Traffic Assignment: Methods and Simulations for an Alternative Formulation of the Fixed Demand Problem

Ovidiu Bagdasar*, Stuart Berry, Sam O'Neill

Department of Electronics, Computing and Mathematics, University of Derby, Kedleston Road, DE22 1GB, United Kingdom

Nicolae Popovici

Babeş-Bolyai University, Faculty of Mathematics and Computer Science, 400084 Cluj-Napoca, Romania

Ramachandran Raja

Ramanujan Centre for Higher Mathematics, Alagappa University, Karaikudi-630 004, India

Abstract

Motorists often face the dilemma of choosing the route enabling them to realise the fastest (i.e., shortest) journey time. In this paper we examine discrete and continuous optimisation and equilibrium-type problems for a simplified parallel link traffic model using a variance based approach. Various methodologies used for solving these problems (brute force, dynamic programming, tabu search, steepest descent) are explored and comparison is made with the Beckmann cost function traditionally employed in transport modelling.

Key words: Traffic assignment, Optimal flow, Equilibrium flow, Tabu search, Dynamic programming

1. Introduction

A dilemma often facing transport planners is to choose whether to leave motorists free to make their own route choices where they aim to minimise their own travel times, or to try to actively manage the traffic flows in order to minimise the total journey times for all motorists travelling between origin and destination, i.e., whether to plan or not to plan?

Assuming that journey time is the only criteria for route choice, car travellers may be seen to act selfishly as self optimisers insofar as they usually want to minimise their own journey times. As a consequence of this policy, in the absence of any effective traffic control measures, route switching by the travellers to what they perceive to be the fastest route will act to produce a steady state where all (used) routes have an approximately equal travel time. The resultant total travel time at this *equilibrium* flow will be greater than that obtained for the *optimal* flow, achieved in the presence of a perfect traffic control system.

*Corresponding author

Email addresses: o.bagdasar@derby.ac.uk (Ovidiu Bagdasar), s.berry@derby.ac.uk (Stuart Berry), S.Oneill@derby.ac.uk (Sam O'Neill), popovici@math.ubbcluj.ro (Nicolae Popovici), rajarchm2012@gmail.com (Ramachandran Raja)

These two states of the system, as defined by Wardrop [1], are generally referred to as user equilibrium (UE) and system optimal (SO). This difference between UE and SO travel times can lead to the decidedly counter-intuitive result that additions to road capacity, typically through more road construction, resulting in increased rather than the expected slower journey times.

This class of problems, known as the Traffic Assignment Problem (TAP), was first formulated by Dafermos and Sparrow [2] and has a number of known mathematical programs for solving variations of the fixed demand problem (where the number of cars being transported from an origin to destination is fixed) [3]. We present a closely related formulation of SO and UE using a simplified parallel link model. Several discrete and continuous versions of this model are presented, together with a comparison between various solution methodologies. Various minimisation problems with separable goal function and simple constraints have been treated by numerous authors. The interested reader may consult the survey paper of Patriksson [4], or the more recent work focused on networks involving parallel routes by Krylatov [5].

Concerning features of more realistic traffic models, we just mention the network equilibrium problems under demand uncertainty and capacity constraints studied via scalarization approaches by Cao *et al.* [6], or the seasonal heteroscedasticity in vehicular traffic flow investigated by Huang *et. al* [7].

A fundamental feature of road transportation is that car travel time is dependent on the number of cars accessing the route. If there are $m \geq 2$ routes between the origin and destination points, the time t_i for a car accessing route i $(i = 1, \ldots, m)$ is a monotonic increasing polynomial function of the traffic flow x_i as measured in"units of vehicle" per "unit of tim" accessing route i, namely

$$t_i = f_i(x_i) = a_i + b_i \left(\frac{x_i}{c_i}\right)^{p_i}, \quad \text{where } p_i \geq 1,\ x_i, a_i \geq 0 \text{ and } b_i, c_i > 0, \tag{1}$$

proposed by Youn *et al.* [8] and referred to as the *BPR Formula* (Bureau of Public Roads [9]).

The term x_i/c_i is effectively the (traffic) flow to capacity ratio of the road. Road capacity may be conceptualised in different ways (see Minderhoud et al 1997) [10] but here it is taken to mean the specific design capacity of the road. Since many, if not most, roads operate at traffic flows well above their design capacity this allows for the situation where $x > c$ if not $x \gg c$. The assumptions are that travel time along two roads having the same speed limit and length should be equal when the traffic levels meet the design capacity. It may be noted that as $x_i \to \infty$ we have $f_i(x_i) \to \infty$. Also, if $x_i = 1$ then $f_i(x_i) \simeq a_i$, while if $x_i = c_i$ we have $f_i(x_i) = a_i + b_i$ (individual travel time at the design capacity).

Denoting the cost of transporting x_i vehicles along route i $(i = 1, \ldots, m)$ by

$$g_i(x_i) = x_i f_i(x_i), \tag{2}$$

the total travel time $T(x)$ for n vehicles distributed on m routes is given by the formula

$$T(x) = \sum_{i=1}^{m} g_i(x_i) = \sum_{i=1}^{m} x_i f_i(x_i), \tag{3}$$

where $x = (x_1, \ldots, x_m) \in \mathbb{N}^m$ and $x_1 + \cdots + x_m = n$.

2. Mathematical Programs

In this section we present some mathematical programs having discrete or continuous state spaces, which are related to optimal flow, equilibrium flow, and optimal equilibrium flow.

2.1. Optimal Flow Programs

First, we define two mathematical programs related to optimal flow, whereby a fixed number of cars is assigned to each route in such a way that the total travelling time is minimised. We consider a discrete program having non-negative integer solutions, and a continuous counterpart which has the solution in the set of non-negative real numbers, denoted by $\mathbb{R}_{\geq 0} = \{x \in \mathbb{R} \mid x \geq 0\}$.

