p_i}^j = y_{p_i}^j - u_{p_i}^j \leq 0$. Hence,

one has $(\min_{p_\alpha \in A_w} h(\overline{C}_{p_\alpha}, \overline{C}_{p_i}) - t_w)(y_{p_i}^j - \overline{y}_{p_i}^j) \geq 0$. *Case 2:* $\min_{p_\alpha \in A_w} h(\overline{C}_{p_\alpha}, \overline{C}_{p_i}) - t_w > 0$. Then

$\min_{p_\alpha \in A_w} h(\overline{C}_{p_\alpha}, \overline{C}_{p_i}) > 0$. Moreover, as $\min_{p_\alpha \in A_w} h(\overline{C}_{p_\alpha}, \overline{C}_{p_s}) \leq 0, \forall p_s \in A_w$, we deduce $p_i \notin A_w$

and hence $y_{p_i}^j - \overline{y}_{p_i}^j = y_{p_i}^j - l_{p_i}^j \geq 0$. This implies that $(\min_{p_\alpha \in A_w} h(\overline{C}_{p_\alpha}, \overline{C}_{p_i}) - t_w)(y_{p_i}^j - \overline{y}_{p_i}^j) \geq 0$.

Case 3: $\min_{p_\alpha \in A_w} h(\overline{C}_{p_\alpha}, \overline{C}_{p_i}) - t_w = 0$. It is clear that $(\min_{p_\alpha \in A_w} h(\overline{C}_{p_\alpha}, \overline{C}_{p_i}) - t_w)(y_{p_i}^j - \overline{y}_{p_i}^j) = 0$.

By taking the constraint (3.15) into account we deduce that

$$\begin{aligned} \sum_{(p_i, w, j) \in \Gamma} \left(\min_{p_\alpha \in A_w} h(\overline{C}_{p_\alpha}, \overline{C}_i) \right) (y_{p_i}^j - \overline{y}_{p_i}^j) &\geq \sum_{(p_i, w, j) \in \Gamma} t_w (y_{p_i}^j - \overline{y}_{p_i}^j) \\ &\geq \sum_{(p_i, w, j) \in \Gamma} t_w (d_w^j - \overline{d}_w^j) = 0, \end{aligned}$$

which proves that \overline{y} is a solution of (6.5). \square

We are now able to construct specific variational inequality problems to obtain strong vector equilibrium flows by using augmented signed distance functions.

Theorem 6.2.7 *Let \overline{Y} be a feasible flow. If it satisfies the condition (6.2) and*

$$\sum_{(p_i, w, j) \in \Gamma} \left(\min_{p_\alpha \in A_w} D_{\overline{C}_{p_\alpha}}^\epsilon(\overline{C}_{p_i}) \right) (y_{p_i}^j - \overline{y}_{p_i}^j) \geq 0 \text{ for every } Y \in K, \quad (6.6)$$

for some $\epsilon > 0$, then it is a strong vector equilibrium. Conversely, if \overline{Y} is a strong vector equilibrium, then there is $\epsilon_0 > 0$ such that it satisfies (6.6) for all $\epsilon \in (0, \epsilon_0)$.

Proof. For the first part of the theorem, we set $h(a, b) = D_a^\epsilon(b)$. By the definition this function is increasing with respect to b . In view of Lemma 6.2.6 every solution of (6.6) is a strong vector equilibrium. To prove the second part of the theorem we apply Lemma 2.0.15 to find some $\epsilon_0 > 0$ such that for every $\epsilon \in (0, \epsilon_0)$, $a, b \in \{\overline{C}_{p_i} : p_i \in P\}$ one has that $a \not\geq b$ if and only if $D_a^\epsilon(b) \geq 0$. The proof now follows from Lemma 6.2.6. \square

The same argument of proof shows that the above theorem is true when the augmented signed distance functions $D_a^\epsilon(b)$ are replaced by the augmented smallest increasing functions $G_a^\epsilon(b)$. In the next result we have a variational inequality problem obtained by augmented biggest functions.

Theorem 6.2.8 *Let \overline{Y} be a feasible flow. If it satisfies the condition (6.2) and*

$$\sum_{(p_i, w, j) \in \Gamma} \left(\max_{p_\beta \in B_w} g_{\overline{C}_{p_\beta}}^\epsilon(\overline{C}_{p_i}) \right) (y_{p_i}^j - \overline{y}_{p_i}^j) \geq 0 \text{ for every } Y \in K, \quad (6.7)$$

for some $\epsilon > 0$, then it is a strong vector equilibrium. Conversely, if \bar{Y} is a strong vector equilibrium, then there is $\epsilon_0 > 0$ such that it satisfies (6.7) for all $\epsilon \in (0, \epsilon_0)$.

Proof. Apply the same argument as in the proof of the preceding theorem with a suitable arrangement of Lemma 2.0.14. \square

This result is a correction to Theorem 5.1 of [38], in which "min" was mistakenly employed instead of "max". Again, it remains true when the functions $g_a^\epsilon(b)$ are substituted by the functions $d_a^\epsilon(b)$. In the above theorems, if the condition (6.2) is not satisfied, then a solution of the variational inequality problem (6.6) (or (6.7)), is not necessarily a strong vector equilibrium. This is seen by the following example.

Example 6.2.9 Consider a network problem with only one pair of origin-destination nodes $w = (x, x')$, two criteria and two products to traverse the network with two available paths: $P_w = \{p_1, p_2\}$. Assume that $d_w^1 = 7$, $d_w^2 = 19$, $l_{p_i}^j = 2$, $u_{p_i}^j = 10$ for $p_i, p_j \in P_w$ and

$$\begin{array}{cccc} \bar{y}_{p_1}^1 = 2 & \bar{y}_{p_1}^2 = 9 & \bar{c}_{p_1}^1 = (15, 35)^T & \bar{c}_{p_1}^2 = (25, 20)^T \\ \bar{y}_{p_2}^1 = 5 & \bar{y}_{p_2}^2 = 10 & \bar{c}_{p_2}^1 = (10, 20)^T & \bar{c}_{p_2}^2 = (15, 12)^T \end{array}$$

Then, \bar{Y} does not satisfy the condition (6.2). In fact, $\bar{Y}_{p_1} \geq L_{p_1}$ and $\bar{Y}_{p_2} \leq U_{p_2}$, and there is not any $j \in \{1, 2\}$ such that $\bar{y}_{p_1}^j > l_{p_1}^j$ and $\bar{y}_{p_2}^j < u_{p_2}^j$. Moreover, $D_{\bar{C}_{p_1}}^\epsilon(\bar{C}_{p_2}) < 0$ since $\bar{C}_{p_1} \geq \bar{C}_{p_2}$. Then, for all $Y \in K$ one has

$$\begin{aligned} \sum_{(p_i, w, j) \in \Gamma} \left(\min_{\alpha \in A_w} D_{\bar{C}_{p_\alpha}}^\epsilon(\bar{C}_{p_i}) \right) (y_{p_i}^j - \bar{y}_{p_i}^j) &= D_{\bar{C}_{p_1}}^\epsilon(\bar{C}_{p_2}) (y_{p_2}^1 - \bar{y}_{p_2}^1) + D_{\bar{C}_{p_1}}^\epsilon(\bar{C}_{p_2}) (y_{p_2}^2 - \bar{y}_{p_2}^2) \\ &= D_{\bar{C}_{p_1}}^\epsilon(\bar{C}_{p_2}) (y_{p_2}^1 - 5) + D_{\bar{C}_{p_1}}^\epsilon(\bar{C}_{p_2}) (y_{p_2}^2 - 10). \end{aligned}$$

We have $y_{p_2}^1 \leq 5$ since $y_{p_1}^1 \geq 2$. Hence,

$$\sum_{(p_i, w, j) \in \Gamma} \left(\min_{\alpha \in A_w} D_{\bar{C}_\alpha}^\epsilon(\bar{C}_{p_i}) \right) (y_{p_i}^j - \bar{y}_{p_i}^j) \geq 0 \text{ for all } Y \in K,$$

which shows that the path flow \bar{Y} solves the variational inequality problem (6.6). However, \bar{Y} is not a strong vector equilibrium as $\bar{C}_{p_1} \geq \bar{C}_{p_2}$ but $\bar{Y}_{p_1} \neq L_{p_1}$ and $\bar{Y}_{p_2} \neq U_{p_2}$.

6.3 Algorithms

As we saw in Section 6.1 finding a weak vector equilibrium of a multi-product and multi-criteria network is quite clear. It suffices to fix one product and to consider a model with this single product while keeping the other products unchanged and with a single cost function being the sum of all components of the vector cost function. This amounts to saying that the problem of finding a weak vector equilibrium is computationally equivalent to a classical network equilibrium problem, or a variational inequality problem. The situation is more complicated with strong vector equilibrium. In fact, firstly, strong vector equilibrium does not always exist even when the cost function is linear and the set of feasible flows is compact (see Example 6.1.1). Secondly, almost all equivalent variational inequality problems in our disposal do not satisfy sufficient conditions needed for convergence of existing algorithms, namely the functions determining the associated variational inequalities are not monotone. Thirdly, a solution to a variational inequality problem is not necessarily a strong vector equilibrium, which requests an additional computational effort to check

certain sufficient conditions for strong vector equilibrium. Because of these reasons, solving a multi-product and multi-criteria network equilibrium problem is a challenging task. In this section we develop an algorithm for finding a strong vector equilibrium by using the variational inequality generated by the augmented signed distance function and discuss its implementability through a number of numerical examples. As in multi-objective programming, by solving a multi-criteria network equilibrium problem we mean generating the entire set of equilibrium flows or at least a representative part of it. In this sense our algorithm can be seen as a first step towards a complete resolution of the problem.

We begin with expressing the variational inequality problem (6.6) in the standard form:

$$\langle F(\bar{Y}), Y - \bar{Y} \rangle \geq 0 \quad \forall Y \in K. \quad (6.8)$$

To obtain an explicit form of F , we assume that $W = \{w_1, \dots, w_s\}$, that is there are s origin-destination pairs in the network. For each $w_t, t = 1, \dots, s$ we define

$$K_{w_t} = \{Y_{w_t} \in \mathbb{R}^{|P_{w_t}| \times q} : l_{p_i}^j \leq y_{p_i}^j \leq u_{p_i}^j, j = 1, \dots, q; p_i \in P_{w_t} \text{ and } \sum_{p_i \in P_{w_t}} y_{p_i}^j = d_{w_t}^j, j = 1, \dots, q\},$$

where $\sum_{t=1}^s |P_{w_t}| = m$ with $|\cdot|$ denoting the cardinality of a set. Then the feasible set K is decomposed by

$$K = \prod_{t=1}^s K_{w_t}. \quad (6.9)$$

We may assume that the components of Y are grouped according to origin-destination pairs: $Y = (Y_{w_1}, Y_{w_2}, \dots, Y_{w_s})$ and define $F(Y) = (F_{w_1}(Y), \dots, F_{w_t}(Y), \dots, F_{w_s}(Y))$ with

$$F_{w_t}(Y) = (F_{w_t}^1(Y), \dots, F_{w_t}^{|P_{w_t}|}(Y)) \quad (6.10)$$

where for each $p_i \in \{p_1, \dots, p_{|P_{w_t}|}\}$, the vector $F_{w_t}^i(Y)$ has q components, which are all equal to $\min_{p_\alpha \in A_{w_t}} D_{C_{p_\alpha}(Y)}^\epsilon(C_{p_i}(Y))$.

Below we propose an algorithm to find a strong vector equilibrium of our model in which the modified projection method (see [5, 42, 43, 46]) is applied to solve the associated variational inequality problem (6.8):

Step 0: Initialization

Start with a feasible flow $Y^0 \in K$. Select a coefficient $\rho \in (0, 1)$ for the modified projection method, a positive coefficient ϵ and a small error tolerance $\delta > 0$. Set $k := 1$.

Step 1: Solving (6.8)

- Find A_{w_t} and compute $D_{C_{p_\alpha}(Y^{k-1})}^\epsilon(C_{p_i}(Y^{k-1}))$ for all $\alpha \in A_{w_t}, p_i \in P_{w_t}, w_t \in W$.
- Compute the functions $F_{w_t}(Y^{k-1})$ for all $w_t \in W$

$$F_{w_t}(Y^{k-1}) = \left(\min_{\alpha \in A_{w_t}} D_{C_{p_\alpha}(Y^{k-1})}^\epsilon(C_{p_1}(Y^{k-1})), \dots, \min_{\alpha \in A_{w_t}} D_{C_{p_\alpha}(Y^{k-1})}^\epsilon(C_{p_1}(Y^{k-1})), \right. \\ \left. \dots, \min_{p_\alpha \in A_{w_t}} D_{C_{p_\alpha}(Y^{k-1})}^\epsilon(C_{|P_{w_t}|}(Y^{k-1})), \dots, \min_{p_\alpha \in A_{w_t}} D_{C_{p_\alpha}(Y^{k-1})}^\epsilon(C_{|P_{w_t}|}(Y^{k-1})) \right).$$

- Solve s linearized variational inequality subproblems:

$$\left\langle [F_{w_t}(Y^{k-1}) + (U - Y_{w_t}^{k-1})]^T, U' - U \right\rangle \geq 0, \forall U' \in K_{w_t} \quad (6.11)$$

by using the modified projection method. Let

$$U = Y_{w_t}^{k-1} - \rho[Y_{w_t}^{k-1} - P_{K_{w_t}}(Y_{w_t}^{k-1} - \rho F_{w_t}(Y_{w_t}^{k-1}))] \quad (6.12)$$

be a solution.

Set $Y_{w_t}^k = U$ and $Y^k = (Y_{w_t}^k)_{w_t \in W}$.

Step 2: Convergence verification

If $\|Y^k - Y^{k-1}\| \leq \delta$, then stop and go to Step 3; otherwise, set $k := k + 1$ and go to Step 1.

Step 3: Final verification

For Y^k obtained in Step 2, compute $C(Y^k)$ and check three conditions (i) – (iii) of Theorem 6.2.5. If three conditions are satisfied, then Y^k is a strong vector equilibrium. Otherwise, set $\epsilon = \epsilon/2$ and return to Step 1. If $\epsilon < \delta$ and no strong vector equilibrium is found, then stop. It is considered that the network has no strong vector equilibrium.

