# Static Traffic Assignment with Junction Modelling


VERSIE VOOR NIET-WISKUNDIGEN MET EXTRA UITLEG

Graduation Thesis

Heleen Muijlwijk

March 2012

Supervisors:

Dr. G.J. Still
Prof. Dr. M.J. Uetz
Dr. Ir. W.R.W. Scheinhardt

Ir. M.P. Schilpzand

UNIVERSITY OF TWENTE.

OmniTRANS
Transport Planning Software

## TABLE OF CONTENTS

# PREFACE

With this graduation thesis I finish my master's degree, which was a combination of mathematics, psychology and teacher education. This final research project was a valuable experience to me. It was my first experience in business, which complemented to my experience as a teacher. Omnitrans International provided me an interesting research topic and was a great workplace. I have learned I like the role of a mathematician in a company.

Within this research project, I tried to make a connection between theory and practice. I used mathematical theory to explain the behaviour of practical transport modelling software. Mathematics was the fundamental to my research, but I had to understand both the theoretical and the practical perspective to draw the right conclusions. The tension between theory and practice was often notable. In a conversation with my supervisor of the university, naturally a theorist, he said to me: "At this point we mathematicians usually stop. The function does not exist, so this is it. But when practice necessarily wants to use it, we can only say: be careful!". Although it was a tough job, I enjoyed it to 'translate' between theory and practice.

I am proud of this final thesis, but I could not have done it alone. I want to thank in the first place all my supervisors. Georg Still and Marc Uetz, always 'side by side', for their great enthusiasm and patience. Maarten Schilpzand, for the weekly meetings and a lot of fun at Omnitrans International. And Werner Scheinhardt, for reading my final work. Also I want to thank my colleagues at Omnitrans International, especially Michiel Bliemer, Edwin, Wim, Feike, Mike and Peter, for always being helpful and eager to share and explain. I owe many thanks to Michiel Rutjes, my parents, my brother Ronald, my aunt José, and my friends for always supporting me, asking the right questions and slowing me down when necessary. Finally, I want to thank Gerard Rutjes for revising my thesis.

For any questions or comments on this thesis, with respect to further research or for other purposes, please feel free to contact me.

In this version I have added extra explanation for non-mathematicians. These green texts, which are written in Dutch, can be skipped without loss of continuity.

Welkom niet-wiskundigen, in mijn verslag! Deze groene teksten zul je her en der door het verslag heen vinden, en bevat korte extra uitleg in simpele taal. Ik hoop met deze aanvulling het verslag toegankelijk te maken voor mensen die geen wiskunde hebben gestudeerd. Als je je alleen beperkt tot het lezen van deze groene teksten, krijg je in vogelvlucht een indruk van mijn afstudeeronderzoek.

# 1 INTRODUCTION

## 1.1 BACKGROUND

Modelling traffic can have many purposes, for example predicting the effects on a road network when building a new residential area, predicting the effects of construction work, or visualizing congestion in the morning and evening peaks.

Modelling traffic consists roughly of four steps, this is known as the 'four-step model'. The first step is 'trip generation'. It determines the frequency of trips that are going in and out of zones as a function of socio-economic data. For example, a zone containing a lot of shops will generate many trips going into that zone. The second step is 'trip distribution'. It matches origins and destinations, in such a way that it determines how much travellers will be travelling from a specific origin to a specific destination. Thus a trip matrix is obtained. Often a gravity model is used in this step, based on the fact that masses attract each other: the bigger the mass (higher frequency of trips) and the smaller the distance between the masses (smaller distance between origins and destinations), the bigger the attraction (more trips are made). The third step is 'modal split', where the trips are assigned to different modes, for example cars, bicycles or public transport. The fourth and final step is the traffic assignment. It determines which routes will be chosen by travellers, given their origins and destinations. In this step the travellers are 'placed' on the network, and a resulting load on every road is obtained. This last step, known as the Traffic Assignment Problem (TAP), is the subject of this study.

Dit onderzoek gaat over het modelleren van verkeer. Verkeersmodellen worden onder andere gebruikt door gemeentes, om de invloed te onderzoeken op een verkeersnetwerk van bijvoorbeeld een nieuwe brug, een extra woonwijk, of wegwerkzaamheden. Het modelleren van verkeer bestaat uit verschillende stappen. Onder andere moet berekend worden hoeveel verkeer er van een bepaalde herkomst naar bestemming wil. Dit onderzoek gaat over de laatste stap in de verkeersmodellering, namelijk de toedeling. In deze stap wordt berekend welke route reizigers zullen kiezen, en hoe de uiteindelijke verkeersstromen zullen lopen.

The assignment is based on some assumptions on the network. Firstly, we can assume congestion in the network, meaning that the travel time increases when it gets busier. If we ignore congestion, the travel time is the same, regardless how busy it is. Secondly, we can assume that travellers have perfect knowledge about the network and the travel times, so everyone knows what the shortest routes are and agrees about that. In this case we deal with a deterministic model. Or we can assume that travellers may have different perceptions about the network and their travel times, this is a stochastic model.

The four different categories of 'equilibrium-based' assignment methods are shown in Table 1. This study concerns deterministic models, and we assume congestion in the network, so equilibrium assignment methods are the subject of this study.

**TABLE 1: CATEGORIES OF ASSIGNMENT METHODS**

| | Deterministic | Stochastic |
|---|---|---|
| Without congestion | All-Or-Nothing | Pure stochastic methods |
| With congestion | Equilibrium methods | Stochastic equilibrium methods |

The TAP can be applied on either a static or a dynamic traffic models. In this study we consider static models. In static models the demand is a fixed value of travellers per time unit, for example vehicles per hour, and time does not play a role. Contrary, in dynamic models the demand, and also the load, is a function of time. Summarizing, this study concerns a static, user equilibrium-based TAP with deterministic route choice.