Discrete Optimal Flow [DOF]

$$\begin{cases} \text{Minimise } T(x) \\ \text{subject to} \\ x_1 + \cdots + x_m = n \\ x_1, \ldots, x_m \in \mathbb{N}. \end{cases} \tag{4}$$

Continuous Optimal Flow [COF]

$$\begin{cases} \text{Minimise } T(x) \\ \text{subject to} \\ x_1 + \cdots + x_m = n \\ x_1, \ldots, x_m \in \mathbb{R}_{\geq 0}. \end{cases} \tag{5}$$

2.2. Equilibrium Flow Programs

Second, we define mathematical programs for the equilibrium flow assignment, whereby individual travel times are as similar as possible across all routes. The steady state traffic flow along each route could be seen as a solution of the following system of equations:

$$\begin{cases} f_1(x_1) = \ldots = f_m(x_m) \\ x_1 + \cdots + x_m = n \\ x_1, \ldots, x_m \in \mathbb{N}. \end{cases} \tag{6}$$

However, this system is often inconsistent, hence one may seek to minimise the variance of travel times. The following mathematical programs can be formulated as alternatives to the equilibrium system (6):

Discrete Equilibrium Flow [DEF]

$$\begin{cases} \text{Minimise } \sigma^2(f(x_1), \ldots, f(x_m)) \\ \text{subject to} \\ x_1 + \cdots + x_m = n \\ x_1, \ldots, x_m \in \mathbb{N}. \end{cases} \tag{7}$$

Continuous Equilibrium Flow [CEF]

$$\begin{cases} \text{Minimise } \sigma^2(f(x_1), \ldots, f(x_m)) \\ \text{subject to} \\ x_1 + \cdots + x_m = n \\ x_1, \ldots, x_m \in \mathbb{R}_{\geq 0}. \end{cases} \tag{8}$$

In contrast to the UE defined by Wardrop's first principle [1], *"The journey times on all routes are equal, and less than those which would be experienced by a single vehicle on any unused route"*, the above model considers the cost differences between all roads, whether used or not. If the demand is sufficiently high, then the solution to the above model and Wardrop's first principle will be one and the same. The difference between the two can be especially noticed at low demand.

2.3. Formulation of the programs [DOF] (4) *and [COF]* (5) *as equilibrium problems*

It was shown in [11] that the solution of [COF] (5) corresponds to that of the equilibrium system

$$\begin{cases} g_1'(x_1) = \ldots = g_m'(x_m) \\ x_1 + \cdots + x_m = n \\ x_1, \ldots, x_m \geq 0, \end{cases} \tag{9}$$

where

$$g_i'(x_i) = f_i(x_i) + x_i f_i'(x_i). \tag{10}$$

As an alternative of the equilibrium system (9) we can consider the following mathematical programs:

Discrete Optimal Equilibrium Flow [DOEF]

$$\begin{cases} \text{Minimise } \sigma^2(g'(x_1), \ldots, g'(x_m)) \\ \text{subject to} \\ x_1 + \cdots + x_m = n \\ x_1, \ldots, x_m \in \mathbb{N}. \end{cases} \tag{11}$$

$$\begin{cases} \text{Minimise } \sigma^2(g'(x_1), \ldots, g'(x_m)) \\ \text{subject to} \\ x_1 + \cdots + x_m = n \\ x_1, \ldots, x_m \in \mathbb{R}_{\geq 0}. \end{cases} \tag{12}$$

2.4. Existence of a Solution for the Continuous Equilibrium Systems

The existence of a solution for all defined equilibrium systems can be shown to depend on the demand, which in turn must be greater than a certain value. Here we briefly discuss a necessary condition which involves the parameters of the model.

Theorem 2.1. *Let $m \geq 1$ be an integer and $f_i : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, $i = 1, \ldots, m$ strictly increasing and unbounded continuous functions. The system* (6) *has a solution if and only if the demand $D = x_1 + \cdots + x_m$ satisfies*

$$D = \sum_{i=1}^{m} x_i > \sum_{i=1}^{m} f_i^{-1}(M_0) = D_0, \tag{13}$$

where $M_0 = \max_{1 \leq i \leq m} \{f_i(0)\}$.

Proof. Without loss of generality, functions can be relabeled such that $f_1(0) \leq f_2(0) \leq \cdots \leq f_m(0) = M_0$.

If the system (6) has a solution $(x_1, \ldots, x_m)$, then

$$f_1(x_1) = \cdots = f_m(x_m) > f_m(0) = M_0. \tag{14}$$

From the equality $f_i(x_i) = f_m(x_m)$ written for $i = 1, \ldots, m$, one recovers the unique value

$$x_i = f_i^{-1}(f_m(x_m)) > f_i^{-1}(M_0). \tag{15}$$

Adding for $i = 1, \ldots, m$ we obtain $D > D_0$.

Conversely, if $D > D_0$ we prove that the system (6) has a solution. Indeed, equation

$$D = \sum_{i=1}^{m} x_i = \sum_{i=1}^{m} f_i^{-1}(f_m(x_m)),$$

has a solution $x_m > 0$, hence by (15) we also obtain x_i, $i = 1, \ldots, m-1$ which satisfy (14).

Example. For the functions $f_i(x)$ defined by (1), we have the identity

$$D_0 = \sum_{i=1}^{m} c_i \left(\frac{M_0 - a_i}{b_i} \right)^{\frac{1}{p_i}}. \tag{16}$$

3. Methodology

In this section we discuss some methods used in the analysis of the optimal flow and equilibrium flow problems, providing some details especially for the dynamic programming and tabu search approaches. We also discuss the complexity of these methods, in relation to computations detailed in [12], where we have also analyzed exhaustive search and numerical methods based on steepest descent.

3.1. Solution of the mathematical program [DOF] (4) by dynamic programming

The cost function T of the mathematical program [DOF] (4) has separable variables, being a sum of terms containing independent variables (3). Since the feasible set $S \cap \mathbb{N}^m$ is finite, problem (4) can be solved using Bellman's algorithm of dynamic programming (see Bellman [13] and Bazaraa *et al.* [14]).

Defining recursively the Bellman functions $G_1, \ldots, G_m : [0, n] \cap \mathbb{N} \to \mathbb{R}$ for all $c \in [0, n] \cap \mathbb{N}$

$$\begin{cases} G_1(c) & = g_1(c); \\ G_k(c) & = \min_{x \in [0,c] \cap \mathbb{N}} \left[g_k(x) + G_{k-1}(c-x) \right], \quad k = 2, 3, \ldots, m. \end{cases} \tag{17}$$

Then, the optimal value of problem (4) is given by

$$\min\{T(x) \mid x = (x_1, \ldots, x_m) \in \mathbb{N}^m,\ x_1 + \cdots + x_m = n\} = G_m(n). \tag{18}$$

An optimal solution $x^0 = (x_1^0, \ldots, x_m^0)$ of problem (4) can be deduced by the backward recursive procedure:

Let $c := n$ and choose $x_m^0 \in \underset{x \in [0,c] \cap \mathbb{N}}{\operatorname{argmin}} \left[g_m(x) + G_{m-1}(c-x) \right]$,

Let $c := n - x_m^0$ and choose $x_{m-1}^0 \in \underset{x \in [0,c] \cap \mathbb{N}}{\operatorname{argmin}} \left[g_{m-1}(x) + G_{m-2}(c-x) \right]$,

...