Before presenting numerical examples we discuss some convergence aspects of the algorithm. We assume throughout that the cost function $C(\cdot)$ is continuous, which is a sufficient condition for existence of solutions to the variational inequality problem (6.8).

Convergence under condition (6.2)

It follows from the definition of the functions F that with $q > 1$ it is never strictly monotone. Therefore the uniqueness of solution to the variational inequality problem (6.8) is not guaranteed [43]. However, if the condition (6.2) is satisfied, then any solution of (6.8) is a strong vector equilibrium (Theorem 6.2.7). Under this hypothesis the last step of the algorithm is unnecessary. Moreover, for every $\delta > 0$ the algorithm will terminate after a finite number of iterations and yields an approximate solution of (6.8) which is considered as an approximate strong vector equilibrium of the network. When δ tends to zero, that approximate solution converges to an exact solution of (6.8), which is also a strong vector equilibrium of the network.

Convergence without condition (6.2)

The network may have a strong vector equilibrium without condition (6.2). According to Theorem 6.2.7 at least one of the solutions of the variational inequality problem (6.8) is a strong vector equilibrium when ϵ is sufficiently small. Of course, if a solution of the variational inequality problem (6.8) with ϵ small is unique, it must be a unique strong vector equilibrium and as before no verification at Step 3 is requested. When a solution to (6.8) is not unique, the last step that verifies the necessary and sufficient conditions for strong vector equilibrium, is needed.

Nonexistence of strong vector equilibrium

Except for the models without capacity constraints (or when u_{ij} are large), verification of condition (6.2) is not easy, and therefore it is not known a priori whether a strong vector equilibrium exists or not. We observe that if the model has no strong vector equilibrium, then at each iteration we decrease ϵ by half because the conditions of Theorem 6.2.7 are not satisfied, and so, after a finite number of iterations we have $\epsilon < \delta$, by which the algorithm terminates. When the algorithm stops without producing strong vector equilibrium, there are two possible situations: either the solution obtained by the projection method is a wrong solution of (6.8) and cannot be a strong vector equilibrium, or that solution is a solution of (6.8), but it is not a strong vector equilibrium. Therefore, in the implementation of the algorithm, when no strong vector equilibrium is detected in Step 3 with $\epsilon < \delta$, we restart it with a new initial flow. If after a number of trials no equilibrium is found, we may consider that the network has no strong vector equilibrium.

6.4 Numerical examples

In this subsection we give some examples to illustrate the algorithm. In Example 6.4.1, with $\epsilon > 0$ relatively large, a solution of the variational inequality problem (6.8) obtained in Step 2 is not a strong vector equilibrium, but with $\epsilon > 0$ small, we do obtain a strong vector equilibrium after some iterations.

Example 6.4.1 Consider a network problem with only one pair of origin-destination nodes $w = (x, x')$, two criteria and two products to traverse the network with three available paths: $P_w = \{p_1, p_2, p_3\}$. Assume that $d_w^1 = 20$, $d_w^2 = 25$ and

$$\begin{array}{cccccc} l_{p_1}^1 = 1 & l_{p_2}^1 = 0 & l_{p_3}^1 = 1 & l_{p_1}^2 = 2 & l_{p_2}^2 = 2 & l_{p_3}^2 = 3 \\ u_{p_1}^1 = 12 & u_{p_2}^1 = 15 & u_{p_3}^1 = 10 & u_{p_1}^2 = 10 & u_{p_2}^2 = 11 & u_{p_3}^2 = 15 \end{array}$$

$$\begin{array}{l} c_{p_1}^1(Y) = (y_{p_1}^1 + 10y_{p_1}^2 + y_{p_2}^1 + 50y_{p_3}^2, 2y_{p_1}^1 + 10y_{p_1}^2 + y_{p_2}^1 + 9y_{p_3}^2)^T \\ c_{p_2}^1(Y) = (3y_{p_2}^1 + 8y_{p_3}^2, 2y_{p_1}^1 + 9y_{p_2}^1)^T \\ c_{p_3}^1(Y) = (y_{p_1}^1 + 3y_{p_3}^1, 8y_{p_3}^1)^T \\ c_{p_1}^2(Y) = (y_{p_1}^2 + y_{p_2}^1, y_{p_1}^2 + 5y_{p_2}^1)^T \\ c_{p_2}^2(Y) = (y_{p_1}^1 + 3y_{p_2}^2, y_{p_2}^2)^T \\ c_{p_3}^2(Y) = (5y_{p_3}^2, 5y_{p_3}^2)^T \end{array}$$

Set the initial path flow $(y_{p_1}^1)^0 = 3, (y_{p_2}^1)^0 = 7, (y_{p_3}^1)^0 = 10, (y_{p_1}^2)^0 = 5, (y_{p_2}^2)^0 = 10, (y_{p_3}^2)^0 = 10, \rho = 0.6$ and $\delta = 0.0001$. For $\epsilon = 2.5$, after 42 iterations, we obtain the following solution of the variational inequality (6.8): $\bar{y}_{p_1}^1 = 1, \bar{y}_{p_2}^1 = 9.5968, \bar{y}_{p_3}^1 = 9.4032, \bar{y}_{p_1}^2 = 2, \bar{y}_{p_2}^2 = 8.3058, \bar{y}_{p_3}^2 = 14.6942$ which is not a strong vector equilibrium.

Nevertheless, when we reduce the value of ϵ until it is sufficiently small, for example $\epsilon = 0.1$, after 13 iterations, we obtain the following solution of the variational inequality (6.8): $\bar{y}_{p_1}^1 = 1, \bar{y}_{p_2}^1 = 15, \bar{y}_{p_3}^1 = 4, \bar{y}_{p_1}^2 = 2, \bar{y}_{p_2}^2 = 11, \bar{y}_{p_3}^2 = 12$ which is a strong vector equilibrium.

Note that without condition (6.2) a solution of (6.8) is not necessarily a strong vector equilibrium, but another initial flow may lead to a good solution (which is a strong vector equilibrium). In Example 6.4.2, we show that the time for finding strong vector equilibrium depends largely on the choice of the initial path flow Y^0 .

Example 6.4.2 Consider a network problem with only one pair of origin-destination nodes $w = (x, x')$, three criteria and three products to traverse in the network with three available paths: $P_w = \{p_1, p_2, p_3\}$. Assume that $d_{w_1} = 25, d_w^2 = 30, d_w^3 = 20$ and

$$\begin{array}{l} l_{p_1}^1 = l_{p_2}^2 = l_{p_3}^3 = l_{p_2}^2 = l_{p_3}^3 = 1 \quad ; \quad l_{p_1}^2 = l_{p_2}^1 = 0 \quad ; \quad l_{p_1}^3 = l_{p_3}^1 = 2 \\ u_{p_1}^1 = u_{p_2}^2 = u_{p_3}^3 = u_{p_1}^1 = u_{p_2}^2 = u_{p_3}^3 = 20 \quad ; \quad u_{p_2}^1 = u_{p_2}^2 = 25 \quad ; \quad u_{p_2}^3 = 15 \end{array}$$

$$\begin{aligned} c_{11}^1(Y) &= 18y_{p_1}^1 + y_{p_2}^1 + 14y_{p_3}^1 + 13y_{p_1}^2 + 19y_{p_2}^2 + 17y_{p_3}^2 + 17y_{p_3}^3; \\ c_{11}^2(Y) &= 20y_{p_1}^1 + 6y_{p_2}^1 + 3y_{p_3}^1 + 18y_{p_1}^2 + 12y_{p_2}^2 + 17y_{p_1}^3 + 6y_{p_2}^3 + 7y_{p_3}^3; \\ c_{11}^3(Y) &= 14y_{p_1}^1 + 13y_{p_3}^1 + y_{p_2}^2 + 11y_{p_1}^3 + 7y_{p_2}^3 + 17y_{p_3}^3; \end{aligned}$$

$$\begin{aligned} c_{12}^1(Y) &= 9y_{p_1}^1 + 9y_{p_2}^1 + y_{p_3}^1 + 16y_{p_1}^2 + 8y_{p_2}^2 + 15y_{p_3}^2 + 19y_{p_1}^3 + y_{p_2}^3 + 13y_{p_3}^3; \\ c_{12}^2(Y) &= 11y_{p_1}^1 + 5y_{p_2}^1 + 5y_{p_3}^1 + 18y_{p_1}^2 + 5y_{p_2}^2 + 3y_{p_3}^2 + 5y_{p_2}^3; \\ c_{12}^3(Y) &= 7y_{p_1}^1 + 2y_{p_2}^1 + 15y_{p_3}^1 + 5y_{p_1}^2 + 8y_{p_2}^2 + 6y_{p_3}^2 + 17y_{p_2}^3 + 12y_{p_3}^3; \end{aligned}$$

$$\begin{aligned} c_{13}^1(Y) &= 3y_{p_2}^1 + 17y_{p_3}^1 + 3y_{p_1}^2 + 6y_{p_3}^2 + 18y_{p_1}^3 + 4y_{p_2}^3 + 8y_{p_3}^3; \\ c_{13}^2(Y) &= 8y_{p_2}^1 + 9y_{p_3}^1 + 9y_{p_1}^2 + 4y_{p_3}^2 + 20y_{p_1}^3 + 9y_{p_2}^3; \\ c_{13}^3(Y) &= 12y_{p_1}^1 + 7y_{p_1}^2 + 7y_{p_3}^2 + 14y_{p_3}^3; \end{aligned}$$

$$\begin{aligned} c_{21}^1(Y) &= 18y_{p_1}^1 + 4y_{p_2}^1 + 16y_{p_3}^1 + 9y_{p_1}^2 + y_{p_2}^2 + 15y_{p_3}^2 + 17y_{p_1}^3 + 12y_{p_2}^3 + 6y_{p_3}^3; \\ c_{21}^2(Y) &= 7y_{p_1}^1 + 9y_{p_2}^1 + 4y_{p_3}^1 + 4y_{p_1}^2 + 3y_{p_2}^2 + 13y_{p_3}^2 + 5y_{p_2}^3 + 10y_{p_3}^3; \\ c_{21}^3(Y) &= 17y_{p_1}^1 + 13y_{p_2}^1 + 6y_{p_1}^2 + 7y_{p_2}^2 + 9y_{p_3}^2 + 8y_{p_1}^3 + 11y_{p_2}^3; \end{aligned}$$

$$\begin{aligned} c_{22}^1(Y) &= 15y_{p_1}^1 + 11y_{p_2}^1 + 7y_{p_3}^1 + 19y_{p_1}^2 + 2y_{p_2}^2 + 12y_{p_3}^2 + 16y_{p_1}^3 + 7y_{p_2}^3 + 5y_{p_3}^3; \\ c_{22}^2(Y) &= 17y_{p_1}^1 + 12y_{p_2}^1 + 20y_{p_3}^1 + 4y_{p_1}^2 + 10y_{p_2}^2 + 4y_{p_3}^2 + 7y_{p_1}^3 + 12y_{p_2}^3 + 17y_{p_3}^3; \\ c_{22}^3(Y) &= 18y_{p_1}^1 + 8y_{p_2}^1 + 11y_{p_2}^2 + 20y_{p_2}^2 + 8y_{p_3}^2 + 6y_{p_1}^3 + 9y_{p_2}^3 + 18y_{p_3}^3; \end{aligned}$$

$$\begin{aligned} c_{23}^1(Y) &= 16y_{p_1}^1 + 8y_{p_2}^1 + 9y_{p_3}^1 + 16y_{p_1}^2 + 20y_{p_2}^2 + 4y_{p_3}^2 + 3y_{p_1}^3 + y_{p_2}^3 + 13y_{p_3}^3; \\ c_{23}^2(Y) &= 20y_{p_1}^1 + 12y_{p_2}^1 + 5y_{p_1}^2 + 15y_{p_2}^2 + 8y_{p_3}^2 + 12y_{p_1}^3 + 20y_{p_2}^3 + 15y_{p_3}^3; \\ c_{23}^3(Y) &= 6y_{p_1}^1 + 13y_{p_2}^1 + 12y_{p_3}^1 + 20y_{p_1}^2 + 10y_{p_2}^2 + 13y_{p_3}^2 + 14y_{p_1}^3 + 11y_{p_2}^3 + 6y_{p_3}^3; \end{aligned}$$

$$\begin{aligned} c_{31}^1(Y) &= 17y_{p_1}^1 + 2y_{p_2}^1 + 20y_{p_3}^1 + 9y_{p_1}^2 + 8y_{p_2}^2 + 13y_{p_3}^2 + 16y_{p_1}^3 + 5y_{p_3}^3; \\ c_{31}^2(Y) &= 15y_{p_3}^1 + 4y_{p_1}^2 + 4y_{p_2}^2 + 2y_{p_3}^2 + 11y_{p_1}^3 + 2y_{p_2}^3 + 16y_{p_3}^3; \\ c_{31}^3(Y) &= 17y_{p_1}^1 + 4y_{p_2}^1 + 10y_{p_3}^1 + 7y_{p_1}^2 + 5y_{p_3}^2 + 13y_{p_1}^3 + 20y_{p_3}^3; \end{aligned}$$

$$\begin{aligned} c_{32}^1(Y) &= 7y_{p_1}^1 + 13y_{p_2}^1 + 16y_{p_3}^1 + y_{p_1}^2 + 5y_{p_2}^2 + 8y_{p_3}^2 + 5y_{p_3}^3; \\ c_{32}^2(Y) &= 3y_{p_2}^1 + 17y_{p_3}^1 + 7y_{p_1}^2 + 5y_{p_2}^2 + 19y_{p_3}^2 + 14y_{p_2}^3 + 5y_{p_3}^3; \\ c_{32}^3(Y) &= y_{p_1}^1 + 9y_{p_2}^1 + 15y_{p_3}^1 + 9y_{p_3}^2 + 8y_{p_2}^3 + 12y_{p_3}^3; \end{aligned}$$

$$\begin{aligned} c_{33}^1(Y) &= 20y_{p_1}^1 + 2y_{p_2}^1 + 11y_{p_3}^1 + 9y_{p_2}^2 + 11y_{p_2}^3 + y_{p_3}^3; \\ c_{33}^2(Y) &= 2y_{p_2}^1 + 20y_{p_1}^2 + 12y_{p_2}^2 + 11y_{p_2}^3 + 2y_{p_3}^3; \\ c_{33}^3(Y) &= 9y_{p_2}^1 + 15y_{p_2}^2 + 13y_{p_3}^2 + 7y_{p_2}^3 + 5y_{p_3}^3. \end{aligned}$$

With the initial path flow $(y_{p_1}^1)^0 = 2, (y_{p_2}^1)^0 = 6, (y_{p_3}^1)^0 = 17; (y_{p_1}^2)^0 = 3, (y_{p_2}^2)^0 = 20, (y_{p_3}^2)^0 = 7, (y_{p_1}^3)^0 = 15, (y_{p_2}^3)^0 = 3, (y_{p_3}^3)^0 = 2, \epsilon = 0.4, \rho = 0.1, \text{tolerance } \delta = 0.0001,$ we obtain a solution of the variational inequality (14): $\bar{y}_{p_1}^1 = 5.6863, \bar{y}_{p_2}^1 = 0.0002, \bar{y}_{p_3}^1 = 19.3135, \bar{y}_{p_1}^2 = 10.3780, \bar{y}_{p_2}^2 = 1.0006, \bar{y}_{p_3}^2 = 18.6214, \bar{y}_{p_1}^3 = 9.8743, \bar{y}_{p_2}^3 = 1.0001, \bar{y}_{p_3}^3 = 9.1256$ after 98 iterations and it is a strong vector equilibrium.