Er zijn allerlei manieren om verkeer te modelleren, gebaseerd op verschillende aannames. Dit onderzoek richt zich op statisch modellen, dat betekent dat de tijd geen rol speelt. Files kunnen dus niet ontstaan of oplossen, ze zijn er gewoon. De verkeerstromen worden uitgedrukt in voertuigen per uur, en dat is het. Ook wordt er vanuit gegaan dat alle reizigers precies weten wat de korste route is, en die ook nemen. Zelfs als er een route is die één seconde sneller is, wordt die gekozen. Verder gaan we ervan uit dat er een 'gebruikersevenwicht' ontstaat, dat is een situatie waarin niemand zijn reistijd nog kan verkorten (anders had hij dat wel gedaan). Tot slot gaan we uit van 'congestie': hoe drukker het is op de weg, hoe langer de reis zal duren.

Beckmann, McGuire and Winsten (1956) stated the Traffic Assignment Problem as an optimization problem, and the first algorithm that solves the TAP was developed also in 1956 (Frank & Wolfe, 1956). This algorithm, the Frank-Wolfe (FW) algorithm, is traditionally the standard way of solving the TAP. Later on, especially in the last decade, several other algorithms have been developed. Some of them are improvements of the FW algorithm, but also new algorithms have been developed based on paths or bushes instead of links.

This study is done on behalf of Omnitrans International. This company makes transport planning software called OmniTRANS. They use the FW algorithm (and some variations) for the assignment, and they want to explore other possibilities for the assignment in their software. However, OmniTRANS contains an extensive junction modelling module, which makes the assignment of traffic more complicated.

Dit onderzoek is gedaan in opdracht van Omnitrans International. Zij maken software om verkeer te modelleren, genaamd OmniTRANS. Zij gebruiken een standaard algoritme (oplossingsmethode) voor de toedeling, maar ze zijn benieuwd of dit sneller en beter kan, vooral omdat er de laatste jaren veel nieuwe algoritmes voor de toedeling zijn ontwikkeld. Echter, het is niet zeker dat zij zomaar alle algoritmes kunnen gebruiken, omdat zij een uitgebreide kruispuntmodellering in hun verkeersmodellen hebben, wat de toedeling moeilijker kan maken.

This leads to the following research question.

## 1.2 Research question

***In the static user equilibrium-based Traffic Assignment Problem with deterministic route choice, expanded with junction delays, which algorithm converges the fastest[1] to an accurate[2] solution, within limited memory capacity?***

[1] in terms of calculations time

[2] the solution must be unique, stable, path-based, and the duality gap must be small enough ($<10^{-6}$)

### Sub questions

1. a) What is the standard TAP?
   b) What is the influence of the addition of junction delays on the standard TAP?

c) When does the TAP with junction delays have a unique solution? What assumptions may be needed to make it have a unique solution?

2 What are the requirements of the assignment algorithm
   a) resulting from the addition of junction delays?
   b) resulting from the junction modelling in OmniTRANS and the practical implementation of the algorithm in OmniTRANS?
3 What are the possible algorithms for solving the standard TAP, including the developments in the last decade? Are these algorithms capable to cope with junction delays?
4 Which (existing or new) algorithm is the best way for solving the TAP with junction delays, in OmniTRANS?

Dit heeft geleid tot de volgende onderzoeksvraag: 'Voor de toedeling in verkeersmodellen met kruispuntmodellering, welk algoritme werkt goed (geeft een juiste oplossing) en werkt het snelst (zodat je computer geen dagen aan het rekenen is)?' Voordat deze vraag beantwoord kan worden, moeten er eerst wat deelvragen beantwoord worden, zoals: 'Welke invloed heeft de kruispuntmodellering eigenlijk op het model en op de toedeling?' en 'Waar moet het algoritme dus aan voldoen?'.

## 1.1 Organization of the Thesis

The remainder of this thesis is organized as follows. In Chapter 2 the problem is formulated. First the mathematical formulations of the TAP are given, then the influence of junction delays on the TAP is discussed. Also the existence and uniqueness of solutions are considered. We will zoom in on junction modelling in OmniTRANS. Chapter 2 concludes with a final problem formulation.

In Chapter 3 the current assignment methods in OmniTRANS are discussed. In Chapter 4 their limitations with respect to junction modelling are discussed and some adaptations are suggested. In Chapter 5 new possible solution methods are proposed. Finally, in Chapter 6 and 7 the conclusions and a discussion is given.