Let $c := n - x_m^0 - \ldots - x_3^0$ and choose $x_2^0 \in \underset{x \in [0,c] \cap \mathbb{N}}{\operatorname{argmin}} \left[g_2(x) + G_1(c-x) \right]$,

Let $x_1^0 := n - x_m^0 - \ldots - x_3^0 - x_2^0$.

A full explanation of this method and examples are given in [12].

3.2. Numerical optimisation methods

The polynomial functions $f_1, \ldots, f_m$ defined by (1) are convex $(0, \infty)$ (as all coefficients are non-negative), hence the exact solution can also be approximated by various numerical methods.

These numerical methods will be particularly relevant in the study of the equilibrium problem for numerous links m and large number of vehicles n. In that case the exhaustive search becomes ineffective, while Bellman's algorithm is not applicable due to the cost function having non-separable variables.

3.3. Heuristic method

A Tabu Search solution was implemented to solve the mathematical programs [DEF] (7), [CEF] (8), [DOEF] (11) and [COEF] (12). The basic 'Tabu Search Step' is detailed below and the implementation of the adaptive step is shown in Figure 1. For the equilibrium flow, the road travel time function $c_i(x_i) = f_i(x_i)$ (1) is used, whereas the road travel time for the optimal flow is $c_i(x_i) = g_i'(x_i) = f_i(x_i) + x_i f_i'(x_i)$ (10).

3.3.1. Tabu Search Step

Step 1. Initialization

a) Make an initial allocation of n vehicles to m routes such that the solution satisfies

$$x_i \geq 0, \quad i = 1, \ldots, m, \text{ with } \sum_{i=1}^{m} x_i = n.$$

b) Compute the travel time per vehicle along each road $i = 1, \ldots, m$, denoted by $R_i = c_i(x_i)$.

c) Define the objective function *OptSol* as the variance across all routes

$$OptSol = \sigma^2(R_1, \ldots, R_m).$$

d) Initialise the Tabu lists $U(i)$ and $D(i)$ to be the number of iterations before a given route i can be increased and decreased, respectively

$$U(i) = 0, \quad \forall i \in \{0, 1, \ldots, m\},$$
$$D(i) = 0, \quad \forall i \in \{0, 1, \ldots, m\}.$$

Step 2. Locate the following roads:

a) Road M where $R_M \geq R_i$ for all $i = 1, \ldots, m$ with $x_M > h$ and $D(M) = 0$;

b) Road N where $R_N \leq R_i$ for all $i = 1, \ldots, m$ with $U(N) = 0$.

Step 3. Reassign the traffic with the updated load values

$$x_M := x_M - h \text{ and } x_N := x_N + h.$$

Step 4. Update the Tabu list. Reduce the Tabu value for all routes, but those for M, N, so that:

- x_M cannot be increased until TabuTime has elapsed;
- x_N cannot be decreased until TabuTime has elapsed.

$$U(i) = \max(U(i) - 1, 0)$$
$$D(i) = \max(D(i) - 1, 0)$$
$$U(M) = U(M) + \text{TabuTime}$$
$$D(N) = D(N) + \text{TabuTime}$$