However the number of loops decreases a lot when we use another initial path flow. Particularly, with the initial path flow $(y_{p_1}^1)^0 = 3, (y_{p_2}^1)^0 = 7, (y_{p_3}^1)^0 = 15; (y_{p_1}^2)^0 = 7, (y_{p_2}^2)^0 = 10, (y_{p_3}^2)^0 = 13, (y_{p_1}^3)^0 = 1, (y_{p_2}^3)^0 = 5, (y_{p_3}^3)^0 = 14, \epsilon = 0.4, \rho = 0.1, \text{tolerance } \delta = 0.0001,$ after 55 iterations, we obtain a solution of the variational inequality (14): $\bar{y}_{p_1}^1 = 6.2591, \bar{y}_{p_2}^1 = 0.8254, \bar{y}_{p_3}^1 = 17.9156, \bar{y}_{p_1}^2 = 10.6192, \bar{y}_{p_2}^2 = 2.3925, \bar{y}_{p_3}^2 = 16.9883, \bar{y}_{p_1}^3 = 3.2446, \bar{y}_{p_2}^3 = 1.2637, \bar{y}_{p_3}^3 = 15.4917$ and it is also a strong vector equilibrium.

In the proof of Theorem 6.1.2 we used a variational inequality problem in which the function is given by the sum of all components of the cost function of q products, which corresponds to the weighted method with equal weight for every criterion and every product. This problem seems to be much simpler than the ones developed in Subsection 6.2.3 and

in our algorithm. However, in general it is not sufficiently subtle to solve multi-criteria network equilibrium problems because of lack of convexity. The example below shows that the weighted sum method does not find strong vector equilibrium, while the method by augmented increasing functions does.

Example 6.4.3 Consider a network problem with only one pair of origin-destination nodes $w = (x, x')$, two criteria and two products to traverse the network with three available paths: $P_w = \{p_1, p_2, p_3\}$. Assume that $d_w^1 = 15$, $d_w^2 = 20$ and

$$\begin{array}{llllll} l_{p_1}^1 = 1 & l_{p_2}^1 = 0 & l_{p_3}^1 = 2 & l_{p_1}^2 = 1 & l_{p_2}^2 = 1 & l_{p_3}^2 = 1 \\ u_{p_1}^1 = 15 & u_{p_2}^1 = 20 & u_{p_3}^1 = 10 & u_{p_1}^2 = 20 & u_{p_2}^2 = 17 & u_{p_3}^2 = 25 \end{array}$$

$$\begin{array}{ll} c_{p_1}^1(Y) = (18y_{p_1}^1, 29y_{p_1}^1)^T & c_{p_1}^2(Y) = (37y_{p_1}^2, 32y_{p_1}^2)^T \\ c_{p_2}^1(Y) = (34y_{p_2}^1, 30y_{p_2}^1)^T & c_{p_2}^2(Y) = (39y_{p_2}^2, 32y_{p_2}^2)^T \\ c_{p_3}^1(Y) = (23y_{p_3}^1, 26y_{p_3}^1)^T & c_{p_3}^2(Y) = (39y_{p_3}^2, 32y_{p_3}^2)^T \end{array}$$

Set the initial path flow $(y_{p_1}^1)^0 = 5$, $(y_{p_2}^1)^0 = 8$, $(y_{p_3}^1)^0 = 2$, $(y_{p_1}^2)^0 = 10$, $(y_{p_2}^2)^0 = 5$, $(y_{p_3}^2)^0 = 5$, $\rho = 0.5$ and $\delta = 0.0001$. When using the method of taking the sum of all components of the cost function, after 200 iterations, we obtain the following solution of the associated variational inequality: $\bar{y}_{p_1}^1 = 5.0000$, $\bar{y}_{p_2}^1 = 6.8571$, $\bar{y}_{p_3}^1 = 3.1429$, $\bar{y}_{p_1}^2 = 5.8571$, $\bar{y}_{p_2}^2 = 10.1429$, $\bar{y}_{p_3}^2 = 4.0000$ which is not a strong vector equilibrium.

Nevertheless, using our algorithm with ϵ sufficiently small, for example $\epsilon = 0.0001$, after 42 iterations, we obtain the following solution of the variational inequality (6.8): $\bar{y}_{p_1}^1 = 4.9946$, $\bar{y}_{p_2}^1 = 4.4553$, $\bar{y}_{p_3}^1 = 5.5501$, $\bar{y}_{p_1}^2 = 7.0112$, $\bar{y}_{p_2}^2 = 5.6238$, $\bar{y}_{p_3}^2 = 7.3650$ which is a strong vector equilibrium.

In the last example we randomly generate the cost functions to see what is the percentage of problems that have strong vector equilibrium flows according to our algorithm.

Example 6.4.4 Consider a network problem with one pair of origin-destination nodes $w = (x, x')$, two criteria and two products to traverse the network with three available paths: $P_w = \{p_1, p_2, p_3\}$. The other data are given as below $d_w^1 = 15$, $d_w^2 = 20$, $l_{p_i}^j = 2$, $u_{p_i}^j = 15$ for $p_i \in P_w$ and $j = 1, 2$, and

$$\begin{array}{ll} c_{p_1}^1(Y) = (a_{11}^1 y_{p_1}^1, a_{11}^2 y_{p_1}^1)^T & c_{p_1}^2(Y) = (a_{12}^1 y_{p_1}^2, a_{12}^2 y_{p_1}^2)^T \\ c_{p_2}^1(Y) = (a_{21}^1 y_{p_2}^1, a_{21}^2 y_{p_2}^1)^T & c_{p_2}^2(Y) = (a_{22}^1 y_{p_2}^2, a_{22}^2 y_{p_2}^2)^T \\ c_{p_3}^1(Y) = (a_{31}^1 y_{p_3}^1, a_{31}^2 y_{p_3}^1)^T & c_{p_3}^2(Y) = (a_{32}^1 y_{p_3}^2, a_{32}^2 y_{p_3}^2)^T \end{array}$$

in which the positive coefficients a_{ij}^k , $i = 1, 2, 3$; $j, k = 1, 2$ are randomly generated. With $\epsilon = 0.01$, $\rho = 0.5$ and $\delta = 0.0001$, we carried out about 100 tests and obtained a strong vector equilibrium only in less than half of them.

Conclusion

In this thesis we studied several concepts of equilibrium in multi-criteria traffic networks. For us solving a multi-criteria network equilibrium problem means finding the set of all equilibria of the network or a representative part of it. This problem is very hard, and by our knowledge, there exists no numerical method to solve it because multi-criteria models generally do not have necessary properties for convergence of existing numerical methods.

In the thesis, we applied two approaches to this problem. The first approach constructs an optimization problem the solutions of which are equilibria of the network and the second approach proposes an equivalence between the vector variational inequality problem and the vector equilibrium problem under some appropriate assumptions.

The main results of this thesis are presented in the three Chapters 4, 5 and 6.

In the fourth and fifth chapters, we concentrated our attention on studying the optimization problems based on a vector version of the Heaviside Step function and the distance function to Pareto minimal elements. We proved that the optimal solutions of these problems are exactly the equilibria of the single-product multi-criteria traffic network with and without capacity constraints. Due to the generic differentiability of the objective functions of these optimization problems, we developed an algorithm based on modified Frank-Wolfe gradient method and obtained a representative set of equilibria. A method of smoothing the objective functions by analytic approximations of the Heaviside Step function was also considered in order to see how global optimization may help. We also investigated the robustness of equilibrium and gave a formula to calculate the radius of robustness. Many numerical examples were given to illustrate the above algorithms.

In the sixth chapter we analyzed a complex model of multi-product, multi-criteria networks with capacity constraints, constructed an equivalent variational inequality problem and presented an algorithm to find strong vector equilibria of the model. We also gave sufficient and necessary conditions for strong vector equilibrium in terms of efficient elements of the value set of the criteria function. Major difficulty we face when solving a multi-criteria network equilibrium problem are due to the fact that the conditions for convergence of variational inequality methods are not satisfied, the function determining the associated variational inequality problem is generally not monotone. Nevertheless, the use of a particular class of scalarizing functions in the algorithm, which are proposed in Chapter 2 seems to be quite successful.

In future research we wish to develop an algorithm for multi-product, multi-criteria networks with capacity constraints where convergence conditions can be found. Stability and robustness of equilibrium in the above models will be of our attention. Applications to real models of urban transportation in big cities such as Hanoi and Ho Chi Minh City will be also of interest.

References

1. R. K. Ahuja, T. L. Magnanti and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Englewood Cliffs, NJ, 1993.
2. J-P. Aubin and H. Frankowska, *Set-valued Analysis*, Birkhauser Boston, 1990.
3. M. Beckmann, C. B. McGuire and C. B. Winsten, *Studies in the Economics of Transportation*, Yale University Press (1956).
4. A. Ben-Tal, L. E. Ghaoui and A. Nemirovski, *Robust Optimization*, Princeton University Press, Princeton, 2009.
5. G. Bigi, M. Castellani, M. Pappalardo and M. Passacantando, Existence and solution methods for equilibria, *European Journal of Operational Research*. 227 (2013), pp. 1-11.
6. G. Bigi and B. Panucchi, A successive linear programming algorithm for nonsmooth monotone variational inequalities, *Optimization Methods and Software*, 25 (2010), pp. 29-35.
7. G. Bigi and M. Passacantando, Gap functions and penalization for solving equilibrium problems with nonlinear constraints, *Comput Optim. Appl.* DOI 10.1007/s10589-012-9481-z. 53 (2012), pp. 323-346.
8. G. Chen, On vector network equilibrium problems, *Journal of Systems Science and Systems Engineering*, 14 (2005), pp. 454-461.
9. G. Y. Chen, C. J. Goh and X. Q. Yang, Vector network equilibrium problems and nonlinear scalarization methods, *Mathematical Methods of Operations Research*. 49 (1999), pp. 239-253.
10. G. Y. Chen and N. D. Yen, On the variational inequality model for network equilibrium, Internal Report 3.196 (724), Department of Mathematics, University of Pisa, Pisa (1993).
11. T. C. E. Cheng and Y. N. Wu, A multiclass, multicriteria supply-demand network equilibrium model, *Operations Research*, 53 (2006), pp. 544-554.
12. S. Dafermos, A multicriteria route-mode choice traffic equilibrium model, Lefschetz Center for Dynamical Systems, Brown University, Providence, RI (1981).
13. D. Daniele, A. Maugeri and W. Oettli, Variational inequalities and time-dependent traffic equilibria, *C.R. Acad. Sci. Paris Sér. I. Math.*, 326 (1998), pp. 1059-1062.
14. R. B. Dial, A model and algorithms for multicriteria route-mode choice, *Transportation Research* 13B (1979), pp. 311-316.
15. A. Dontchev and R. T. Rockafellar, *Implicit Functions and Solution Mappings: A View From Variational Analysis*, Springer, New York, 2009.
16. M. Ehrgott, *Multicriteria Optimization*, Originally published as volume 491 in the series: *Lecture Notes in Economics and Mathematical Systems*, 2nd ed. 2005.
17. M. Ehrgott, A discussion of scalarization techniques for multiple objective integer programming, *Ann Oper Res* 147 (2006), pp. 343-360.
18. R. Ferrentino, Variational inequalities and optimization problems, *Applied Mathematical Sciences* 1 (2007), no. 47, pp. 2327-2343.
19. M. Florian, An introduction to network models used in transportation planning, *Transportation Planning Models*, Elsevier Science Publishers (1984), pp. 137-152.