# 2 PROBLEM FORMULATION

## 2.1 NOTATIONS OF TRANSPORTATION NETWORK

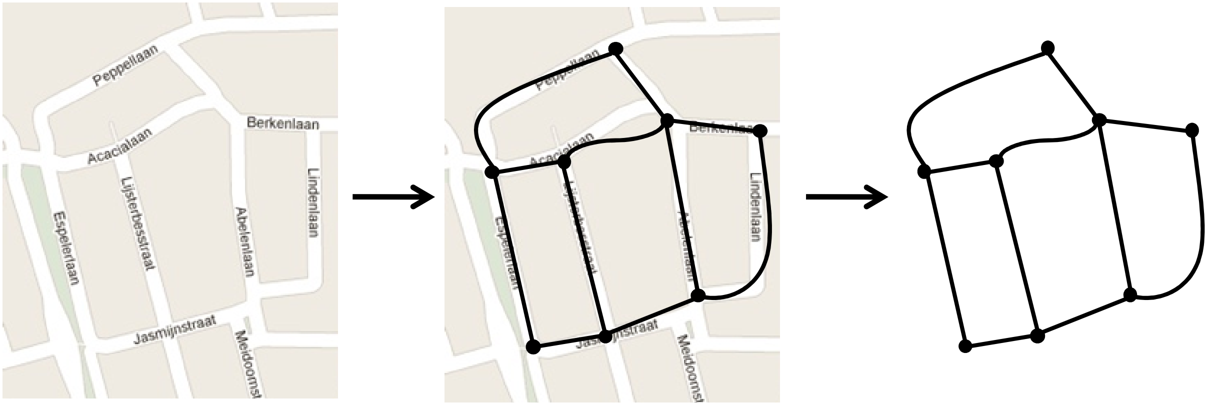



**FIGURE 1: TRANSPORTATION NETWORK REPRESENTED AS A GRAPH**

For studying the TAP, we need to introduce some notation. We consider a graph $G = (E, A)$ representing a transportation network, with nodes and links. The links represent roads, the set of links is denoted as $A$ and a link as $a$. The nodes represent intersections of links. The set of nodes are denoted as $E$ and a node as $e$. On the nodes we can define junctions. For a detailed explanation of junctions in the model, see Section 2.5. The traffic flow on the network is represented as load on the links, in vehicles per hour, denoted by $x_a$. The cost of travelling experienced by the user is dominated by travel time. In our model other variables, such as distance and toll, are ignored, and we set the travel costs equal to travel time. We consider congested networks, this means when it gets busy on a road, the travel time increases. Therefore the travel time on link $a$ is a monotonically increasing function $c_a(x_a)$, which we call the cost function of link $a$. Later on, we will generalize the model by setting cost functions to $c_a(x)$, where $x$ is a vector of all link loads $x = (x_a, a \in A)$.

Een verkeersmodel is natuurlijk gebaseerd op een echt verkeersnetwerk, met wegen en kruispunten. We geven dit vereenvoudigd weer, de kruispunten zijn punten, daartussen lopen lijnen die de wegen weergeven, zie Figuur 1. Aan die lijnen (wegen) 'hangen' we een kostenfunctie, die berekent de reistijd, geven de hoeveelheid verkeer op de weg. Deze kostenfuncties zullen een centrale rol spelen in dit onderzoek.

There are several ways to define the cost function $c_a(x_a)$. The conventional approach is the Bureau of Public Roads (BPR) function:

$$c_a(x_a) = \frac{L_a}{v_a^{\max}}\left(1 + \alpha_a \left(\frac{x_a}{q_a}\right)^{\beta_a}\right), \tag{2.1}$$

where

$L_a$ is the length of link $a$;
$v_a^{\max}$ is the maximum speed on link $a$;
$x_a$ is the load on link $a$;
$q_a$ is the capacity of link $a$ and
$\alpha_a$ and $\beta_a$ are constants defined for every link.

Note that $\frac{L_a}{v_a^{\max}}$ corresponds to the 'free flow travel time', that is the travel time when there is no load on a link. $\beta$ is usually set on 4. The value of $\alpha$ depends on the type of road. For highways a low value of $\alpha$ is used, for example $\alpha = 0.5$, for urban roads usually a higher value of $\alpha$ is used, for example $\alpha = 2$. In Figure 2, the graphs of two BPR functions are shown.

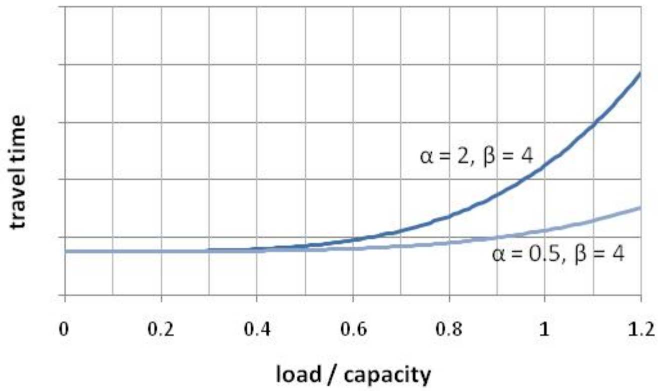



**FIGURE 2: GRAPH OF BPR-FUNCTION**

Furthermore, notation with respect to paths, origins and destinations is needed. Travellers begin their journey at their origin, denoted by $r$, and end at their destination, denoted by $s$. The set of all origins is denoted by $O$ and the set of all destinations by $D$, and $rs$ is called an $OD$-pair. The set of all $OD$-pairs is denoted as $OD$. The demand is the amount of travellers that wants to travel from $r$ to $s$, and is denoted by $d_{rs}$. A path from $r$ to $s$ is denoted by $k_{rs}$, the set of all paths from $r$ to $s$ is denoted by $K_{rs}$. All paths are in the set $K$. Load is not solely defined on links, but also on paths. The relation between the link loads $x_a$ and path loads $f_k^{rs}$ is given by the link-path incidence relationship

$$x_a = \sum_{r,s} \sum_{k \in K_{rs}} \delta_{a,k}^{rs} f_k^{rs} , \qquad (2.2)$$

where

$$\delta_{a,k}^{rs} = \begin{cases} 1, \text{ if link } a \text{ is on path } k \text{ connecting } r \text{ and } s; \\ 0, \text{ otherwise and} \end{cases}$$

$f_k^{rs}$ is the load on a path $k$ connecting $r$ and $s$.

All notations used in this thesis are also explained in Appendix I.