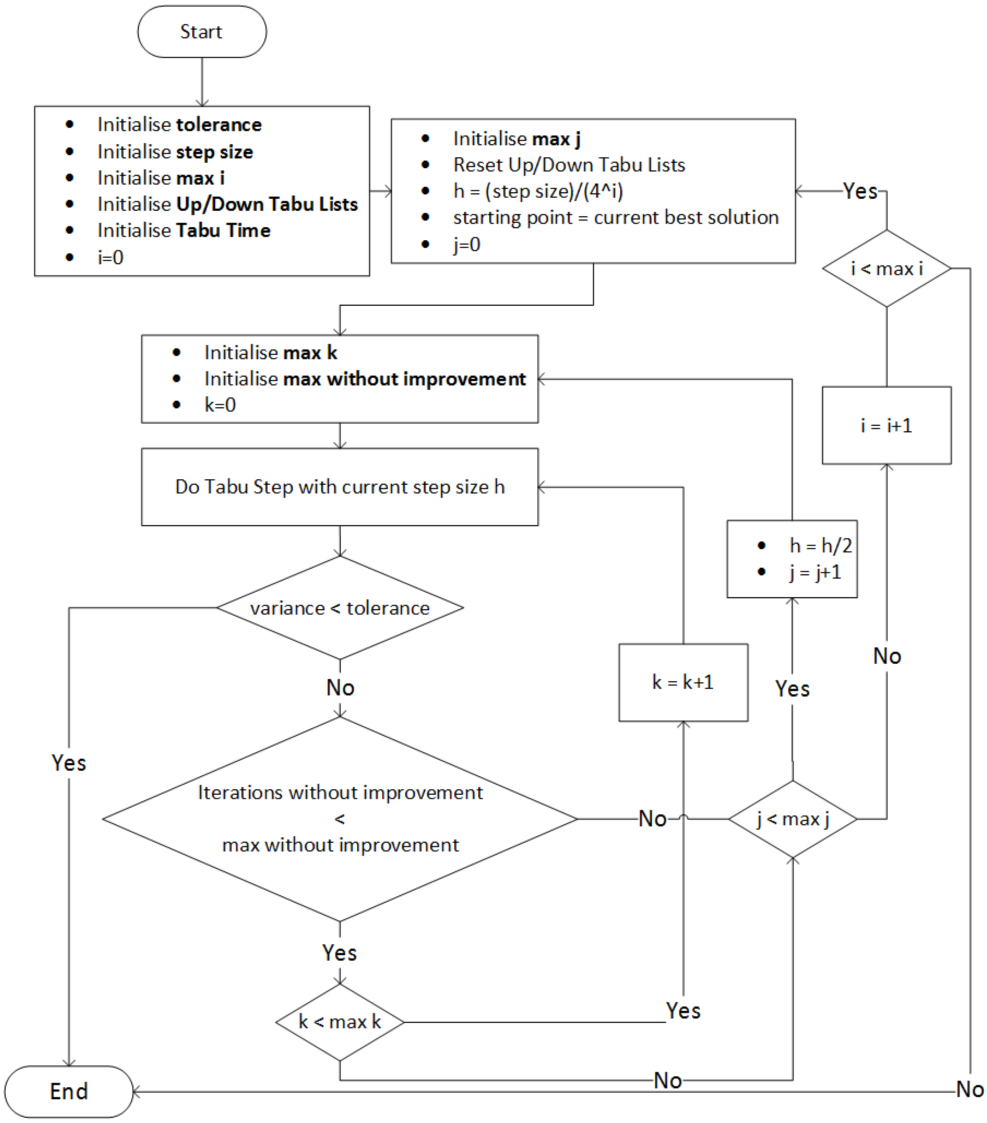


Figure 1: Flow chart detailing adaptive step size routine for the Tabu Search

Step 5. Update the value of the objective function

$$OptSol = \min\{\sigma^2(R_1, \ldots, R_m), OptSol\}.$$

Step 6. At the subsequent stages, repeat **Steps 2, 3, 4 and 5**.

Note. The starting step size is initialised as an integer power of 4, ensuring an appropriate started step size, and that the method can be used for both continuous and discrete based solutions.

$$h = 4^{\lfloor \log_{10}(n) \rfloor}.$$

In the discrete case the minimum stepsize will result in a value of 1.

3.4. Complexity calculations

Here we give the computational complexity for exhaustive (presented in more detail in [12]), dynamic programming and tabu search, as a function of the number of links m and the number of vehicles n.

3.4.1. Exhaustive Search

Let $N_m(n)$ denote the number of possible configurations $(x_1, \dots, x_m)$ such that $x_1, \dots, x_m \geq 0$ and $x_1 + \cdots + x_m = n$ (i.e., the size of the feasible space for (4)). Clearly, $N_1(n) = 1$ and recursively we have

$$N_2(n) = N_1(0) + N_1(1) + \cdots + N_1(n-1) + N_1(n) = \sum_{i=0}^{n} N_1(i) = \sum_{i=0}^{n} 1 = n + 1,$$

$$N_3(n) = N_2(0) + N_2(1) + \cdots + N_2(n-1) + N_2(n) = \sum_{i=0}^{n} N_2(i) = \sum_{i=0}^{n} i + 1 = \frac{(n+2)(n+1)}{2},$$

$$\dots$$

$$N_m(n) = \sum_{i=0}^{n} N_{m-1}(i) = \frac{1}{(m-1)!} \prod_{i=1}^{m-1} (n+i).$$

Therefore the complexity for an exhaustive search of n vehicles on m roads is,

$$O\left(\frac{n^{m-1}}{(m-1)!}\right).$$

3.4.2. Dynamic Programming

The complexity of dynamic programming depends on the main recursive operations given in Section 3.1. For fixed $k \in \{1, m\}$ and $c \in [0, n]$, the number of operations required to compute $G_k(c)$ is $c + 1$ by (17). Therefore, the number of operations required to compute $G_k(c)$ for $c \in [0, n] \cap \mathbb{N}$ is

$$\sum_{c=0}^{n} (c+1) = \frac{(n+1)(n+2)}{2}.$$

Evaluating for m roads, the total number of operations required will be

$$m\left[\frac{(n+1)(n+2)}{2}\right],$$

thus resulting in a complexity of

$$O\left(mn^2\right).$$

3.4.3. Tabu Search

Step 2 requires $2m$ operations per iteration. Given the different termination criteria used for the continuous problem we can only state that the maximum number of iterations is given by ijk (set by the user). For the discrete problem we can provide a strict upper bound by utilizing the fact that the step sizes considered are integers and for each iteration i, h is reduced by a factor of 4. An upper bound for the maximum number of iterations is $\log_{10}(n) \cdot j \cdot k$, thus giving a complexity $O(m \cdot \log_{10}(n))$.

4. Results

In Part A of this section we investigate solutions of the discrete/continuous optimal flow and equilibrium flow mathematical programs [DOF] (4), [COF] (5), [DEF] (7) and [CEF] (8) for a model with three roads. Exhaustive search is used to validate the discrete solutions obtained by dynamic programming and heuristic methods. We also illustrate the complexity of the chosen methods.

In Part B, using an extended 10 road model, we compare the continuous variance formulations of optimal flow [COEF] (12) and equilibrium flow [CEF] (8) with the continuous optimal flow program [COF] (5) and the classic Beckmann formulation for UE (19), defined for our parallel-links model as

User Equilibrium Flow [UEF]

$$\begin{cases} \text{Minimise } \sum_{i=1}^{m} \int_0^{x_i} f_i(x_i) \\ \text{subject to} \\ x_1 + \cdots + x_m = n \\ x_1, \ldots, x_m \in \mathbb{R}. \end{cases} \tag{19}$$

The relationship between the optimal and equilibrium states of the system for the variance-based and traditional formulations is illustrated through the *price of anarchy* [15], defined as

$$P_A = \frac{\text{Total cost at equilibrium flow}}{\text{Total cost at optimal flow}} \tag{20}$$

More details on selfish routing and the price of anarchy can be found in the book of Roughgarden [16].

4.1. Part A

Travel time functions (1) for a 3 road example are given in Table 1. In this case, moving x_1 cars along route 1 costs $g_1(x_1) = x_1 f_1(x_1)$. Similarly, $g_2(x_2) = x_2 f_2(x_2)$ and $g_3(x_3) = x_3 f_3(x_3)$. Therefore, the total travel time of $n = x_1 + x_2 + x_3$ cars along these routes is $T(x) = g_1(x_1) + g_2(x_2) + g_3(x_3)$, according to (3).

Road Parameters and Road Travel Time per Vehicle					
Road No.	a_i	$b_i = 0.15a_i$	c_i	p_i	$f_i(x_i) = a_i + b_i \left(\frac{x_i}{c_i}\right)^{p_i}$
1	1.85	0.2775	4000	2	$f_1(x_1) = 1.85 + 0.2775 \left(\frac{x_1}{4000}\right)^2$
2	1.5	0.225	1500	3	$f_2(x_2) = 1.5 + 0.225 \left(\frac{x_2}{1500}\right)^3$
3	2.15	0.3225	1000	5	$f_3(x_3) = 2.15 + 0.3225 \left(\frac{x_3}{1000}\right)^5$

Table 1: Road Cost Functions $f_i(x_i)$ for 3 road example.