20. M. Gendreau and P. Marcotte (eds.), *Transportation and Network Analysis: Current Trends*, Kluwer Academic Publishers, 2002.
21. F. Giannessi, *Vector Variational Inequalities and Vector Equilibria*, Kluwer Academic Publishers, Dordrecht, Netherlands, 2000.
22. F. Giannessi and A. Maugeri, Variational inequalities and network equilibrium problems, in *Proceedings of the Conference Held in Erice*, Plenum Press, New York, 1994.
23. C. Gert and P. Weidner, Nonconvex separation theorems and some applications in vector optimization, *J. Optim. Theory Appl.* 67 (1990), pp. 297-320.
24. C. J. Goh, and X. Q. Yang, Vector equilibrium problem and vector optimization, *European Journal of Operational Research*, 116 (1999), pp. 615-628.
25. V. V. Gorokhovich, *Convex and Nonsmooth Problems of Vector Optimization*, Nauka i Tekhnika, Minsk, 1990.
26. D. Han and W. Sun, A new modified Gold Stein-Levitin-Dolyak projection method for variational inequality problems, *Computers and Mathematics with Applications*, 47 (2004), pp. 1817-1825.
27. J. B. Hiriart-Urruty, New concepts in non-differentiable programming, *Bull. Soc. Math. France*, 60 (1979), pp. 57-85.
28. I. V. Konnov, Vector network equilibrium problems with elastic demands, *J. Global Optim.* DOI 10.1007/s 10898-011-9798-7.
29. F. Leurent, *Multicriteria assignment modeling: Making explicit the determinants of mode or path choice* (1998).
30. S. Li and G. Chen, On relations between multiclass, multicriteria traffic network equilibrium models and vector variational inequalities, *J Syst Sci Eng.* 15 (2006), pp. 284-297.
31. S. J. Li, X. Q. Yang and G. Y. Chen, A note on vector network equilibrium principles, *Math. Meth. Oper. Res.* 64 (2006), pp. 327-334.
32. S. J. Li, K. L. Teo, and X. Q. Yang, Vector equilibrium problems with elastic demands and capacity constraints, *Journal of Global Optimization*, 37 (2007), pp. 647-660.
33. S. J. Li, K. L. Teo, and X. Q. Yang, A remark on a standard and linear vector network equilibrium problems with capacity constraints, *European Journal of Operational Research* 184 (2008), pp. 13-23.
34. C. Li and G. Chen, Multiclass, two-criteria traffic network equilibrium models and vector variational inequalities, *Systems Engineering-Theory & Practice.* 28 (2008), pp. 141-145.
35. Z. Lin, On existence of vector equilibrium flows with capacity constraints of arcs, *Nonlinear Analysis: Theory, Methods and Applications.* 72 (2010), pp. 2076-2079.
36. Z. Lin, The study of traffic equilibrium problems with capacity constraints of arcs, *Nonlinear Analysis: Real World Applications.* 11 (2010), pp. 2280-2284.
37. D. T. Luc, Multi-product supply demand networks with elementary flows, *Acta Mathematica Vietnamica* 36 (2011), no. 2, pp. 299-317.
38. D. T. Luc, M. Rocca and M. Papalia, Equilibrium in a vector supply demand network with capacity constraints, *Applicable Analysis* 90 (2011), no. 6, pp. 1029-1045.
39. D. T. Luc, *Theory of Vector Optimisation*, Lecture Notes in Economics and Mathematical Systems 319 (1989), Springer-Verlag, Berlin, New York.
40. P. Marcotte and S. Nguyen (Eds.), *Equilibrium and advanced transportation modelling*, Kluwer Academic Publishers, Bostons, MA, pp. 153-174.
41. G. Mastroeni, Gap functions and descent methods for Minty variational inequality, *Optimization and Control with Applications*, 96 (2005), Appl. Optim., Springer, New York, pp. 529-547.
42. V. S. Michael and T. Paul, Modified projection-type methods for monotone variational inequalities, *SIAM Journal on Control and Optimization.* 34 (1998), no. 5, pp. 1814-1830.
43. A. Nagurney, *Network Economics, A Variational Inequality Approach*, Kluwer Academic, Dordrecht (1999).
44. A. Nagurney, A multiclass, multicriteria traffic network equilibrium model, *Mathematical and Computer Modelling*, 32 (2000), pp. 393-411.

45. A. Nagurney, J. Dong, A multiclass, multicriteria traffic network equilibrium model with elastic demand, *Transportation Research Part B*, 36 (2002), pp. 445–469.
46. M. A. Noor, A modified projection method for monotone variational inequalities, *Applied Mathematics Letters* (1999), pp. 83–87.
47. W. Oettli, Necessary and sufficient conditions of Wardrop type for vectorial traffic equilibria, in: GIANNESSI F., MAUGERI A. and PARDALOS P. (eds), *Equilibrium Problems: Nonsmooth Optimization and Variational Inequality Models*, *Nonconvex Optimization and Applications*, 58 (2001), pp. 223–228.
48. A. Pascoletti and P. Serafini, Scalarizing vector optimisation problems, *J. Optim. Theory Appl.* 42 (1984), pp. 499–524.
49. A. Pigou, *The Economics of Welfare*, McMillan (1920).
50. R. E. Quandt, A probabilistic abstract mode model, in *Studies in Travel Demand VIII* (1967), pp. 127–149, *Mathematica, Inc.*, Princeton, NJ.
51. F. Raciti, Equilibrium Conditions and Vector Variational Inequalities: A Complex Relation, *J Glob Optim.* 40 (2008), pp. 353–360.
52. A. Raith and M. Ehrgott, On vector equilibria, vector optimization and vector variational inequalities, *Journal of Multi-Criteria Decision Analysis* 18 (2011), pp. 39–54.
53. Y. Sheffi, *Urban Transportation Networks*, Prentice-Hall, Englewood, NJ, 1985.
54. M. Schneider, Access and land development, *Urban Development Models*. Highway Research Board Special Report 97 (1968), pp. 164–177.
55. M. V. Solodov and P. Tseng, Modified projection-type method for monotone variational inequalities, *SIAM J. Control and Optimization* 34 (1996), no. 5, pp. 1814–1830.
56. M. V. Solodov and B. F. Svaiter, A new projection method for variational inequality problems, *SIAM J. Control Optim.*, 37 (1999), pp. 765–776.
57. Z. Tan, H. Yang and R. Guo, Pareto efficiency of reliability-based traffic equilibria and risk-taking behavior of travelers, *Transportation Research Part B: Methodological*, In Press, Corrected Proof (2014).
58. X. Q. Tian and Y. D. Xu, Traffic network equilibrium problems with capacity constraints of arcs and linear scalarization methods, *Journal of Applied Mathematics*, DOI 10.1155/2012/612142 (2012).
59. T. T. P. Truong and D. T. Luc, Equilibrium in multi-criteria supply and demand networks with capacity constraints, *Mathematical Methods of Operations Research*, DOI 10.1007/s00186-014-0487-4 (2014).
60. Y. D. Xu, S. J. Li and K. L. Teo, Vector network equilibrium problems with capacity constraints of arcs, *Transportation Research Part E*. 48 (2012), pp. 567–577.
61. X. Q. Yang and C. J. Goh, On vector variational inequalities: application to vector equilibria, *Journal of Optimization Theory and Applications* 95 (1997), no. 2, pp. 431–443.
62. M. Wand and D. P. Ertsekas, *Incremental Constraint Projection Methods for Variational Inequalities*, Massachusetts Institute of Technology, Cambridge, MA Laboratory for Information and Decision Systems, Report LIDS-P-2898 (2012).
63. J. Y. T. Wang and M. Ehrgott, Modelling route choice behaviour in a tolled road network with a time surplus maximisation bi-objective user equilibrium model, *Procedia, Social and Behavioral Sciences*. 80 (2013), pp. 266–288.
64. J. G. Wardrop and J. I. Whitehead, Correspondence. Some theoretical aspects of road traffic research, *ICE Proceedings: Engineering Divisions* 1 (1952).
65. A. P. Wierzbicki, A mathematical basis for satisficing decision making, *Mathematical Modelling*, 3 (1982), pp. 391–405.
66. Y. Wu and T. C. E. Cheng, Benson efficiency of a multi-criterion network equilibrium model, *Pacific Journal of Optimization* 5 (2009), pp. 443–458.

Appendices

Appendix A. Description of the algorithm (A)-Section 4.3

A1. For the model in which the cost functions are linear.

Main Program.

```

q = input ('A positive integer number q = ');
epsilon = input ('A tolerance level epsilon = ');
m = input ('Number of paths m = ');
d_w = input ('Demand value d_w = ');
delta = d_w/(q * m);
Y0 = Matrix K2(q)*delta (If we consider two paths in the model)
Y0 = Matrix K3(q)*delta (If we consider three paths in the model)
Y0 = Matrix K6(q)*delta (If we consider six paths in the model)
Y0 = Matrix K7(q)*delta (If we consider seven paths in the model)
s=size(Y0, 2)
n = input ('Number of loops n= ');
l = input ('Number of criteria l = ');
C = input ('Cost coefficient matrix C = ')
Solution = zeros(0);
tic;
for j=1:s
y0 = Y0(1 : m, j);
[T i] = FindSolution(m, d_w, delta, y0, C, l, epsilon, n)
if size(T)== [0, 0]
P = [ones(m, 1); i];
Solution=[Solution P];
else
Solution=[Solution [T;i]];
end
end
Solution
Final.Solution = zeros(0);
for ii=1:s
B= Solution(1:m,ii)- ones(m,1);
if Matrix O(B) == 0
Final.Solution = [Final.Solution Solution(1:m+1,ii)];
end
end
SO = Final.Solution

```

```
wtime =toc;
fprintf(1, 'Elapsed CPU time = f\n', wtime)
```

The following code lines are to simulate the set of vector equilibrium in the two-dimension space.

```
for i=1:size(SO,2)
plot(SO(1,i),SO(2,i),'b.')
hold on
end
grid on
xlabel('Number of products on the first path')
ylabel('Number of products on the second path')
title('Optimization Method ')
hold on
```

The following code lines are to simulate the set of vector equilibrium in the three-dimension space.

```
for i=1:size(SO,2)
plot3(SO(1,i),SO(2,i),SO(3,i),'b.')
hold on
end
grid on
xlabel('Number of products on the first path')
ylabel('Number of products on the second path')
zlabel('Number of products on the third path')
title('Optimization Method ')
hold on
```

```
Triangle3D(d_w);
hold on
```

1. Subprogram

```
function K= Matrix K2(q)
k1= 0:1:2*q;
k2= 0:1:2*q;
S=zeros(0);
for i1=1:2*q+1
for i2=1:2*q+1
if (k1(1,i1)+ k2(1,i2))==2*q)
S=[S;k1(1,i1) k2(1,i2)];
end
end
end
K = S';
end
```

2. Subprogram

```
function K= Matrix K3(q)
k1= 0:1:3*q;
k2= 0:1:3*q;
k3= 0:1:3*q;
S=zeros(0);
for i1=1:3*q+1
for i2=1:3*q+1
for i3=1:3*q+1
if (k1(1,i1)+ k2(1,i2) + k3(1,i3))==3*q)
end
end
end
end
```

```

S=[S;k1(1,i1) k2(1,i2) k3(1,i3)];
end
end
end
end
K=S.';
end

```

3. Subprogram

```

function K= Matrix K6(q)
k1= 0:1:6*q;
k2= 0:1:6*q;
k3= 0:1:6*q;
k4= 0:1:6*q;
k5= 0:1:6*q;
k6= 0:1:6*q;
S=zeros(0);
for i1=1:6*q+1
for i2=1:6*q+1
for i3=1:6*q+1
for i4=1:6*q+1
for i5=1:6*q+1
for i6=1:6*q+1
if (k1(1,i1)+ k2(1,i2) + k3(1,i3)+ k4(1,i4)+ k5(1,i5)+ k6(1,i6))==6*q)
S=[S;k1(1,i1) k2(1,i2) k3(1,i3) k4(1,i4) k5(1,i5) k6(1,i6)];
end
end
end
end
end
end
end
end
K=S.'
end

```

4. Subprogram

```

function K= Matrix K7(q)
k1 = 0 : 1 : 7 * q;
k2 = 0 : 1 : 7 * q;
k3 = 0 : 1 : 7 * q;
k4 = 0 : 1 : 7 * q;
k5 = 0 : 1 : 7 * q;
k6 = 0 : 1 : 7 * q;
k7 = 0 : 1 : 7 * q;
S=zeros(0);
for i1=1:7*q+1
for i2=1:7*q+1
for i3=1:7*q+1
for i4=1:7*q+1
for i5=1:7*q+1
for i6=1:7*q+1
for i7=1:7*q+1

```



```

end
else
 $y^k = (A(1:m,i-1) + A(1:m,i))/2$ 
disp ('Return to Step 2 with received  $y^k$ ')
 $y^0 = y^k$ 
end
i
end
Solution;
end

```

6. Subprogram

```

function V=Comparetwovectors (y,delta,m)
V=zeros(0);
for i=1:m
if (y(i,1) - delta) < 0
V=[V;y(i,1) - delta];
else
V=[V;0];
end
end
end

```

7. Subprogram

```

function Same= Samecoefficient (m,a)
Same = zeros(0);
for i=1:m
Same=[Same;a];
end
end

```

8. Subprogram

```

function Cost = Transfer.cost.matrix (CY,k,m)
Cost=zeros(0);
for i = 1 : m
Cost=[Cost CY((i-1)*k+1:i*k,1)] ;
end
end

```

9. Subprogram

```

function Compute.si=Compute.function.si (CY, Y)
[k, m] = size(CY);
Compute.si=0;
for j=1:m
J =  $J_j(CY,j)$ ;
Compute.si=Compute.si+ Y(j,1)* Compute.partial.function.si(CY, J, j);
end
end

```

10. Subprogram

```

function Compute.partial.sum= Compute.partial.function.si (CY, Jj, j)
l=size(CY,1);

```



```

n=size(Jj,1);
Compute.partial.sum=0;
for s=1:n
i=Jj(s,1);
Compute.partial.sum = Compute.partial.sum + sum(CY(1:1,j)-CY(1:1,i));
end
end

```

11. Subprogram

```

function Jj= Jj (CY,j)
Jj=zeros (0);
[k,m]=size (CY);
for i=1:m
Subtraction.two.matrix = CY(1:k,j)-CY(1:k,i);
if Matrixnonnegative(Subtraction.two.matrix) == 1;
Jj = [Jj; i];
end
end
Jj=Jj;
end

```

12. Subprogram

```

function n= Matrixnonnegative (A)
sizeA = size(A,1);
B=zeros (0);
for i = 1: size (A)
if A(i,1) <= 0
B = [B;1];
end
end
if size (B,1) == size(A)
n=1;
else n=0;
end
end

```

13. Subprogram

```

function d = Gradient.si(C,Y,k,m)
Cost=C*Y;
CY= Transfer.cost.matrix(Cost, k, m);
g1=zeros(0);
for i=1:m
J1 = Jj(CY, i);
g1=[g1; Compute.partial.function.si(CY, J1, i)];
end
g1
g2=0;
g3=zeros(0);
for t=1:m
g2 = The.second.part.si(C, Y, m, k, t);
g3 = [g3; g2];
end

```

```

d = g1 + g3;
end

```

14. Subprogram

```

function G= Compute.partial.gradient.si (C,Y,i,k,t)
m=size(C,2);
Cost=C*Y;
CY=Transfer.cost.matrix(Cost,k,m);
Jj=Jj(CY,i);
n=size(Jj,1);
G=0;
for s=1:n
j=Jj(s,1);
G = G + sum(C((i-1)*k+1 :i*k,t)- C((j-1)*k+1 :j*k,t));
end
end

```

15. Subprogram

```

function g3= The.second.part.si(C,Y,m,k,t)
Cost=C*Y;
CY= Transfer.cost.matrix(Cost,k,m);
g2=0;
g3=zeros(0);
for i=1:m
J2 = Jj(CY,i);
g2 = g2 + Y(i,1)*Compute.partial.gradient.si(C,Y,i,k,t);
end
g3 = [g3; g2];
end

```

16. Subprogram

```

function Compute.siu=Compute.function.siu(CY,Cu,u)
[l,m] = size(Cu);
Compute.siu=0;
for j=1:m
J = Jj(CY,j);
Compute.siu=Compute.siu+ u(j,1)* Compute.partial.function.si(Cu,J,j);
end
end

```

17. Subprogram

```

function n=Matrix.O (A)
k=size (A,1);
B=zeros (0);
for i = 1: k
if A(i,1) == 0
B = [B;1];
end
end
if size (B,1) == k
n=1;
else n=0;
end

```

```
end
end
```

A2. For the model in which the cost functions are not linear.