## 2.2 MATHEMATICAL FORMULATIONS OF THE TAP

When choosing a specific TAP, one has to quantify a goal. That is, state some characteristics of the final flow on the road network. In traffic modelling, we want to assign traffic in a way that it approximates reality. We assume that in a realistic situation every traveller behaves independently and seeks to minimize his own travel time. Wardrop (1952) stated in his 'first principal' that this leads to User Equilibrium (UE), which means that traffic will distribute over the network in a way that no individual can decrease his travel time by changing his route. So our goal is to assign traffic such that UE is obtained.

We gaan ervan uit dat we in de verkeersmodellen een 'gebruikersevenwicht' moeten krijgen, een situatie waarbij niemand zijn eigen reistijd nog kan verkleinen. Wardrop heeft in 1952 al gezegd dat dit een realistische situatie is. We willen dat het verkeersmodel een realistisch verkeerspatroon geeft, dus we nemen gebruikersevenwicht als doel van de toedeling. In de formules hieronder wordt beschreven hoe je kunt controleren of de situatie een gebruikersevenwicht is. Eén van de dingen die bijvoorbeeld moet gelden is: als er verkeer over een route gaat, dan moet die route wel een optimale reistijd hebben, anders had niemand die route gekozen. En als er geen verkeer over een route gaat, dan moet die minstens zo lang duren als de optimale route, anders zou iemand die route wel gekozen hebben. Ook zijn er nog wat vanzelfsprekende eisen, zoals: er mag geen negatieve hoeveelheid verkeer over een route rijden.

UE is obtained, when a path flow solution $\bar{f} = (\bar{f}_k^{rs}, k \in K_{rs}, rs \in OD)$, satisfies the following Wardrop equilibrium conditions:

$$\bar{f}_k^{rs}\left(c_k^{rs}(\bar{f}) - \pi^{rs}\right) = 0\,, \qquad \forall k \in K_{rs}, \quad \forall rs; \tag{2.3}$$

$$c_k^{rs}(\bar{f}) \geq \pi^{rs}\,, \qquad \forall k \in K_{rs}, \quad \forall rs; \tag{2.4}$$

$$\textstyle\sum_k \bar{f}_k^{rs} = d_{rs}\,, \qquad \forall k \in K_{rs}, \quad \forall rs; \tag{2.5}$$

$$\bar{f}_k^{rs} \geq 0, \;\; c_k^{rs} \geq 0, \qquad \forall k \in K_{rs}, \quad \forall rs, \tag{2.6}$$

where $\pi^{rs}$ is the optimal travel time from $r$ to $s$. This is also known as the complementarity problem.

Equation (2.3) has to hold at equilibrium because either a route has no load on it: $\bar{f}_k^{rs} = 0$, or there is load on a route and the travel time of that route equals the optimal travel time: $c_k(\bar{f}) - \pi_{rs} = 0$. Equation (2.4) holds at equilibrium because all routes from $r$ to $s$ have either the optimal travel time or the travel time is longer. Equation (2.5) concerns flow conservation, it ensures that the total flow from $r$ to $s$ always meets the demand. Finally non-negativity constraints (2.6) have to hold.

In the remainder of this section two mathematical formulations of the TAP are given, first as an optimization problem, then as a Variational Inequality Problem. Also, proofs are given that these formulations are equivalent, and that solutions to both formulations are UE flows.

### 2.2.1 OPTIMIZATION PROBLEM

Beckmann et al. (1956) was the first who formulated the TAP as an optimization problem. He formulated this for two purposes, the one given below aims to find flow patterns at UE. The other formulation aims to find System Optimum (SO). The difference between UE and SO will be explained in an example in Section 2.2.1.2.

The UE-based 'Beckmann formulation' of the TAP is as follows.

$$\min \quad z(x) = \sum_a \int_0^{x_a} c_a(\omega)d\omega\,, \tag{2.7}$$

$$\text{subject to} \quad \sum_k f_k^{rs} = d_{rs}\,, \qquad \forall k \in K_{rs}, \quad \forall rs; \tag{2.8}$$

$$f_k^{rs} \geq 0\,, \qquad \forall k \in K_{rs}, \quad \forall rs; \tag{2.9}$$

$$x_a = \sum_{r,s,k} \delta_{a,k}^{rs} f_k^{rs}\,, \quad \forall a. \tag{2.10}$$

This optimization problem has an objective function $z(x)$, see equation (2.7), which is nonlinear and convex, and the constraints are linear. Constraint (2.8) is the flow conservation constraint. Constraint (2.9) ensures that the load is positive. The link-path incidence relationship is added as a constraint (2.10). A solution $x$ is a certain traffic flow pattern, and $x$ is said to be feasible if it meets the constraints. The set of all feasible $x$ form the feasible region, which is a polyhedron on a hyper plane. The optimal solution $\bar{x}$, which minimizes the objective function, is the flow pattern at user equilibrium. Actually, a solution to the Beckmann formulation is both an $x$ and an $f$. We simply call $\bar{x}$ user equilibrium when $(\bar{x}, \bar{f})$ satisfies (2.8) – (2.10) and $\bar{f}$ satisfies the Wardrop equilibrium conditions as stated in equation (2.3) – (2.6). Why $\bar{x}$ is user equilibrium will be explained with an example in Section 2.2.1.1, and the proof will be given in Section 2.2.3.