For $x = (x_1, x_2, x_3) = (1000, 500, 500)$, the total travelling cost given by (3) is $T(x) = 3701.55$. This example highlights some key features of the traffic problems formulated in the introduction.

4.1.1. Solutions of the Discrete mathematical program [DOF]

Numerical solutions for the mathematical program [DOF] (4) with demands $\{5000, 6000, 7000, 8000\}$ computed by using Exhaustive Search (ES), Dynamic Programming (DP) and Tabu Search (TS) are given in Table 2. As expected, the exact solutions obtained by Exhaustive Search and Dynamic Programming are identical. The Tabu Search heuristic delivers solutions which are close enough to the exact solutions, suggesting the viability of this method for solving larger scale models.

Demand	Method	x_1	x_2	x_3	$T(x)$
5000	ES	2953	1444	603	9677.471
	DP	2953	1444	603	9677.471
	TS	2946	1453	601	9677.541
6000	ES	3686	1582	732	12101.67
	DP	3686	1582	732	12101.67
	TS	3669	1595	736	12101.9
7000	ES	4448	1729	823	14814.04
	DP	4448	1729	823	14814.04
	TS	4456	1725	819	14814.11
8000	ES	5225	1879	896	17883.06
	DP	5225	1879	896	17883.06
	TS	5225	1879	896	17883.06

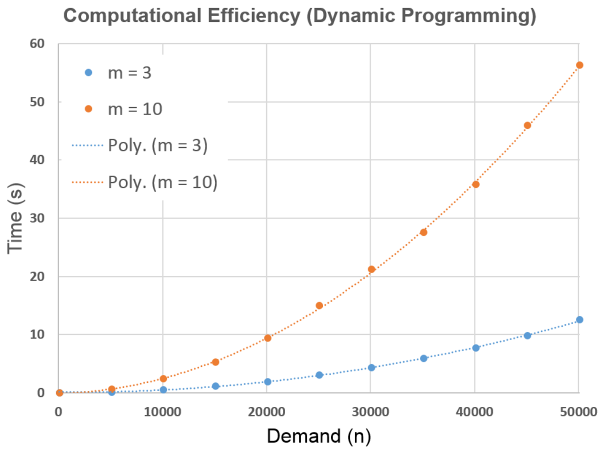
Table 2: Comparison of Integer Solutions for problem [DOF] (4) obtained by ES, DP and TS .

4.1.2. Comparison of Computation Efficiency of Methods for Discrete Solutions

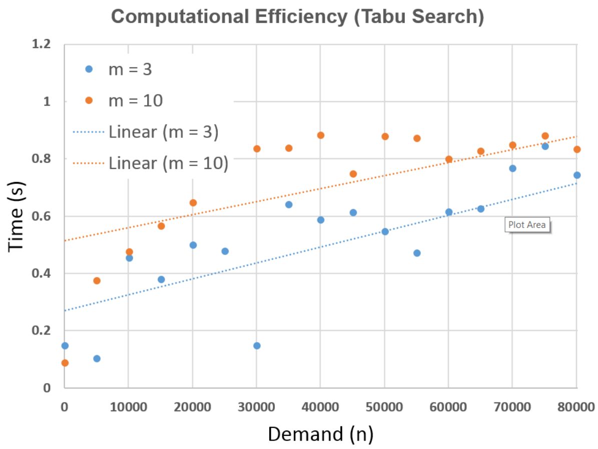
The execution time required to find the solution for the mathematical program [DOF] (4) by Dynamic Programming and Tabu Search are depicted in Figure 2. The results confirm the analysis presented in the methodology section. For high demand, Dynamic Programming method requires much more time than Tabu Search method. Tabu Search is preferred when handling larger networks and higher demands, however Dynamic Programming serves well as a means to validate and tune the parameters used by the Tabu Search.

4.1.3. Comparison of Discrete Optimal [DOF] and Discrete Equilibrium [DEF] Solutions

Tabu Search solutions for optimal flow [DOF] and equilibrium flow [DEF] programs are plotted in Figure 3 for demands from 1000 up to 50000, in increments of 1000. Vehicles initially prefer road 2, while as demand increases road 1 quickly becomes dominant and the percentage of vehicles on the road settles down. The long-term dominance of a particular road i is determined by a combination of its capacity c_i and power p_i.



(a) Dynamic Programming



(b) Tabu Search

Figure 2: Computational complexity.