Main program

```
epsilon = input ('A tolerance level epsilon= ');
m = input ('Number of paths m = ');
q = input ('q = ');
d_w = input ('Demand value d_w = ');
delta = d_w/(q*m);
Y0 = Matrix K2(q)*delta (If we consider two paths in the model)
Y0 = Matrix K3(q)*delta (If we consider three paths in the model)
Y0 = Matrix K6(q)*delta (If we consider six paths in the model)
Y0 = Matrix K7(q)*delta (If we consider seven paths in the model)
s=size(Y0,2)
n = input ('Number of loops n= ');
l = input ('Number of criteria l = ');
Solution = zeros(0);
tic;
for j=1:s
y0 = Y0(1:m,j);
T = FindSolution.nl(m, d_w, delta, y0, l, epsilon, n);
if size(T)==[0,0]
P=ones(m,1);
Solution=[Solution P];
else
Solution=[Solution T]
end
end
Final.Solution = zeros(0);
for i=1:s
B= Solution(1:m,i)- ones(m,1);
if Matrix.O(B) == 0
Final.Solution =[Final.Solution Solution(1:m,i)];
end
end
SO = Final.Solution
wtime =toc;
fprintf(1, 'Elapsed CPU time = f\ n', wtime)
```

1. Subprogram

```
function Solution =FindSolution.nl(m, d_w, delta, y0, l, epsilon, n)
Aeq = ones(1,m);
beq= d_w;
lb=Comparetwovectors(y0, delta, m);
ub= Samecoefficient(m, delta) + y0;
alpha = inf;
Matrix.si = zeros (0);
A= zeros (0);
Solution=zeros(0);
for i=1:n
A = [A y0]
```

```

y=A(1:m,i)
Cost=C6(y)
CY=Transfer.cost.matrix(Cost,l,m)
si=Compute.function.si(CY,A(1:m,i))
Matrix.si=[Matrix.si si]
if si j= epsilon
disp ('The present value is a vector epsilon-equilibrium')
i;
Solution = [Solution;A(1:m,i)];
return
end
if ((si-alpha) j= epsilon) && (-epsilon j= (si-alpha))
return
elseif si j (alpha-epsilon)
alpha = si
Gradient = Gradient.si.nl(A(1:m,i),l,m)
u = linprog(Gradient,[],[],Aeq,beq,lb,ub)
ProductCu= C6(u);
Cu= Transfer.cost.matrix(ProductCu,l,m);
siu=Compute.function.siuk(CY,Cu,u)
while ((si-siu) j= epsilon) && (-epsilon j= (si-siu))
break
end
while ((si-siu) j epsilon) —— (-epsilon j (si-siu))
disp ('Return to Step 2 with received  $u^k$  ')
 $y^0 = u^k$ ;
break
i;
end
else
 $y^k = (A(1:m,i-1) + A(1:m,i))/2$ 
disp ('Return to Step 2 with received  $y^k$ ')
 $y^0 = y^k$  ;
i;
end
i
end
Solution;
end

```

2. Subprogram function d = Gradient.si.nl(Y,l,m)

```

Cost = C6(Y)
CY= Transfer.cost.matrix(Cost,l,m);
g1=zeros(0);
for i=1:m
J1 = Jj(CY,i);
g1=[g1; Compute.partial.function.si(CY,J1,i)];
end
g1
g2 = 0;
g3=zeros(0);
for t=1:m

```

```

g2=The.second.part.si.nl(Y,m,l,t);
g3 = [g3; g2]
end
d = g1 + g3;
end

```

3. Subprogram

```

function g4= The.second.part.si.nl(Y,m,l,t)
g2=0;
g4=zeros(0);
for i=1:m
g2 = g2+ Y(i,1)*Compute.partial.gradient.si.nl(Y,i,l,m,t);
end
g4 = [g4; g2];
end

```

Appendix B. Smoothing the objective function (Subsection 4.3.4)

Main Program.

```

q = input ('A positive integer number q= ');
m = input ('Number of paths m = ');
dw = input ('Demand value dw = ');
l = input ('Number of criteria l = ');
delta = dw/(q*m);
Y0= Matrix K2(q)*delta (apply for in Example 4.3.5)
Y0= Matrix K3(q)*delta (apply for in Example 4.3.6)
Y0= Matrix K6(q)*delta (apply for in Example 4.3.10)
s=size(Y0,2)
Aeq = ones(1,m)
beq= dw;
lb=zeros(m,1);
ub=Samecoefficient(m, +Inf);
opts = optimoptions(@fmincon,'Algorithm','interior-point')
Solution = zeros(0);
tic;
for j=1:s
y0 = Y0(1:m,j);
problem= createOptimProblem('fmincon','x0',y0, 'objective',@Compute.function.si.nu,
'Aeq',Aeq,'beq',beq,'lb',lb,'ub',ub,'options',opts);
[y fval] = fmincon (problem)
z=[y;fval];
Solution=[Solution z];
end
Solution
wtime =toc;
fprintf(1, 'Elapsed CPU time = f \ n, wtime)

```

1. Subproblem

```

function Same= Samecoefficient (m,a)
Same = zeros(0);
for i=1:m
Same=[Same;a];
end

```

```
end
end
```

2. Subproblem

```
function Compute.si.nu=Compute.function.si.nu (y)
m=size(y,1);
l=2;
nu=1, ..., 10;
Cost = [3 * y(1, 1) + y(2, 1); 5 * y(1, 1) + 3 * y(2, 1); y(1, 1) + y(2, 1); 3 * y(1, 1) + 5 * y(2, 1)]
(the cost function in Example 3.3.4)
Cost= [y(1, 1)2 + y(2, 1)2 + y(3, 1)2; 2 * y(1, 1) + 5 * y(2, 1) + 3 * y(3, 1); 8 * y(1, 1) * y(2, 1) +
y(2, 1)2; y(2, 1) + 10 * y(3, 1); y(1, 1) + y(2, 1)2 + y(3, 1)2; 10 * y(3, 1)3] (the cost function in
Example 3.3.5)
Cost= [2 * y(1, 1)2 + 7 * y(2, 1)2; 2 * y(1, 1) + 5 * y(2, 1) + 3 * y(3, 1); 8 * y(1, 1) * y(2, 1) +
y(2, 1)2; 3 * y(2, 1) + 10 * y(3, 1)2; y(1, 1) + y(2, 1)2 + y(3, 1)2; y(3, 1) + 10 * y(3, 1)3;
y(1, 1)3 + y(4, 1) * y(5, 1); 2 * y(2, 1) + y(4, 1)2; y(2, 1) + y(5, 1) + y(6, 1); y(1, 1) * y(5, 1) +
y(3, 1) * y(6, 1) + 10; y(3, 1) + y(6, 1); y(6, 1)2 + y(1, 1)] (the cost function in Example 3.3.8)
CY=Transfer.cost.matrix(Cost, l, m);
[k, m] = size (CY);
Compute.si.nu=0;
for j=1:m
Compute.si.nu=Compute.si.nu+ y(j,1)* Partial.sum.function.si.nu(CY, j, m, nu);
end
end
```

3. Subproblem

```
function Partial.sum= Partial.sum.function.si.nu (CY, j, m, nu)
l=size(CY,1);
Partial.sum=0;
for i=1:m
a = H.tilde.nu (CY, i, j, nu);
Partial.sum = Partial.sum + (CY(1:l,j)-CY(1:l,i))l*a;
end
end
```

4. Subproblem

```
function Htildenu= H.tilde.nu (CY, i, j, nu)
l=size(CY,1);
a=1;
for k=1:l
a = a * (1+tanh(nu*(CY(k,j)-CY(k,i))))/2;
end
Htildenu= Samecoefficient(l, a);
end
```

Appendix C. Description of the algorithm (Section 4.4)

Main Program.

```
t= size(SO,2); (SO is the matrix of vector equilibria)
S-Ra=zeros(0);
for i=1:t
Yi=SO(:,i);
D=Transfer.cost.matrix(C * Yi, l, m);
```

```

Cw=D';
Rho= Rhofunction(Yi,Cw,l);
if Rho == 0
Ra= RadiusRobustness(Cw,l)
S-Ra=[S-Ra; Yi' Ra]
end
end
P= S-Ra'

```

1. Subproblem

```

function Rho= Rhofunction (Y,Cw,l)
m=size(Y,1);
Rho=0;
for i=1:m
Rho= Rho + Y(i,1)* Khifunction(Cw(i,:),Cw,i,l);
end
end

```

2. Subproblem

```

function SumKhi= Khifunction (c,Cw,k,l)
[I1 MinA]= MinMatrix(Cw,l);
n1=size(I1,1);
SumKhi=0;
for j=1:n1
i= I1(j,1);
n0= norm (c - Cw(i,:));
if (n0==0) && (i = k)
SumKhi=SumKhi+1;
else
SumKhi=SumKhi+0;
end
end
end
end

```

3. Subproblem

```

function [I1 MinA]= MinMatrix(A,l)
r1=size(A,1);
MinA=zeros(0);
B1=zeros(0);
I1=zeros(0);
for i=1:r1
for j=1:r1
B=A(j,1:l)-A(i,1:l);
B1=[B1;B];
end
end
s1=size(B1,1)/r1;
for i=1:s1
B2=B1((i-1)*r1+1:i*r1,1:l);
s2=size(B2,1);
t=TextMin(B2,l);
if t==s2

```

```

MinA=[MinA;A(i,1:1)];
I1=[I1;i];
else MinA=[MinA;zeros(0)];
end
end
end

```

4. Subproblem

```

function a = TextMin(D,l)
r1=size(D,1);
a=0;
s1=r1;
for i=1:s1
D1=D(i,1:1);
if DauMTTMin(D1)==1—D1==zeros(1,1)
a=a+1;
end
end
end

```

5. Subproblem

```

function SignMT = SignMTTMin(D1)
[r1,r2]=size(D1);
E1=D1;zeros(r1,r2);
E2=D1;zeros(r1,r2);
if (SumMatrix(E1)i=1 & SumMatrix(E2)i=1)—SumMatrix(E2)i=1
SignMT = 1;
else SignMT = 0;
end
end

```

6. Subproblem

```

function summatrix = SumMatrix( A )
summatrix = 0;
for i = 1:size(A,1)
for j = 1:size(A,2)
summatrix = summatrix + A(i,j);
end
end
end

```

7. Subproblem

```

function Ra= RadiusRobustness (Cw,l)
m= size(Cw,1);
[Iw MinA]= MinMatrix(Cw,l);
n= size(Iw,1);
A=zeros(0);
for j=1:n
i= Iw(j,1);
for ii=1:m
if ii = i
a= NormPositiveMatrix(Cw(ii,:) - Cw(i,:));

```



```

A=[A a];
end
end
end
Ra=sqrt(1)/2* min(A);
end

```

8. Subproblem

```

function PE= NormPositiveMatrix (C)
l= size(C,2);
Cplus = zeros(0);
for i=1:l
if C(1,i) < 0
Cplus = [Cplus C(1,i)];
end
end
PE=norm (Cplus);
end

```

Appendix D. Description of the algorithm (Chapter 5)

Main Program.

```

q = input ('A positive integer number q= ');
epsilon = input ('A tolerance level epsilon= ');
m = input ('Number of paths m = ');
d_w = input ('Demand value d_w = ');
n = input ('Number of loops n= ');
l = input ('Number of criteria l = ');
C = input ('Cost coefficient matrix C = ');
lb = input ('Lower bound matrix lb = ');
ub = input ('Upper bound matrix ub = ');
delta = d_w/(q*m);
Y0= Matrix K2(q,lb,ub,d_w,m)*delta
s=size(Y0,2)
Solution = zeros(0);
tic;
for j=1:s
y0 = Y0(1:m,j);
[T i] = FindSolutionConstraint(m, d_w, delta, y0, C, l, epsilon, n, lb, ub)
if size(T)==[0,0]
P=[ones(m,1);i];
Solution=[Solution P];
else
Solution=[Solution [T;i]];
end
end
Final.Solution = zeros(0);
for ii=1:s
B= Solution(1:m,ii)- ones(m,1);
if Matrix.O(B) == 0
Final.Solution =[Final.Solution Solution(1:m+1,ii)];
end
end
end

```

```

SO = Final.Solution
wtime =toc;
fprintf(1, 'Elapsed CPU time = f\ n', wtime)

```

1. Subproblem

```

function K= Matrix K2(q,lb,ub,dw,m)
delta = dw/(q*m)
k1= 0:1:2*q;
k2= 0:1:2*q;
S=zeros(0);
for i1=1:2*q+1
for i2=1:2*q+1
if (k1(1,i1)+ k2(1,i2)==2*q) && (lb(1,1)j=k1(1,i1)*delta)&&(k1(1,i1)*deltaj=ub(1,1)) &&
(lb(2,1)j=k2(1,i2)*delta)&&(k2(1,i2)*deltaj=ub(2,1))
S=[S;k1(1,i1) k2(1,i2)];
end
end
end
K=S.';
end

```

2. Subproblem

```

function [Solution i] = FindSolutionConstraint (m,dw,delta,y0,C,l,epsilon,n,lb,ub)
Aeq = ones(1,m);
beq= dw;
lbb= Comparetwolowerbounds(y0, delta, lb, m);
ubb= Comparetwoupperbounds(y0, delta, ub, m);
alpha = inf;
Matrix.si = zeros (0);
A= zeros (0);
Solution=zeros(0);
for i=1:n
A = [A y0]
y=A(1:m,i)
Cost=C*y;
CY=Transfer.cost.matrix(Cost, l, m);
si=Compute.si.constraint(CY, y, lb, ub)
Matrix.si=[Matrix.si si]
if si j= epsilon
disp ('The present value is a vector epsilon-equilibrium')
Solution = [Solution;A(1:m,i)]
i
return
end
if ((si-alpha) j= epsilon) && (-epsilon j= (si-alpha))
return
elseif si j (alpha-epsilon)
alpha = si
Gradient = Gradient.function.si(A(1 : m, i), lb, ub, C, l)
u = linprog(Gradient,[],[],Aeq,beq,lbb,ubb)
ProductCu= C*u;
Cu= Transfer.cost.matrix(ProductCu, l, m)