Zo'n toedeling, hoe doe je dat eigenlijk? Hoe beschrijf je het als een opgave, of zoals wiskundigen zeggen liever 'probleem', en hoe los je het op? Beckmann beschreef het probleem in 1956 als een optimalisatie probleem. Hij gaf daarmee gelijk ook een hint voor het oplossen: optimaliseren zou moeten werken. Het optimalisatie probleem stelt dat er allerlei oplossingen zijn, maar dat er één optimale oplossing is. Met een oplossing bedoelen we een bepaald verkeerspatroon, bijvoorbeeld zeven mensen rijden linksom en twee mensen rijdt rechtsom, of ze rijden alle negen rechtsom. Er zijn vaak meerdere oplossingen die 'kloppen', dat betekent dat er aan eisen wordt voldaan zoals: iedereen komt vanuit zijn herkomst aan op zijn bestemming, er is geen negatieve hoeveelheid verkeer, etc. Maar welke van die 'kloppende' oplossingen is nu het gebruikersevenwicht waar we naar op zoek zijn? Die oplossing kun je krijgen door een bepaalde functie (vergelijking (2.5)) te minimaliseren. We noemen deze functie de doelfunctie. In deze functie wordt de oppervlakte onder de grafieken van de kostenfuncties gebruikt (zie Figuur 4). Om precies te zijn, als je de som van de oppervlakte onder de grafieken minimaliseert, dan vind je een situatie waarbij reistijden gelijk (en dus optimaal) zijn, en dat is het gebruikersevenwicht.

The objective function $z(x)$ in the Beckmann formulation above is expressed in link flows, but via the link-path incidence relationship $x_a = \sum_{r,s,k} \delta_{a,k}^{rs} f_k^{rs}$, also a path flow solution can be obtained. Note that the path flow solution is not necessarily unique. The relation between link flows and path flows will be discussed in Section 2.6.

#### 2.2.1.1 BECKMANN FORMULATION AND USER EQUILIBRIUM

In this example it is explained why minimizing the objective function $z(x) = \sum_a \int_0^{x_a} c_a(\omega)d\omega$ of the Beckmann formulation yields user equilibrium.

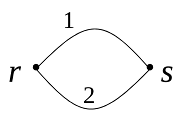



**FIGURE 3: EXAMPLE NETWORK**

Let us consider the case of a simple network shown in Figure 3, with one origin $r$, one destination $s$, and two links between them. Assume the cost function of the links are: $c_1 = 10 + 3x_1$ and $c_2 = 15 + 2x_2$.

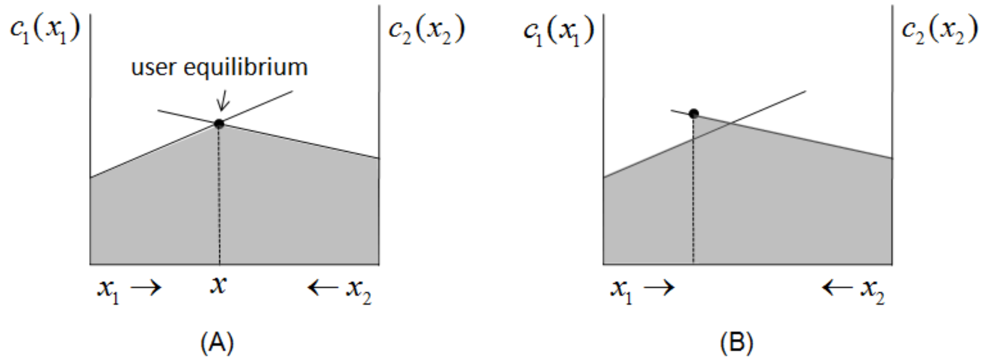



**FIGURE 4: BECKMANN TRANSFORMATION AND USER EQUILIBRIUM**

In Figure 4(A) the two cost functions are shown in one coordinate system. The load is shown on the $x$-axis, where a point on the $x$-axis is corresponding to a flow pattern. The sum of all loads meets the demand, that is $x_1 + x_2 = d_{rs}$. The load $x_1$ is shown from left to right, and the load $x_2$ is shown from right to left. For example, the left hand side of the x-axis corresponds to the case where all flow is send through link 2, and nothing through link 1. The cost functions are plotted. At user equilibrium the travel time on all used paths must be equal. This is at the intersection of the cost functions, marked as 'user equilibrium' in the figure. This intersection (and the corresponding flow pattern) is obtained by minimizing the surface under the graphs. Figure 4(B) shows that choosing a different flow pattern (represented by another value on the x-axis), yields a higher value of the surface under the graphs. Therefore, the minimization of the surface yields user equilibrium.

A formal proof of the equivalence of the Beckmann formulation and UE is given in Section 2.2.3.

#### 2.2.1.2 USER EQUILIBRIUM VERSUS SYSTEM OPTIMUM

The Beckmann formulation of the TAP can be formulated for two purposes, to yield either a System Optimum (SO-based TAP) or a User Equilibrium (UE-based TAP). In the SO-based TAP the aim is to minimize the sum of all travel times, the solution is 'optimal for the system'. Still, it is not necessarily optimal to an individual. In this example the difference between the SO and UE is explained.

Given the network presented in Figure 3, with demand $d_{rs} = 12$. First consider the UE solution. Substituting the cost functions in the Beckmann formulation yields to

$$\begin{aligned} \min \quad z(x) &= \textstyle\sum_a \int_0^{x_a} c_a(\omega)d\omega \\ &= \int_0^{x_1}(10 + 3\omega)\, d\omega + \int_0^{x_2}(15 + 2\omega)\, d\omega \\ &= 10x_1 + \tfrac{3}{2}{x_1}^2 + 15x_2 + {x_2}^2 , \end{aligned}$$

$$\text{subject to} \quad x_1 + x_2 = 12 ;$$

$$x_a \geq 0 , \qquad \forall a.$$

Substituting $x_2 = 12 - x_1$ in the above formulation yields to

$$\min \quad z(x) = 10x_1 + \frac{3}{2}{x_1}^2 + 15(12 - x_1) + (12 - x_1)^2, \quad x_a \geq 0, \quad \forall a.$$

Differentiating $z(x)$ and equate to zero, leads to the solution $\bar{x}_1 = 5.8$ and $\bar{x}_2 = 6.2$. Since this flow pattern is User Equilibrium, the travel time on both paths should be the same. We can check this, on both paths the travel time is $10 + 3 \cdot 5.8 = 15 + 2 \cdot 6.2 = 27.4$. For comparison with the System Optimum described below, note that the total travel time is $\bar{x}_1 c_1 + \bar{x}_2 c_2 = 5.8(10 + 3 \cdot 5.8) + 6.2(15 + 2 \cdot 6.2) = 328.8$.