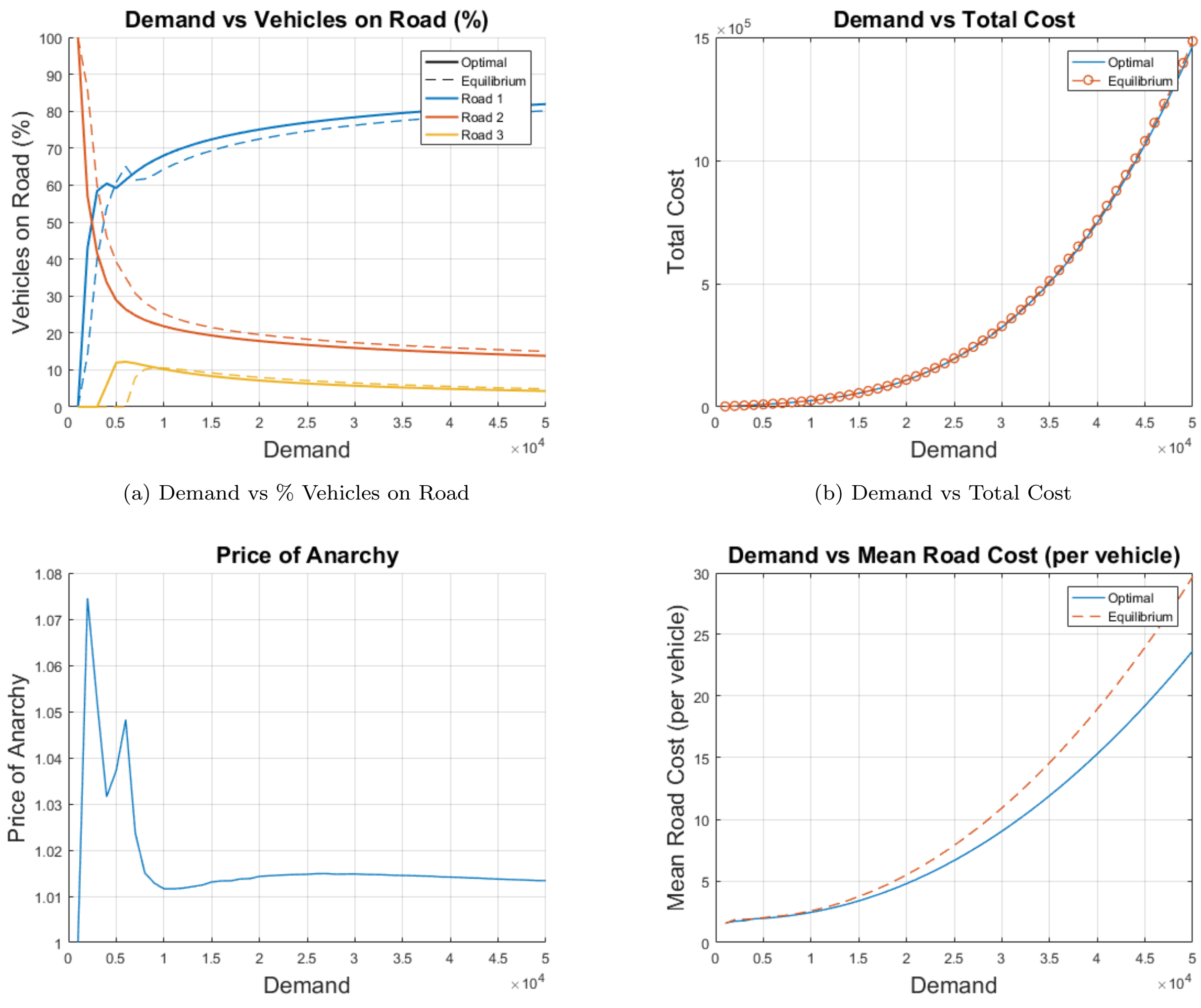
Demand	Method	x_1	x_2	x_3	$f_1(x_1)$	$f_2(x_2)$	$f_3(x_3)$	$\sigma^2(x)$	$T(x) \times 10^5$
1000	DP	0	1000	0	1.85	1.57	2.15	0.085	1566.67
	fmincon	0	1000	0	1.85	1.57	2.15	0.085	1566.67
5000	DP	2953	1444	603	2.00	1.70	2.18	0.057	9677.47
	fmincon	2952.96	1444.48	602.57	2.00	1.70	2.18	0.057	9677.47
10000	DP	6804	2179	1017	2.65	2.19	2.50	0.055	25365.26
	fmincon	6803.76	2178.91	1017.33	2.65	2.19	2.50	0.055	25365.26

Table 3: Optimal [DOF] Integer Solutions (Dynamic Programming) Vs Real [COF] Solutions (*fmincon*).

For low demands, some roads may be empty. As the demand increases, new roads are brought into use, which reflects in spikes of the *Price of Anarchy*. The two spikes in Figure 3 (c) correspond to the introduction of new roads, as shown by Figure 3 (a). It appears that the optimal solution responds to such changes faster than the equilibrium solution. Also, while the total costs between optimal the flow and equilibrium flow solutions shown in Figure 3 (b) are very similar, the mean cost of a road (per vehicle) may differ significantly (difference of about 25% for a demand of $n = 50000$), as suggested by Figure 3 (d).

4.1.4. Discrete 'vs' Continuous Solutions

Table 3 presents solutions of the optimal flow programs [DOF] (4) and [COF] (5), whilst table 4 presents solutions for the equilibrium flow programs [DEF] (7) and [CEF] (8). The continuous solutions for both the optimal flow and equilibrium flow formulations compare well against the discrete solutions. This is as expected due to the convexity of $f_1, \ldots, f_m$.



(a) Demand vs % Vehicles on Road

(b) Demand vs Total Cost

(c) Price of Anarchy

(d) Demand vs Mean Road Cost per Vehicle

Figure 3: An example with three roads.

Demand	Method	x_1	x_2	x_3	$f_1(x_1)$	$f_2(x_2)$	$f_3(x_3)$	$\sigma^2(x)$	$T(x)$
1000	TS	1	998	1	1.85	1.57	2.15	$8.5\ \times 10^{-2}$	1566.67
	fmincon	0.003	999.994	0.003	2.15	1.57	2.15	$8.5\ \times 10^{-2}$	1566.67
5000	TS	3012	1986	2	2.01	2.02	2.15	$6.1\ \times 10^{-3}$	10138.00
	fmincon	2961.50	2038.37	0.13	2.00	2.06	2.15	$5.5\ \times 10^{-3}$	10138.00
10000	TS	6429	2519	1052	2.57	2.57	2.57	$5.5\ \times 10^{-7}$	25665.59
	fmincon	6427.76	2519.73	1052.51	2.57	2.57	2.57	$8.0\ \times 10^{-10}$	25665.59

Table 4: Equilibrium Integer Solutions (Tabu Search) Vs Real Solutions (*fmincon*)

For the equilibrium flow program [CEF] (8) to converge we require a solution to the equilibrium system (6), by Theorem 2.1, we must have $D_0 > \sum_i f_i^{-1}(M_0)$, where $M_0 = \max_i\{f_i(0)\}$. In our example $M_0 = 2.15$, and the equilibrium optimisation problem (8) has a solution if and only if $D > D_0 = 6295.33$.

$$x_1 > f_1^{-1}(2.15) = 4000\sqrt{\frac{2.15-1.85}{0.2775}} = 4159.002; \quad x_2 > f_2^{-1}(2.15) = 1500\sqrt[3]{\frac{2.15-1.5}{0.225}} = 2136.329.$$

Table 5 displays the first solutions for a system of 1, 2 and 3 roads with cost functions given by Table 1. For a small $\varepsilon > 0$ and $D = D_0 + \varepsilon$, the solution is close to $(x_1, x_2, x_3)_{equ} = (4159.002, 2136.329, \varepsilon)$.

System of m roads	x_1	x_2	x_3	$f_1(x_1)$	$f_2(x_2)$	$f_3(x_3)$	D_0
1	-	ε	-	-	1.5	-	ε
2	ε	1738.013	-	1.85	1.85	-	1738.013
3	4159.002	2136.329	ε	2.15	2.15	2.15	6295.33

Table 5: First equilibirum solutions to a system of 1,2 and 3 roads (road cost functions given by Table 1.