```

```

siu=Compute.function.siuk.constraint(CY,Cu,u,lb,ub)
while ((si-siu) >= epsilon) && (-epsilon <= (si-siu))
break
end
while ((si-siu) < epsilon) && (-epsilon < (si-siu))
disp ('Return to Step 2 with received  $u^k$ ')
 $y^0 = u$ 
break
end
else
 $y^k = (A(1:m,i-1) + A(1:m,i))/2$ 
disp ('Return to Step 2 with received  $y^k$ ')
 $y^0 = y^k$ 
end
end
end
end

```

3. Subproblem

```

function V=Comparetwolowerbounds (y,delta,lb,m)
V=zeros(0);
for i=1:m
if (y(i,1) - delta) < lb(i,1)
V=[V;y(i,1) - delta];
else
V=[V;lb(i,1)];
end
end
end
end

```

4. Subproblem

```

function Z=Comparetwoupperbounds (y,delta,ub,m)
Z=zeros(0);
for i=1:m
if (y(i,1) + delta) > ub(i,1)
Z=[Z;y(i,1) + delta];
else
Z=[Z;ub(i,1)];
end
end
end
end

```

5. Subproblem

```

function Compute.si=Compute.si.constraint (CY,Y,lb,ub)
[k,m] = size (CY);
Compute.si=0;
for i=1:m
Ji =  $J_j(CY,i)$ ;
Compute.si=Compute.si+ (Y(i,1)-lb(i,1))* Compute.partial.function.si.constraint
(CY, Ji, i, Y, ub);
end
end

```

6. Subproblem

```

function Compute.partial.sum= Compute.partial.function.si.constraint (CY, Jj, i, y, ub)
l=size(CY,1);
n=size(Jj,1);
Compute.partial.sum=0;
for s=1:n
j=Jj(s,1);
Compute.partial.sum = Compute.partial.sum + (ub(j,1)-y(j,1))*sum(CY(1:l,i)-CY(1:l,j));
end
end

```

7. Subproblem

```

function d = Gradient.function.si(Y,lb,ub,C,l)
g1=zeros(0);
for i=1:size(Y,1)
g1=[g1; derivative.si(Y, lb, ub, C, i, l)];
end
d=g1;
end

```

8. Subproblem

```

function x3= derivative.si(Y,lb,ub,C,k,l)
x4=0;
for i=1:size(Y,1)
x4=x4+ith.part.of.derivative(Y, lb, ub, C, k, i, l)
end
x3=x4 - ith.part.of.derivative(Y, lb, ub, C, k, k, l)+Part2(Y, ub, C, k, l)+ $(Y(k, 1) - lb(k, 1)) * Part1(Y, ub, C, k, k, l)$ 
end

```

9. Subproblem

```

function f= ith.part.of.derivative (Y,lb,ub,C,k,i,l)
m=size(C,2);
Cost=C*Y;
CY=Transfer.cost.matrix(Cost, l, m);
Ji=Jj(CY,i);
n=size(Ji,1);
if belongtoamatrix(Ji, k)==0
f=  $(Y(i,1)-lb(i,1)) * Part1(Y, ub, C, k, i, l)$ 
else
f= $(Y(i,1)-lb(i,1)) * Part1(Y, ub, C, k, i, l) + (Y(i,1)-lb(i,1)) * (-1) * sum(CY(1:l,i) - CY(1:l,k))$ 
end
end

```

10. Subproblem

```

function a = belongtoamatrix(A,k)
B=zeros(0);
for i=1:size(A,1)
if k==A(i,1)
B=[B;1];
end
end
end

```

```

if sum(B)==1
a=1;
else a=0;
end
end

```

11. Subproblem

```

function x1= Part1(Y,ub,C,k,i,l)
m=size(C,2);
Cost=C*Y;
CY=Transfer.cost.matrix(Cost,l,m);
Ji= $J_j$ (CY,i);
n=size(Ji,1);
x1=0;
for s=1:n
j=Ji(s,1);
x1 = x1 + (ub(j,1)-Y(j,1))* sum(C((i-1)*l+1 :i*l,k)- C((j-1)*l+1 :j*l,k));
end

```

12. Subproblem

```

function x2= Part2(Y,ub,C,k,l)
m=size(C,2);
Cost=C*Y;
CY=Transfer.cost.matrix(Cost,l,m);
Jk= $J_j$ (CY,k);
n=size(Jk,1);
x2=0;
for s=1:n
j=Jk(s,1);
x2 = x2 + (ub(j,1)-Y(j,1))*sum(CY(1:l,k)-CY(1:l,j));
end
end

```

Appendix E. Description of the algorithm (Chapter 6)**Main Program.**

```

s = input ('Number of pairs s = ');
l = input ('Number of criteria l = ');
q = input ('Number of products q = ');
M0 = zeros(0);
for i=1:s
mi = input ('Enter mi = ');
M0 = [M0 mi];
end
M=M0;
L = zeros(0);
for i=1:s
Li = input ('Enter Li = ');
L = [L ; Li];
end
L=L;
U = zeros(0);
for i=1:s

```

```

Ui = input ('Enter Ui = ');
U = [U ; Ui];
end
U=U;
UL=U-L;
for i=1 : summ(M,s)
for j = 1:q
if UL(i,j) > 0
error('Verify the coefficients of the matrix L and U')
end
end
end
d = zeros(0);
for i=1:s
dwi = input ('Enter dwi = ');
d = [d ; dwi];
end
for i=1 : s
for j = 1:q
if d(i,j) > min(L(summ(M,i-1)+1:summ(M,i),j))*M(1,i) — d(i,j) < max(U(summ(M,i-1)+1:summ(M,i),j))*M(1,i)
error('Verify the coefficients of the demand matrix d')
end
end
end
epsilon = input ('Epsilon = ');
delta = input ('Delta = ');
gamma = input ('Gamma = ');
tol = input ('Tol = ');
nn=input('nn = ');
t = input('t = ');
Lc = ArrangeLMCtotal(L, M, q, s);
Uc = ArrangeUMCtotal(U, M, q, s);
beq = zeros(0);
for i=1:s
beqi = d(i,1:q)';
beq = [beq ; beqi];
end
beq=beq;
Y01=linprog(zeros(M(1,1)*q,1),[],[],FindAeq(M(1,1),q),Collumdw(M(1,1),q,
beq(1:q,1)),Lc(1:M(1,1)*q,1),Uc(1:M(1,1)*q,1));
Y0i = zeros(0);
for i=2:s
Yi = linprog(zeros(M(1,i)*q,1),[],[],FindAeq(M(1,i),q),Collumdw(M(1,i),q,
beq((i-1)*q+1:i*q,1)),Lc(summ(M,i-1)*q+1:summ(M,i)*q,1),Uc(summ(M,i-1)*q+1:
summ(M,i)*q,1));
Y0i = [Y0i ; Yi];
end
Y0=[Y01;Y0i];
Y00 = ArrangeY0total(Y0, M, q, s)
C = zeros(0);
for i=1:s
Cwi = input ('Cwi = ');

```

```

C = [C ; Cwi];
end
Y = Solutionoptgeneralfdistance(C, nn, delta, rho, s, l, q, L, U, M, epsilon, beq, Y00) Time
= cputime-t
CC=MatrixCYNewtotal(M,q,l,Y,C,s);
for i=1:s
Ci=CC((summ(M, i - 1))*1+1:(summ(M, i))*1,1:q);
Yi=Y(summ(M, i - 1)+1:summ(M, i), 1 : q);
Li=L(summ(M, i - 1)+1:summ(M, i), 1 : q);
Ui=U(summ(M, i - 1)+1:summ(M, i), 1 : q);
CLw=zeros(0);
for ii = 1: M(1,i)
if Yi(ii,1:q)==Li(ii,1:q)
CLw=[CLw;Ci((ii-1)*1+1:ii*1,1:q)];
end
end
CLw;
CUw=zeros(0);
for i1 = 1: M(1,i)
if Yi(i1,1:q)==Ui(i1,1:q)
CUw=[CUw;Ci((i1-1)*1+1:i1*1,1:q)];
end
end
CUw;
CEw=zeros(0);
for i2 = 1: M(1,i)
if (Count(Yi(i2, 1 : q), Li(i2, 1 : q))i=1)&(Count(Ui(i2, 1 : q), Yi(i2, 1 : q))i=1)
CEw=[CEw;Ci((i2-1)*1+1:i2*1,1:q)];
end
end
CEw;
y1=size(CUw,1);
y2=size(CEw,1);
y3=size(CLw,1);
z1=y1/l;
z2=y2/l;
z3=y3/l;
b1=0;
for i3=1:z1
for j=1:z3
if Count(CUw((i3 - 1) * l + 1 : i3 * l, 1 : q), CLw((j - 1) * l + 1 : j * l, 1 : q))i=1
b1=b1+1;
end
end
end
if b1i=1
disp('The first and second conditions are not satisfied')
end
b2=0;
for i4=1:z2
for j=1:z3
if Count(CEw((i4 - 1) * l + 1 : i4 * l, 1 : q), CLw((j - 1) * l + 1 : j * l, 1 : q))i=1

```

```

b2=b2+1;
end
end
end
if b2i=1
disp('The first condition is not satisfied')
else disp('The first condition is satisfied')
end
b3=0;
for i5=1:z1
for j=1:z2
if Count(CUw((i5 - 1) * l + 1 : i5 * l, 1 : q), CEw((j - 1) * l + 1 : j * l, 1 : q))i=1
b3=b3+1;
end
end
end
if b3i=1
disp('The second condition is not satisfied')
else disp('The second condition is satisfied')
end
if size(CEw,1) = 0
A2=MaxMatrixA(CEw, l, q);
A1=MinMatrixA(CEw, l, q);
if (size(CEw)==size(A1))—(size(CEw)==size(A2))
disp('The third condition is satisfied')
else
disp('The third condition is not satisfied')
end
disp('The next O/D pair')
else
disp('The third condition is satisfied')
disp('The next O/D pair')
end
end
end

```

1. Subproblem

```

function LchangeMCtotal = ArrangeLMCtotal(L,M,q,s)
L1=ArrangeMCL(L(1 : M(1, 1), 1 : q), M(1, 1), q);
LL=zeros(0);
for i=2:s
Larrange=[LL;ArrangeMCL(L(summ(M,i-1)+1:summ(M,i),1:q),M(1,i),q)];
LL=Larrange;
end
LchangeMCtotal=[L1;LL];
end

```

2. Subproblem

```

function LchangeMC = ArrangeMCL(L,m,q)
LchangeMC=zeros(0);
for i=1:q
LchangeMC=[LchangeMC;L(1:m,i)];
end

```


end

3. Subproblem

```
function UchangeMCtotal = ArrangeUMCtotal(U,M,q,s)
U1=ArrangeMCU(U(1 : M(1,1), 1 : q), M(1, 1), q);
UU=zeros(0);
for i=2:s
Uarrange=[UU;ArrangeMCU(U(summ(M, i - 1) + 1 : summ(M, i), 1 : q), M(1, i), q)];
UU=Uarrange;
end
UchangeMCtotal=[U1;UU];
end
```

4. Subproblem

```
function UchangeMC = ArrangeMCU(U,m,q)
UchangeMC=zeros(0);
for i=1:q
UchangeMC=[UchangeMC;U(1:m,i)];
end
end
```

5. Subproblem

```
function Aeq=FindAeq(m,q)
Aeq0 = zeros(1);
for i=1:m
Aeq0 = [Aeq0 1];
end
Aeq0(:,1)=[];
Aeq=zeros(m*q);
for j=1:q
Aeq(j,(j-1)*m+1:j*m)=Aeq0;
end
end
```

6. Subproblem

```
function dwi = Collumdw(m,q,beq)
dwi=[beq;zeros(m*q-q,1)];
end
```

7. Subproblem

```
function Yn = FindSolutionoptfdistance(m,s,nn,delta,rho,q,l,L,U,epsilon,C,
Aeq,beq,Y,M,Ytotal)
Loop=0;
B=Solutionoptfdistance(m, s, rho, q, l, L, U, epsilon, C, Aeq, beq, Y, M, Ytotal);
for i=1:nn
diff=norm(B(m*q+1:2*m*q,1)-B(1:m*q,1),2);
if (diff <= delta)
B=Solutionoptfdistance(m, s, rho, q, l, L, U, epsilon, C, Aeq, beq, ArrangeB(B(m * q + 1 :
2 * m * q, 1), m, q), M, ArrangeB(B(m * q + 1 : 2 * m * q, 1), m, q));
Y1=B(m*q+1:2*m*q,1);
Loop=Loop+1;
a=Loop
```

```

else
Y1=B(m*q+1:2*m*q,1);
a=Loop;
break
end
end
Y2=Y1;
Z1=zeros(0);
for k=1:q
Z1=[Z1 Y2((k-1)*m+1:k*m,1)];
end
Tolerance=diff
Yn=Z1;
The numbers of loops=a
end

```

8. Subproblem

```

function B = Solutionoptfdistance(m,s,rho,q,l,L,U,epsilon,C,Aeq,beq,Y,M,Ytotal)
C1=MatrixCY(m,q,l,Ytotal,C,M,s);
B=ArrangeMCL(Y,m,q);
Calpha = zeros(0);
for i = 1:m
if Count(Y(i,1:q),L(i,1:q)) $\leq$ 1
Calpha = [Calpha ; C1(i*(l-1)+1:i*(l-1)+1:q)];
end
end
n=size(Calpha,1)/l;
F=zeros(0);
for j=1:m
fd=zeros(0);
for ii = 1:n
D=C1((j-1)*l+1:j*l,1:q)- Calpha((ii-1)*l+1:ii*l,1:q);
if (D  $\leq$  0) == zeros(1,q)
fCalphaCj = -distance(D) + epsilon*SumMatrix(D);
else
fCalphaCj = distancepo(D) + epsilon* SumMatrix(D);
end
fd=[fd;fCalphaCj];
end
fd1=min(fd);
fd2=zeros(0);
for i = 1:q
fd2=[fd2;fd1];
end
F=[F;fd2];
end
Fchange=[];
for i=1:m
Fchange=[Fchange;F((i-1)*q+1:i*q,1)'];
end
Frechange=ArrangeMCL(Fchange,m,q);
Y1=ArrangeMCL(Y,m,q);