Now let's consider the System Optimum in this network. In the System Optimum, the aim is to minimize the total travel time, so the optimization problem becomes

$$\begin{aligned} \min \quad z(x) &= \textstyle\sum_a c_a(x_a) \cdot x_a \\ &= x_1(10 + 3 \cdot x_1) + x_2(15 + 2 \cdot x_2) \\ &= 10x_1 + 3{x_1}^2 + 15x_2 + 2{x_2}^2, \\ \text{subject to} \quad & x_1 + x_2 = 12; \\ & x_a \geq 0, \quad \forall a. \end{aligned}$$

Substituting $x_2 = 12 - x_1$ in the above formulation yields to

$$\min \quad z(x) = 10x_1 + 3{x_1}^2 + 15(12 - x_1) + 2(12 - x_1)^2, \quad x_a \geq 0, \quad \forall a.$$

Differentiating $z(x)$ and equate to zero, leads to the solution $\bar{x}_1 = 5.3$ and $\bar{x}_2 = 6.7$. Note that the travel times on both paths are different: $c_1 = 10 + 3 \cdot 5.3 = 25.9$ and $c_2 = 15 + 2 \cdot 6.7 = 28.4$. A traveler at path 2 could feel disadvantaged, knowing that travelling along path 1 is faster. Still, the total travel time of the system is $z(\bar{x}) = \bar{x}_1 c_1 + \bar{x}_2 c_2 = 5.3(10 + 3 \cdot 5.3) + 6.7(15 + 2 \cdot 6.7) = 327.55$, and that is smaller than the total travel time in the User Equilibrium (where $x_1 c_1 + x_2 c_2 = 328.8$).

Summarizing, optimizing group performance usually results in different flow patterns and travel times than optimizing individual performance. It depends on the goal which model is used.

### 2.2.2 Variational Inequality Problem

The Beckmann formulation of the TAP as discussed above, deals with costs $c_a = c_a(x_a)$ solely depending on the load on link $a$. Generalizing, we can imagine cases where there is interaction of traffic between links, so the cost of a link also depends on load on other links in the network, for example at junctions or with two-way traffic. Then the cost function becomes non-separable, that is, $c_a = c_a(x)$ where $x$ is a vector of all link loads $x = (x_a, a \in A)$. In other words, the Jacobian of $c(x)$ has non-zero off-diagonal entries. When junctions are modelled, some cost functions become non-separable, because the cost function of a turn also depends on the load on an conflicting turn.

The Beckmann formulation can handle with non-separable costs, but only when the cost functions are symmetric, that is $\frac{\partial c_i}{\partial x_j} = \frac{\partial c_j}{\partial x_i}, \forall ij$. This means that cost of link $i$ is influenced by the load on link $j$ in the same way as the cost of link $j$ is influenced by the load on link $i$. When the costs are asymmetric, the Beckmann formulation cannot deal with the problem, because the objective function does no longer exist. See Textbox 1 below for the argumentation. Therefore, we have to switch to a generalized formulation of the optimization problem, namely the Variational Inequality Problem.

Mooi, dat optimalisatieprobleem van Beckmann, maar het blijkt niet altijd een toerijkende omschrijving. De doelfunctie die geminimaliseerd moet worden gebruikt de oppervlakte onder een grafiek, en die krijgen we door te integreren. Helaas kunnen we niet alle functies integreren, en dat is geen onvermogen, maar de integraal bestáát soms gewoon niet.
De functies die we zouden moeten integreren zijn de kostenfuncties, die voor elke weg een reistijd berekenen. Normaalgesproken is dat geen probleem, deze functies kunnen we integreren en we kunnen dus het optimalisatieprobleem van Beckmann gebruiken. Maar als de reistijd niet alleen bepaald wordt door het verkeer op de weg zelf, maar ook beïnvloed wordt door verkeer op andere wegen, zoals het geval is op kruispunten, dan zou dit wel eens problematisch kunnen worden... De kostenfuncties worden dan namelijk 'niet-seperabel', en als ze ook nog eens asymmetrisch zijn dan is integreren onmogelijk. Wat asymmetrie van een functie precies is is niet belangrijk voor het verhaal, zie het als een willekeurig kenmerk van de kostenfunctie.
Als integreren onmogelijk is bestaat de doelfunctie niet meer, en in dat geval zouden we over moeten stappen naar een algemenere formulering van de toedeling, namelijk de 'variationele ongelijkheid'. Dafermos heeft in 1980 als eerste laten zien dat de toedeling ook zo omschreven kan worden. Die beschrijving van de toedeling als variationele ongelijkheid is lekker kort door de bocht: 'Voldoet je oplossing aan deze ongelijkheid? Dan heb je de optimale oplossing gevonden!'. Zie vergelijking (2.12).

**TEXTBOX 1: BECKMANN AND ASYMMETRICAL COSTS**

In the objective function of the Beckmann formulation we look for a function $z(x)$ such that

$$\nabla z(x) = c(x).$$

***Theorem***
Let $z(x)$ be a twice continuously differentiable function. $z(x)$ exists if and only if $\nabla c(x)$ is symmetric.