4.2. Part B

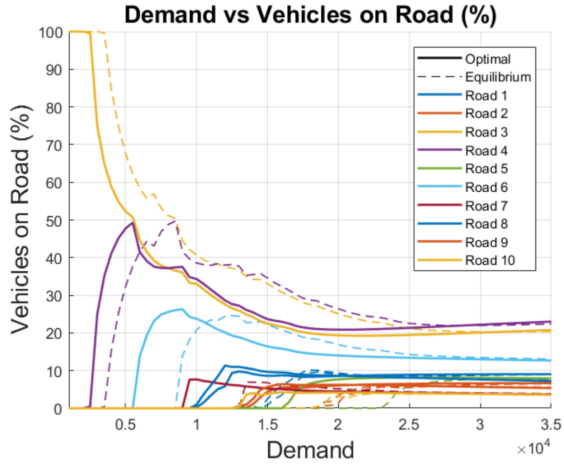
Here we investigate an extended 10 road model, whose parameters are given in Table 6, to compare the variance formulations of the optimal flow [COEF] (12) and equilibrium flow [CEF] (8) with the continuous optimal [COF] (5) and the classic Beckmann formulation for equilibrium [UEF] (19), respectively.

Road Parameters and Road Travel Time per Vehicle					
Road No.	a_i	$b_i = 0.15a_i$	c_i	p_i	$f_i(x_i) = a_i + b_i\left(\frac{x_i}{c_i}\right)^{p_i}$
1	1.2	0.18	2000	5	$f_1(x_1) = 1.2 + 0.18\left(\frac{x_1}{2000}\right)^5$
2	1.3	0.195	1500	5	$f_2(x_2) = 1.3 + 0.195\left(\frac{x_2}{1500}\right)^5$
3	0.8	0.12	3500	4	$f_3(x_3) = 0.8 + 0.12\left(\frac{x_3}{3500}\right)^4$
4	0.9	0.135	4000	4	$f_4(x_4) = 0.9 + 0.135\left(\frac{x_4}{4000}\right)^4$
5	1.4	0.21	2000	6	$f_5(x_5) = 1.4 + 0.21\left(\frac{x_5}{2000}\right)^6$
6	1	0.15	3000	6	$f_6(x_6) = 1 + 0.15\left(\frac{x_6}{3000}\right)^6$
7	1.1	0.165	1000	8	$f_7(x_7) = 1.1 + 0.165\left(\frac{x_7}{1000}\right)^8$
8	1.2	0.18	2000	8	$f_8(x_8) = 1.2 + 0.18\left(\frac{x_8}{2000}\right)^8$
9	1.3	0.195	1500	8	$f_9(x_9) = 1.3 + 0.195\left(\frac{x_9}{1500}\right)^8$
10	1.3	0.195	1000	8	$f_{10}(x_{10}) = 1.3 + 0.195\left(\frac{x_{10}}{1000}\right)^8$

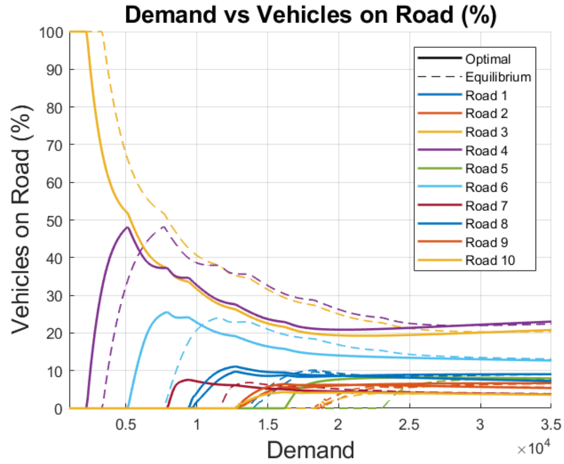
Table 6: Road Cost Functions $f_i(x_i)$ for an example with 10 roads.

4.2.1. Comparisons for the equilibrium mathematical programs

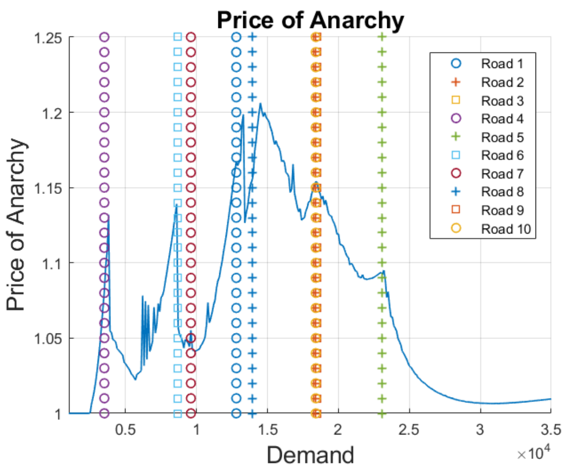
Comparisons between solutions of the optimal flow and equilibrium flow programs are shown in Figure 4. The results are generated using a Tabu Search heuristic at demand intervals of 100.



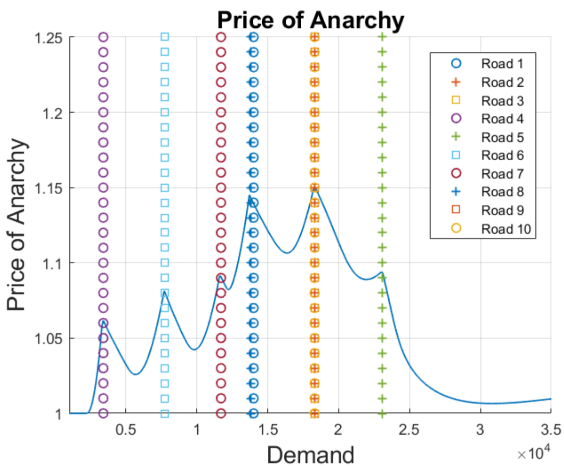
(a) Demand vs % Vehicles ([COEF] (12), [CEF] (8))



(b) Demand vs % Vehicles ([COF] (5), [UEF] (19))



(c) Price of Anarchy ([COEF] (12), [CEF] (8))



(d) Price of Anarchy ([COF] (5), [UEF] (19))

Figure 4: Solution and price of anarchy comparisons for equilibrium program solutions.

Figures 4 (a) and (b) present the percentage of cars (of the total demand) allocated to each road (x_i/n). Notice that once a road starts to be used, the traffic increases to a peak, until another road becomes more attractive. The solutions of the variance formulation compare well against the traditional approach. The price of anarchy is displayed in Figures 4 (c) and (d) and plots vertical lines representing the moment when a new road is used in the equilibrium solution. The noise within the Figure 4 (c) highlights the issues with the methodology at low demand. At high demands, where all roads have flow, the convexity of functions ensures convergence to a unique solution and the *price of anarchy* is identical in both 4 (c) and (d).