```

```

S1= Y1 - rho*Frechange;
lb=ArrangeMCL(L, m, q);
ub=ArrangeMCL(U, m, q);
S2=lsqlin(eye(m*q),S1,[],[],Aeq,beq,lb,ub);
Yn=(1-rho)*Y1+rho*S2;
B=[B;Yn]; end

```

9. Subproblem

```

function CY=MatrixCY(m,q,l,Y,C,M,s)
CY=zeros(m*q);
for i=1:m*q
for j=1:q
CY(i,j)=sum(sum( C((i-1)*summ(M,s)+1 : i*summ(M,s),(j-1)*q+1:j*q).*Y ));
end
end
end

```

10. Subproblem

```

function a = Count(Y,L)
s=(Y-L);
b=(s<0);
[s1 s2]=size(s);
Count=zeros(0);
for i=1:s2
if b(1,i)==1
Count=[Count,1];
end
end
a=size(Count,2);
end

```

11. Subproblem

```

function diss = distancene(x)
[s1, s2]=size(x);
c=zeros(0);
for i=1:s1
c=[c;x(i,1:s2)];
end
d=c;
diss=norm(d,-Inf);
end

```

12. Subproblem

```

function dis = distancepo(x)
a=zeros(0);
for i=1:size(x,1)
for k=1:size(x,2)
if x(i,k)~0
a=[a;x(i,k)];
end
end
end
end

```

```

b=a;
sum=0;
for j=1:size(b,1)
for l=1:size(b,2)
sum=sum+b(j,l).^2;
end
end
dis=sqrt(sum);

```

13. Subproblem

```

function Bchange = ArrangeB(B,m,q)
Bchange=zeros(0);
for i=1:q
Bchange=[Bchange B((i-1)*m+1:i*m,1)];
end
end

```

14. Subproblem

```

function sum = summ(M,s)
sum=0;
for i=1:s
sum=sum+M(1,i);
end
end

```

15. Subproblem

```

function summatrix = SumMatrix( A )
summatrix = 0;
for i = 1:size(A,1)
for j = 1:size(A,2)
summatrix = summatrix + A(i,j);
end
end
end

```

16. Subproblem

```

function Y0total = ArrangeY0total(Y0,M,q,s )
Y1=ArrangeY0(Y0(1:M(1,1)*q,1),M(1,1),q);
YY=zeros(1,q);
for i=2:s
Yarrange=[YY;ArrangeY0(Y0(summ(M,i-1)*q+1:summ(M,i)*q,1),M(1,i),q)];
YY=Yarrange;
end
YY(1,:)=[];
Y0total=[Y1;YY];
end

```

17. Subproblem

```

function Ychange = ArrangeY0(Y0,m,q)
Ychange=zeros(m,1);
for i=0:(q-1)
Ychange=[Ychange Y0(m*i+1:m*(i+1),1)];
end

```

```

end
Ychange(:,1)=[];
end

```

18. Subproblem

```

function Ys = Soluptionoptgeneralfdistance(C,nm,delta,rho,s,l,q,L,U,M,epsilon,beq,Y)
Z1=zeros(0);
for k=1:s
beqk=[beq((k-1)*q+1:k*q,1);zeros(M(1,k)*q-q,1)];
Ck=C(summ(M,k-1)*1*summ(M,s)+1:summ(M,k)*1*summ(M,s),1:q*q);
Yk=Y(summ(M,k-1)+1:summ(M,k),1:q);
Lk=L(summ(M,k-1)+1:summ(M,k),1:q);
Uk=U(summ(M,k-1)+1:summ(M,k),1:q);
Y0k=FindSolutionoptfdistance(M(1,k),s,nm,delta,rho,q,l,Lk,Uk,epsilon,
Ck,FindAeq(M(1,k),q),beqk,Yk,M,Y);
Z1=[Z1;Y0k];
end
Ys=Z1;
end

```

19. Subproblem

```

function CY=MatrixCYtotal(M,q,l,Y,C,s)
CY1=MatrixCY(M(1,1),q,l,Y,C(1:M(1,1)*l*summ(M,s),1:q*q),M,s);
CY=zeros(0);
for i=2:s
Ci=C(summ(M,i-1)*l*summ(M,s)+1:summ(M,i)*l*summ(M,s),1:q*q);
CYi=MatrixCY(M(1,i),q,l,Y,Ci,M,s);
CY=[CY;CYi];
end
CY=[CY1;CY];
end

```

20. Subproblem

```

function MaxA= MaxMatrixA(A,l,q)
[r1,r2]=size(A);
s1=r1/l;
MaxA=zeros(0);
B1=zeros(0);
for i=1:s1
for j=1:s1
B=A((j-1)*l+1:j*l,1:q)-A((i-1)*l+1:i*l,1:q);
B1=[B1;B];
end
end
s2=size(B1,1)/r1;
for i=1:s2
B2=B1((i-1)*r1+1:i*r1,1:q);
s4=size(B2,1)/l;
t=TextDMax(B2,q,l);
if t==s4
MaxA=[MaxA;A((i-1)*l+1:i*l,1:q)];
else MaxA=[MaxA;zeros(0)];
end
end

```

```
end
end
end
```

21. Subproblem

```
function MinA= MinMatrixA(A,l,q)
[r1,r2]=size(A);
s1=r1/l;
MinA=zeros(0);
B1=zeros(0);
for i=1:s1
for j=1:s1
B=A((j-1)*l+1:j*l,1:q)-A((i-1)*l+1:i*l,1:q);
B1=[B1;B];
end
end
B1;
s2=size(B1,1)/r1;
for i=1:s2
B2=B1((i-1)*r1+1:i*r1,1:q);
s4=size(B2,1)/l;
t=TextDMin(B2,q,l);
if t==s4
MinA=[MinA;A((i-1)*l+1:i*l,1:q)];
else MinA=[MinA;zeros(0)];
end
end
end
```

22. Subproblem

```
function a = TextDMax(D,q,l)
r1=size(D,1);
a=0;
s1=r1/l;
for i=1:s1
D1=D((i-1)*l+1:i*l,1:q);
if DauMTTMax(D1)==1—D1==zeros(1,q)
a=a+1;
end
end
end
```

23. Subproblem

```
function DauMT = DauMTTMax(D1)
[r1,r2]=size(D1);
E1=D1;zeros(r1,r2);
E2=D1;zeros(r1,r2);
if (SumMatrix(E1) >= 1&SumMatrix(E2) >= 1)|SumMatrix(E1) >= 1
DauMT = 1;
else DauMT = 0;
end
```

end

24. Subproblem

```
function a = TextDMin(D,q,l)
r=size(D,1);
a=0;
s1=r/l;
for i=1:s1
D1=D((i-1)*l+1:i*l,1:q);
if DauMTTMin(D1)==1-D1==zeros(l,q)
a=a+1;
end
end
end
```

25. Subproblem

```
function DauMT = DauMTTMin(D1)
[r1, r2]=size(D1);
E1=D1;zeros(r1,r2);
E2=D1;zeros(r1,r2);
if (SumMatrix(E1)i=1 & SumMatrix(E2)i=1)—SumMatrix(E2)i=1
DauMT = 1;
else DauMT = 0;
end
end
```

Summary of the thesis in French

L'objectif de cette thèse est d'étudier des propriétés des points d'équilibre dans des réseaux de transport multicrière et de développer des méthodes numériques permettant de trouver l'ensemble de tous les points d'équilibre ou une partie représentative de cet ensemble. Le travail comporte cinq chapitres.

Le chapitre 2 est un rappel de certaines notions que nous utilisons dans les autres. Nous y rappelons le concept de point optimal de Pareto, les fonctions multivoques et les problèmes d'inégalité variationnelle. Nous introduisons certaines fonctions de scalarisation, en particulier les fonctions monotones augmentées et les fonctions distance signées augmentées, puis établissons quelques propriétés que nous allons utiliser plus tard. Voici ces fonctions.

Les fonctions monotones augmentées:

$$g_a^\varepsilon(x) = \min_{i=1, \dots, n} (x_i - a_i) + \varepsilon \sum_{i=1}^n (x_i - a_i)$$

$$G_a^\varepsilon(x) = \max_{i=1, \dots, n} (x_i - a_i) + \varepsilon \sum_{i=1}^n (x_i - a_i)$$

Les fonctions distance signées augmentées:

$$d_a^\varepsilon(x) = \Delta_{(\mathbb{R}_+^n)^c}(x - a) + \varepsilon \sum_{i=1}^n (x_i - a_i)$$

$$D_a^\varepsilon(x) = \Delta_{-\mathbb{R}_+^n}(x - a) + \varepsilon \sum_{i=1}^n (x_i - a_i)$$

où pour A un sous-ensemble non vide de \mathbb{R}^n , Δ_A est la fonction distance signée qui est définie par

$$\Delta_A(x) = d(x, A) - d(x, A^c),$$

où $d(x, A)$ est la distance euclidienne de x à A , et A^c est le complémentaire de A de \mathbb{R}^n .

Dans le chapitre 3, nous décrivons les réseaux de transport qui sont étudiés dans cette thèse. Dans chaque modèle, nous rappelons les définitions des points d'équilibre et donnons une relation entre ces définitions. Nous présentons également certains contre-exemples pour certains résultats existant dans la littérature récente sur ce sujet.

Dans le chapitre 4 nous traitons les réseaux de transport multi-critériès mono-produit sans contraintes de capacité. Notons que K dans ce chapitre est l'ensemble de tous les flots faisables Y satisfaisant les conditions suivantes:

$$y_p \geq 0 \quad \forall p \in P; \quad (9.1)$$

$$\sum_{p \in P_w} y_p = d_w \quad \forall w \in W. \quad (9.2)$$

Dans un premier temps, nous construisons deux problèmes d'optimisation dont les solutions sont exactement l'ensemble des points d'équilibre du modèle initiale. Ce résultat est présenté dans le Théorème 4.1.1. comme suit.

Théorème 4.1.1 *Si \bar{y} est un flot faisable, alors les assertions suivantes sont équivalentes:*

(i) \bar{y} est un équilibre vectoriel.

(ii) \bar{y} est une solution optimale du problème suivant, noté (P1):

$$\begin{aligned} & \text{minimiser} \quad \sum_{p \in P_w, w \in W} y_p d[c_p(y), \text{Min}(C_w(y))] \\ & \text{sous la contrainte} \quad y \in K \end{aligned}$$

et la valeur optimale de ce problème est nulle.

(iii) \bar{y} est une solution optimale du problème suivant, noté (P2):

$$\begin{aligned} & \text{minimiser} \quad \sum_{p \in P_w, w \in W} y_p \sum_{p' \in P_w} [c_p(y) - c_{p'}(y)]^T H_+[c_p(y) - c_{p'}(y)]. \\ & \text{sous la contrainte} \quad y \in K \end{aligned}$$

et la valeur optimale de ce problème est nulle, et où la fonction $H_+ : \mathbb{R}^l \rightarrow \mathbb{R}^l$ est la version vectorielle de la fonction de Heaviside Step qui est définie par

$$H_+(c_p(y) - c_{p'}(y)) = \begin{cases} (1, \dots, 1)^T & \text{si } c_p(y) - c_{p'}(y) \geq 0 \\ 0 & \text{sinon.} \end{cases}$$

Dans un second temps, nous établissons certaines propriétés importantes de continuité et dérivabilité génériques des fonctions objectifs, qui sont introduites dans le Théorème 4.2.4 et la Proposition 4.2.7:

Théorème 4.2.4 *Supposons que les fonctions de coût vectorielles $c_{p_i}, i = 1, \dots, m$ sont continues (respectivement Lipschitz ou localement différentiables). Alors chaque ensemble ouvert dans \mathbb{R}^m contient un sous-ensemble ouvert où les fonctions objectifs ϕ et ψ des problèmes (P1) et (P2) sont continues (respectivement Lipschitz ou différentiable localement).*

Proposition 4.2.7 *Supposons que les fonctions de coût vectorielles c_{p_1}, \dots, c_{p_m} sont continues. Alors la fonction ϕ et ψ sont continues en chaque équilibre vectoriel. Si en outre c_{p_1}, \dots, c_{p_m} sont localement calmes en un équilibre vectoriel, alors ϕ et ψ sont également localement calmes en cet équilibre.*

Dans un troisième temps, nous donnons une formule permettant de calculer le gradient des fonctions objectifs, qui nous permet de modifier la méthode de gradient réduit de Frank-Wolfe pour obtenir une direction de descente vers une solution optimale. Ces résultats sont présentés dans le Théorème 4.2.5 et le Théorème 4.2.6:

Théorème 4.2.5 *Supposons que les fonctions de coût vectorielles $c_{p_i}, i = 1, \dots, m$ sont différentiables. Alors pour chaque point y en dehors de certain sous-ensemble négligeable et pour chaque chemin p_i , il existe un chemin $p_{\nu(i)}$ de $P_{w(i)}$ tel que*

- (i) $c_{p_{\nu(i)}}(y) \in \text{Min}C_{w(i)}(y)$
- (ii) $d[c_{p_i}(y), \text{Min}C_{w(i)}(y)] = \|c_{p_i}(y) - c_{p_{\nu(i)}}(y)\|$
- (iii) ϕ est différentiable en y et son gradient est donné par

$$\nabla\phi(y) = \begin{pmatrix} \|c_{p_1}(y) - c_{p_{\nu(1)}}(y)\| \\ \vdots \\ \|c_{p_m}(y) - c_{p_{\nu(m)}}(y)\| \end{pmatrix} + \sum_{i=1}^m y_{p_i} \begin{pmatrix} \frac{c_{p_i}(y) - c_{\nu(i)}(y)}{\|c_{p_i}(y) - c_{p_{\nu(i)}}(y)\|} \left(\frac{\partial c_{p_i}}{\partial y_{p_1}}(y) - \frac{\partial c_{p_{\nu(i)}}}{\partial y_{p_1}}(y) \right) \\ \vdots \\ \frac{c_{p_i}(y) - c_{p_{\nu(i)}}(y)}{\|c_{p_i}(y) - c_{p_{\nu(i)}}(y)\|} \left(\frac{\partial c_{p_i}}{\partial y_{p_m}}(y) - \frac{\partial c_{p_{\nu(i)}}}{\partial y_{p_m}}(y) \right) \end{pmatrix}.$$