***Proof***
First we will proof: If $z(x)$ exists → if $\nabla c(x)$ is symmetric.

If $z(x)$ exists then the following equation has to hold, obtained by differentiating both sides:

$$\nabla^2 z(x) = \nabla c(x).$$

$\nabla^2 z(x)$ is symmetric, because every Hessian is symmetric. Therefore, also $\nabla c(x)$ has to be symmetric.

Second we have to proof: if $\nabla c(x)$ is symmetric → $z(x)$ exists.

For this proof, we refer to 'Poincaré's Lemma'.

■

Dafermos (1980) and Smith (1983a) showed that the Traffic Assignment Problem can also be formulated as a Variational Inequality Problem (VIP), and that the VIP can deal with non-separable asymmetric cost functions.

In general, a VIP seeks a feasible solution $\bar{x} \geq 0$ such that the following Variational Inequality (VI) holds:

$$\nabla f(\bar{x})(x - \bar{x}) \geq 0\,, \qquad \forall x \in \text{feasible set.} \tag{2.11}$$

The TAP formulated as a Variational Inequality (VI) is: seeks a feasible solution $\bar{x} \geq 0$ such that

$$c(\bar{x})^T(x - \bar{x}) \geq 0\,, \qquad \forall x \in \text{feasible set,} \tag{2.12}$$

where $c(x)$ is a vector of all cost functions $c(x) = (c_a(x), \forall a \in A)$. The solution $\bar{x}$ is the optimal flow pattern, and corresponds to UE.

In the case of symmetric cost functions, it is proven that $\bar{x}$ is the solution of the VI if and only if $\bar{x}$ is the optimal solution of the Beckmann formulation. We will show this in the next section.

### 2.2.3 EQUIVALENCE PROBLEM FORMULATIONS AND USER EQUILIBRIUM

In this section a proof is given about the equivalence of the Beckmann transformation, the Variational Inequality formulation of the TAP and UE. An overview of the theorems is shown in Figure 5.

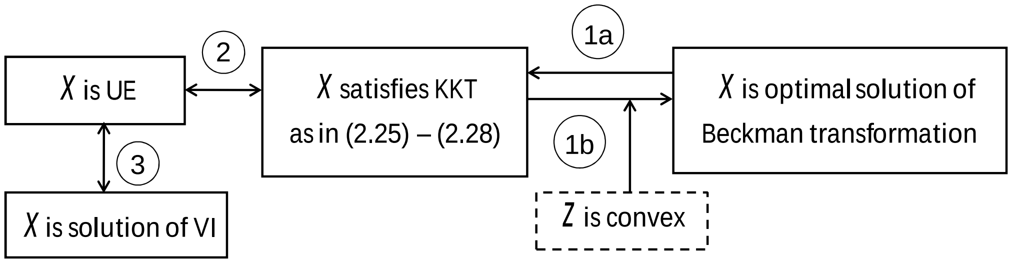



**FIGURE 5: OVERVIEW THEOREMS**

By introducing some new notation,

$\Lambda$ is the path-OD matrix, where an element $\Lambda_{rs,k} = \begin{cases} 1, \text{ if path } k \in K_{rs}; \\ 0, \text{ otherwise;} \end{cases}$

$\Delta$ is the link-path incidence matrix, where an element $\delta_{a,k} = \begin{cases} 1, \text{ if link } a \text{ is on path } k; \\ 0, \text{ otherwise;} \end{cases}$

$f$ is a vector of all path loads, that is $f = \{f_k, \forall k \in K\}$;

$x$ is a vector of all link loads, that is $x = \{x_a, \forall a \in A\}$;

$D$ is a vector of the demand, that is $D = \{d_{rs}, \forall rs \in OD\}$,

we can rewrite the Beckmann transformation like this, under the assumption that $\nabla z(x) = c(x)$,

$$\min \quad z(x) = \sum_a \int_0^{x_a} c_a(\omega)d\omega, \tag{2.13}$$

$$\text{subject to} \quad \Lambda f = D, \tag{2.14}$$

$$\Delta f = x, \tag{2.15}$$

$$-f \leq 0. \tag{2.16}$$

The Lagrangian of this optimization problem is

$$L(x, f, u^1, u^2, u^3) = z(x) + u^1(D - \Lambda f) + u^2(\Delta f - x) - u^3 f, \tag{2.17}$$

$$f \geq 0, \qquad u^3 \geq 0, \tag{2.18}$$

where $u^1$, $u^2$, $u^3$ are Lagrange multipliers (also called dual variables), and have dimensions $u^1 \in \mathbb{R}^{|OD|}$, $u^2 \in \mathbb{R}^{|A|}$ and $u^3 \in \mathbb{R}^{|K|}$.

The optimizer of the optimization problem is denoted by $\bar{x}$, and the corresponding dual parameters are denoted by $\bar{u}$, where $u = \{u^1, u^2, u^3\}$. As a property of the Lagrangian, the optimum is a saddle point of $L$, that is, it is the minimum with respect to $x$ and maximum with respect to $u$, so

$$L(\bar{x}, u) \leq L(\bar{x}, \bar{u}) \leq L(x, \bar{u}), \qquad \forall x, u \in \text{feasible region}. \tag{2.19}$$

***THEOREM 1***

a) *If $x$ is the optimal solution of the Beckmann transformation then it satisfies the KKT conditions.*
b) *If $z$ is convex and $x$ satisfies the KKT conditions then $x$ is the optimal solution of the Beckmann transformation.*