5. Concluding remarks

In this paper we presented a simplified traffic model, for which we have formulated various discrete and continuous optimal flow and equilibrium flow mathematical programs. Solutions of these programs were obtained for two scenarios involving 3 and 10 roads respectively, which were solved using exact (exhaustive search, dynamic programming), numerical (interior point methods such as *fmincon*) and heuristic (Tabu Search) techniques. The latter method seemed to be effective in most of the scenarios considered.

We showed that in this parallel-link setup using traditional BRP travel time function, the variance based mathematical programs performed well at high demand, however at low demand convergence was slow due to the inconsistency of the systems (6) and (9). A full investigation on a more complex network topology - one where multiple origin-destination pairs are connected by routes which share certain roads (i.e., Sioux Falls model [17]) - is required to consider how effective this method is compared with traditional methods. Whilst the variance based method may struggle to match their speed and accuracy, it does allow for the possibility of more complex link travel time functions.

Multi-criteria optimization could also be employed as an alternative methodology. Problem (12) may be seen as a particular scalarisation (with equal weights) of the multi-objective optimisation problem

$$\min\{g_i(x) \mid i = 1, \ldots, m\} \text{ subject to } x_1 + \cdots + x_m = n, \quad x_i \in \mathbb{N}. \tag{21}$$

Then every optimal flow solution of program [COEF] (12) is a Pareto optimal solution (i.e., an efficient solution) of this multi-objective optimisation problem (see for instance [18], [19], [20]). Also, the equilibrium flow program [CEF] (8) can be reformulated as a multi-objective optimisation problem, where the criteria $e_1, e_2, \ldots, e_m$ given by $e_i(x) = |f_i(x_i) - \mu(x)|$ are minimised simultaneously.

Acknowledgement. We would like to thank the anonymous referees for their useful comments which helped us improve the paper. Ovidiu Bagdasar's research was partially supported by the grant CNCS/CCCDI UEFISCDI, project number PN-III-P2-2.1-BG-2016-0333, within PNCDI III. Nicolae Popovici's research was supported by a grant of the Romanian Ministry of Research and Innovation, CNCS-UEFISCDI, project number PN-III-P4-ID-PCE-2016-0190, within PNCDI III.

References

[1] J. G. Wardrop, Road paper. some theoretical aspects of road traffic research, Proceedings of the Institution of Civil Engineers 1 (3) (1952) 325–362. doi:10.1680/ipeds.1952.11259.

[2] S. C. Dafermos, F. T. Sparrow, The traffic assignment problem for a general network, Journal of Research of the National Bureau of Standards, Series B 73 (2) (1969) 91–118. doi:10.6028/jres.073B.010.

[3] P. Patriksson, M. Patriksson, P. Patriksson, The traffic assignment problem: models and methods, 1994.

[4] M. Patriksson, A survey on the continuous nonlinear resource allocation problem, European Journal of Operational Research 185 (1) (2008) 1–46. doi:10.1016/j.ejor.2006.12.006.

[5] A. Y. Krylatov, Network flow assignment as a fixed point problem, Journal of Applied and Industrial Mathematics 10 (2) (2016) 243–256. doi:10.1134/S1990478916020095.

[6] J. Cao, R. Li, W. Huang, J. Guo, Y. Wei, Traffic network equilibrium problems with demands uncertainty and capacity constraints of arcs by scalarization approaches, Science China Technological Sciencesdoi:10.1007/s11431-017-9172-4.

[7] W. Huang, W. Jia, J. Guo, B. M. Williams, G. Shi, Y. Wei, J. Cao, Real-time prediction of seasonal heteroscedasticity in vehicular traffic flow series, IEEE Transactions on Intelligent Transportation Systems (99) (2017) 1–11. doi:10.1109/TITS.2017.2774289.

[8] H. Youn, M. T. Gastner, H. Jeong, Price of anarchy in transportation networks: Efficiency and optimality control, Physical Review Letters 101 (12) (2008) 128701. doi:10.1103/PhysRevLett.101.128701.

[9] US Bureau of Public Roads, Traffic assignment manual for application with a large, high speed computer., U.S. Dept. of Commerce, Bureau of Public Roads, Office of Planning, Urban Planning Division, Washington, 1964.
URL http://catalog.hathitrust.org/Record/000968330

[10] M. Minderhoud, H. Botma, P. Bovy, Assessment of Roadway Capacity Estimation Methods, Transportation Research Record: Journal of the Transportation Research Board 1572 (1997) 59–67. doi:10.3141/1572-08.

[11] S. Berry, V. Lowndes, M. Trovati, Guide to Computational Modelling for Decision Processes: Theory, Algorithms, Techniques and Applications, 2017. doi:10.1007/978-3-319-55417-4.

[12] V. Lowndes, S. Berry, C. Parkes, O. Bagdasar, N. Popovici, Further use of heuristic methods, in: Guide to Computational Modelling for Decision Processes, Springer, 2017, pp. 199–235. doi:10.1007/978-3-319-55417-4_7.

[13] R. Bellman, Dynamic programming, Dover Publications, 2003.

[14] M. S. Bazaraa, H. D. Sherali, C. M. Shetty, Nonlinear Programming: Theory and Algorithms, Wiley-Interscience, 2006.

[15] E. Koutsoupias, C. Papadimitriou, Worst-case equilibria, Computer Science Review 3 (2) (2009) 65–69. doi:10.1016/j.cosrev.2009.04.003.

[16] T. Roughgarden, Selfish routing and the price of anarchy, Vol. 174, MIT press Cambridge, 2005.

[17] H. Bar-Gera, F. Hellman, M. Patriksson, Computational precision of traffic equilibria sensitivities in automatic network design and road pricing, Transportation Research Part B: Methodological 57 (2013) 485–500. doi:10.1016/j.sbspro.2013.05.005.

[18] J. Jahn, Mathematical Vector Optimization in Partially Ordered Linear Spaces, Lang, 1986.

[19] N. Popovici, Pareto reducible multicriteria optimization problems, Optimization 54 (3) (2005) 253–263. doi:10.1080/02331930500096213.

[20] O. Bagdasar, N. Popovici, Local maximizers of generalized convex vector-valued functions, Journal of Nonlinear Convex Analysis 18 (12) (2017) 2229–2250.