Théorème 4.2.6 *Supposons que les fonctions de coût vectorielles $c_{p_i}, i = 1, \dots, m$ sont différentiables. Alors pour chaque point y en dehors de certain sous-ensemble négligable et pour chaque chemin p_i , il existe un sous-ensemble $J_i(y) \subseteq I_{w(i)}$ tel que*

- (i) $c_{p_i}(y) \geq c_{p_j}(y)$ pour chaque $j \in J_i(y)$
- (ii) $\psi(y) = \sum_{i=1}^m y_{p_i} \langle \sum_{j \in J_i(y)} (c_{p_i}(y) - c_{p_j}(y)), e \rangle$
- (iii) ψ est différentiable en y et son gradient est donné par

$$\nabla\psi(y) = \begin{pmatrix} \sum_{j \in J_1(y)} \langle c_{p_1}(y) - c_{p_j}(y), e \rangle \\ \vdots \\ \sum_{j \in J_m(y)} \langle c_{p_m}(y) - c_{p_j}(y), e \rangle \end{pmatrix} + \sum_{i=1}^m y_{p_i} \begin{pmatrix} \sum_{j \in J_i(y)} \langle \frac{\partial c_{p_i}}{\partial y_{p_1}}(y) - \frac{\partial c_{p_j}}{\partial y_{p_1}}(y), e \rangle \\ \vdots \\ \sum_{j \in J_i(y)} \langle \frac{\partial c_{p_i}}{\partial y_{p_m}}(y) - \frac{\partial c_{p_j}}{\partial y_{p_m}}(y), e \rangle \end{pmatrix}.$$

Dans un quatrième temps, nous proposons un algorithme et prouvons sa convergence pour générer une représentation de l'ensemble des points d'équilibre. Puisque les fonctions objectives de nos problèmes d'optimisation ne sont pas continues, une méthode de lissage est également considérée afin d'utiliser quelques techniques d'optimisation globale. *En particulier, nous utilisons les approximations analytiques suivantes*

$$\tilde{H}_\nu(x) = \left(\prod_{i=1}^l \frac{1 + \tanh(\nu x_i)}{2} \right) e \quad \text{for } \nu \geq 1,$$

qui produisent ainsi des approximations lisses de la fonction objectif ψ quand les fonctions de coût sont continues:

$$\psi_\nu(y) := \sum_{p \in P_w, w \in W} y_p \sum_{p' \in P_w} [c_p(y) - c_{p'}(y)]^T \tilde{H}_\nu[c_p(y) - c_{p'}(y)].$$

Le problème d'optimisation notée $(P2_\nu)$, est obtenu à partir de $(P2)$ en remplaçant ψ par ψ_ν .

Enfin, nous introduisons le concept de point d'équilibre robuste, puis nous établissons des critères de robustesse et une formule permettant de calculer le rayon de robustesse. Ces résultats sont présentés dans le Théorème 4.4.1 et le Corollaire 4.4.5 comme suivants.

Théorème 4.4.1 *Soit $\bar{y} \in K$ un équilibre vectoriel de G . Les assertions suivantes sont équivalentes.*

(i) \bar{y} est robuste.

(ii) \bar{y} est une solution optimale du problème d'optimization suivant, noté (P'_1)

$$\begin{aligned} & \text{minimiser} \quad \sum_{p \in P_w, w \in W} y_p (d[c_p(y), \text{Min}(C_w(y))]) \\ & \quad + \sum_{i \in I_w(y), p_i \neq p} \chi_{\{0\}} (\|c_p(y) - c_{p_i}(y)\|) \\ & \text{sous la contrainte} \quad y \in K, \end{aligned}$$

et la valeur optimale de ce problème est égale à zéro, où pour $w \in W$ et $y \in K$, $I_w(y)$ désigne l'ensemble des indices i telle que $p_i \in P_w$ et $c_{p_i}(y) \in \text{Min}(C_w(y))$, et $\chi_{\{0\}}$ est la fonction caractéristique de $\{0\}$.

(iii) Il existe $\epsilon > 0$ tel que pour chaque $w \in W, p \in P_w$ avec $\bar{y}_p > 0$, on a :

$$(\tilde{c}_p(\bar{y}) - \mathbb{R}_+^l) \cap (\tilde{C}_w(\bar{y}) \setminus \{\tilde{c}_p(\bar{y})\}) = \emptyset$$

pour tous $\tilde{c}_{p_i}, p_i \in P_w$ satisfaisant $\|\tilde{c}_{p_i}(\bar{y}) - c_{p_i}(\bar{y})\| \leq \epsilon$.

Corollaire 4.4.5 Soit $\bar{y} \in K$ équilibre vectoriel robuste. Alors le rayon de robustesse en \bar{y} est donné par

$$r(\bar{y}) = \frac{\sqrt{l}}{2} \min_{w \in W, i \in I_w^+(\bar{y})} \min_{p' \in P_w \setminus \{p_i\}} \|(c_{p'}(\bar{y}) - c_{p_i}(\bar{y}))^+\|$$

où $(c_{p'}(\bar{y}) - c_{p_i}(\bar{y}))^+$ désigne la partie positive du vecteur $c_{p'}(\bar{y}) - c_{p_i}(\bar{y})$ et $I_w^+(\bar{y}) = \{i \in I_w(\bar{y}) : \bar{y}_{p_i} > 0\}$.

Dans le chapitre 5 nous étudions des points d'équilibre vectoriel dans le réseau de transport multi-critère mono-produit sous contraintes de capacité. Tout d'abord, nous proposons un problème d'optimisation équivalent et nous établissons également certaines propriétés importantes de continuité et dérivabilité génériques de la fonction objectif. En suite, nous donnons une formule permettant de calculer le gradient de la fonction objectif qui est présenté dans le Théorème 5.3.2 ci-dessous. Puis nous appliquons l'algorithme proposé dans le chapitre 4 avec quelques modifications permettant d'obtenir un sous-ensemble des solutions optimales qui sont des points d'équilibre de notre modèle. Des exemples numériques sont également présentés afin d'illustrer notre approche.

Théorème 5.3.2 Supposons que les fonctions de coût vectorielles $c_{p_i}, i = 1, \dots, m$ sont différentiables. Alors pour chaque point y en dehors de certain sous-ensemble négligable et pour chaque chemin p_i , il existe un sous-ensemble $J_i(y) \subseteq I_{w(i)}$ tel que

- (i) $c_{p_i}(y) \geq c_{p_j}(y)$ pour chaque $j \in J_i(y)$
- (ii) $\psi(y) = \sum_{i=1}^m (y_{p_i} - l_{p_i}) \langle \sum_{j \in J_i(y)} (u_{p_j} - y_{p_j}) (c_{p_i}(y) - c_{p_j}(y)), e \rangle$
- (iii) ψ est différentiable en y et son gradient est donné par

$$\begin{aligned} \frac{\partial \psi(y)}{\partial y_k} &= \sum_{i=1}^m (y_{p_i} - l_{p_i}) \sum_{j \in J_i(y)} (u_{p_j} - y_{p_j}) \left\langle \frac{\partial c_{p_i}(y)}{\partial y_k} - \frac{\partial c_{p_j}(y)}{\partial y_k}, e \right\rangle \\ &+ \sum_{\substack{i=1 \\ i \neq k}}^m (l_{p_i} - y_{p_i}) \langle c_{p_i}(y) - c_{p_k}(y), e \rangle + \sum_{j \in J_k(y)} (u_{p_j} - y_{p_j}) \langle c_{p_k}(y) - c_{p_j}(y), e \rangle \end{aligned}$$

pour $k = 1, \dots, m$.

Dans le dernier chapitre nous considérons des points d'équilibre fort dans le réseau de transport multi-critère multi-produit sous contraintes de capacité. Alors K dans ce chapitre est l'ensemble de tous les flots faisables Y satisfaisant les conditions suivantes:

$$l_p \leq y_p \leq u_p \quad \forall p \in P; \quad (9.3)$$

$$\sum_{p \in P_w} y_p = d_w \quad \forall w \in W. \quad (9.4)$$

Nous établissons des conditions d'existence des points d'équilibre fort. Nous produisons des relations entre des points d'équilibre fort et des points d'équilibre par rapport à une famille de fonctions. Ces résultats sont présentés dans les Lemmes 6.2.1 et 6.2.3 et le Corollaire 6.2.4 comme suit:

Lemme 6.2.1 *Soit un flot faisable \bar{Y} donné. Si la famille \mathcal{F} contient des fonctions croissantes en $\{\bar{C}_{p_i}, p_i \in P\}$, alors chaque \mathcal{F} -équilibre est un équilibre vectoriel fort. Inversement, si \mathcal{F} satisfait la condition suivante: Pour chaque $w \in W$ et $p_i, p_{i'} \in P_w$ on a l'implication*

$$f(\bar{C}_{p_i}) > f(\bar{C}_{p_{i'}}) \quad \forall f \in \mathcal{F} \Rightarrow \bar{C}_{p_i} \geq \bar{C}_{p_{i'}}, \quad (9.5)$$

alors chaque équilibre vectoriel fort est un \mathcal{F} -équilibre.

Lemme 6.2.3 *Soit un flot faisable \bar{Y} donné. Il existe $\epsilon_0 > 0$ tel que pour tous $\epsilon \in (0, \epsilon_0)$, chacune des familles de fonctions croissantes ci-dessous satisfait à la condition (9.5): $\mathcal{F}_1 = \{D_{\bar{C}_{p_i}}^\epsilon : p_i \in P\}$, $\mathcal{F}_2 = \{G_{\bar{C}_{p_i}}^\epsilon : p_i \in P\}$, $\mathcal{F}_3 = \{d_{\bar{C}_i}^\epsilon : p_i \in P\}$ et $\mathcal{F}_4 = \{g_{\bar{C}_{p_i}}^\epsilon : p_i \in P\}$.*

Corollaire 6.2.4 *Soit un flot faisable \bar{Y} donné. Alors il est équilibre vectoriel si et seulement s'il est un \mathcal{F}_i -équilibre pour certain $i \in \{1, \dots, 4\}$.*

Nous établissons également une relation entre des points d'équilibre fort et les points efficaces de l'ensemble des valeurs de la fonction de coût, qui est présentée dans le Théorème 6.2.5 comme suit:

Théorème 6.2.5 *Soit \bar{Y} un flot faisable. Il est un équilibre vectoriel fort si et seulement si pour tout $w \in W$ les conditions sont satisfaites*

$$(i) (\bar{C}_{L_w} + \mathbb{R}_+^{l \times q} \setminus \{0\}) \cap (\bar{C}_{U_w} \cup \bar{C}_{E_w}) = \emptyset,$$

$$(ii) (\bar{C}_{U_w} - \mathbb{R}_+^{l \times q} \setminus \{0\}) \cap (\bar{C}_{L_w} \cup \bar{C}_{E_w}) = \emptyset,$$

$$(iii) \bar{C}_{E_w} \text{ est auto-maximal,}$$

où pour $w \in W$,

$$L_w := \{p_i \in P_w \text{ tel que } \bar{Y}_{p_i} = L_{p_i}\}$$

$$U_w := \{p_i \in P_w \text{ tel que } \bar{Y}_{p_i} = U_{p_i}\}$$

$$E_w := \{p_i \in P_w \text{ tel que } L_i \leq \bar{Y}_{p_i} \leq U_{p_i}\}$$

et pour un ensemble I , $\bar{C}_I = \{\bar{C}_{p_i}, p_i \in I\}$.

En plus nous utilisons les fonctions croissantes déjà discutées au chapitre 2 pour construire des problèmes d'inégalité variationnelle, dont les solutions sont les points d'équilibre fort. Ces résultats sont présentés dans le Théorème 6.2.7 et le Théorème 6.2.8:

Théorème 6.2.7 *Soit \bar{Y} un flot faisable. S'il satisfait la condition: $C_{p_\alpha}(Y) \geq C_{p_\beta}(Y)$ implique soit $Y_{p_\alpha} = L_{p_\alpha}$ ou $Y_{p_\beta} = U_{p_\beta}$ pour $p_\alpha, p_\beta \in P_w$ et*

$$\sum_{(i,w,j) \in \Gamma} \left(\min_{\alpha \in A_w} D_{\bar{C}_\alpha}^\epsilon(\bar{C}_i) \right) (y_{ij} - \bar{y}_{ij}) \geq 0 \text{ pour tout } Y \in K, \quad (9.6)$$

pour certain $\epsilon > 0$, alors il est un équilibre vectoriel fort. Inversement, si \bar{Y} est un équilibre vectoriel fort, alors il existe $\epsilon_0 > 0$ tel que \bar{Y} satisfait (9.6) pour tout $\epsilon \in (0, \epsilon_0)$.

Théorème 6.2.8 *Soit \bar{Y} un flot faisable. S'il satisfait la condition: $C_{p_\alpha}(Y) \geq C_{p_\beta}(Y)$ implique soit $Y_{p_\alpha} = L_{p_\alpha}$ soit $Y_{p_\beta} = U_{p_\beta}$ pour $p_\alpha, p_\beta \in P_w$ et*

$$\sum_{(i,w,j) \in \Gamma} \left(\max_{\beta \in B_w} g_{C_\beta}^\epsilon(\bar{C}_i) \right) (y_{ij} - \bar{y}_{ij}) \geq 0 \text{ pour tout } Y \in K, \quad (9.7)$$

pour certain $\epsilon > 0$, alors il est un équilibre vectoriel fort. Inversement, si \bar{Y} est un équilibre vectoriel fort, alors il existe $\epsilon_0 > 0$ tel que \bar{Y} satisfait (9.7) pour tout $\epsilon \in (0, \epsilon_0)$.

La dernière partie de ce chapitre est consacrée à un algorithme permettant de trouver des points d'équilibre d'un réseau multi-critère sous contraintes de capacité. Certains exemples numériques sont donnés pour illustrer notre méthode.

Nous fermons la thèse avec une liste de références et appendix contenant le code matlab de nos algorithmes.