***PROOF OF 1 A***

A well known result from Karush, Kuhn and Tucker is that, under the slater condition, which are satisfied because the constraints are linear, at a minimum necessarily the first order conditions, called the Karush, Kuhn and Tucker (KKT) conditions, must hold:

$$\frac{\partial L}{\partial x} = 0, \tag{2.20}$$

$$\frac{\partial L}{\partial f} = 0, \tag{2.21}$$

$$\frac{\partial L}{\partial u^j} = 0, \qquad j = 1,2, \tag{2.22}$$

$$u^3 \frac{\partial L}{\partial u^3} = 0, \qquad \frac{\partial L}{\partial u^3} \leq 0, \tag{2.23}$$

$$u^3 \geq 0. \tag{2.24}$$

For the Beckmann transformation, these conditions are

$$\frac{\partial L}{\partial x} = \nabla z(x) - u^2 = c(x) - u^2 = 0, \tag{2.25}$$

$$\frac{\partial L}{\partial f} = -\Lambda^T u^1 + \Delta^T u^2 - u^3 = 0, \tag{2.26}$$

$$u^3 f = 0, \qquad f \geq 0, \tag{2.27}$$

$$u^3 \geq 0. \tag{2.28}$$

***PROOF OF 1 B***

A well known result from convex programming is: if the first order conditions hold, and z is convex, then we have obtained the optimum.

■

***THEOREM 2***

*A feasible $x$ is a UE flow pattern if and only if $x$ satisfies the KKT conditions.*

***PROOF OF 2***

We will show that we can rewrite the KKT conditions to UE.

We can rewrite equation (2.26) in the following way.

$$-\Lambda^T u^1 + \Delta^T u^2 - u^3 = 0, \tag{2.29}$$

$$\Lambda^T u^1 = \Delta^T u^2 - u^3, \tag{2.30}$$

Using condition (2.25), we can rewrite this as

$$\Lambda^T u^1 = \Delta^T c(x) - u^3. \tag{2.31}$$

Because of condition (2.27) and (2.28)

$$u^3 = \begin{cases} = 0, \text{ if } f > 0; \\ \geq 0, \text{ if } f = 0. \end{cases} \tag{2.32}$$

Also, the link costs are transformed into path costs using

$$\Delta^T c(x) = c(f). \tag{2.33}$$

Using (2.32) and (2.33) in equation (2.31), we obtain

$$\Lambda^T u^1 = \Delta^T c(x) - u^3 = c(f) - u^3 \tag{2.34}$$

Recall an entry in $\Lambda$ is 1 if path $k$ is connecting $OD$-pair $rs$. Because a path can only connect one OD-pair, but an OD-pair can be connected by more paths, $\Lambda^T \in \mathbb{R}^{|k|\times|OD|}$ has the form:

$$\Lambda^T = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & & \\ 0 & 1 & 0 & & \vdots \\ 0 & 1 & 0 & & 0 \\ & & & \ddots & 1 \\ & & & 0 & 1 \end{bmatrix}$$

For a specific $rs$ and $k \in K_{rs}$, equation (2.34) is

$$u_{rs}^1 = [\Delta^T c(x)]_k - u_k^3 = c_k(f) - u_k^3 = \begin{cases} = c_k(f), \text{ if } f_k^{rs} > 0; \\ \leq c_k(f), \text{ if } f_k^{rs} = 0. \end{cases} \tag{2.35}$$

This is exactly UE, since the costs of a path are equal for all used paths, and an unused path has equal or higher costs.

■

***THEOREM 3***

*The feasible $\bar{x}$ is a solution of the Variational Inequality formulation of the TAP if and only if the feasible $\bar{x}$ is a User Equilibrium.*

***PROOF OF 3***

First we will proof that: $\bar{x}$ is UE → $\bar{x}$ is solution of VI.

If $\bar{\boldsymbol{x}}$ is UE, with corresponding $\bar{f}$ the equilibrium conditions hold

$$\bar{f}_k^{rs}\left(c_k^{rs}(\bar{f}) - \pi^{rs}\right) = 0, \quad \forall k \in K_{rs}, \quad \forall rs; \tag{2.36}$$

$$c_k^{rs}(\bar{f}\,) \geq \pi^{rs}\,, \qquad \forall k \in K_{rs}, \quad \forall rs; \tag{2.37}$$

$$\sum_k \bar{f}_k^{rs} = d_{rs}\,, \qquad \forall k \in K_{rs}, \quad \forall rs; \tag{2.38}$$

$$\bar{f}_k^{rs} \geq 0, \;\; c_k^{rs} \geq 0. \tag{2.39}$$

Also, the constraints hold

$$\sum_k \bar{f}_k^{rs} = d_{rs}\,, \qquad \forall k \in K_{rs}, \quad \forall rs; \tag{2.40}$$

$$\bar{f}_k^{rs} \geq 0\,, \qquad \forall k \in K_{rs}, \quad \forall rs; \tag{2.41}$$

$$\bar{x}_a = \sum_{r,s,k} \delta_{a,k}^{rs} \bar{f}_k^{rs}\,, \quad \forall a. \tag{2.42}$$

From equations (2.36) and (2.37) and (2.39), we know

$$f_k^{rs}\left(c_k^{rs}(\bar{f}\,) - \pi^{rs}\right) \geq 0. \tag{2.43}$$

Subtracting (2.36) from (2.43) we obtain:

$$\left(f_k^{rs} - \bar{f}_k^{rs}\right)\left(c_k^{rs}(\bar{f}\,) - \pi^{rs}\right) \geq 0, \tag{2.44}$$

$$c_k^{rs}\left(\bar{f}_k^{rs}\right)\left(f_k^{rs} - \bar{f}_k^{rs}\right) - \pi^{rs}\left(f_k^{rs} - \bar{f}_k^{rs}\right) \geq 0. \tag{2.45}$$

Summing over paths $k$ yields

$$\sum_{rs,k} c_k^{rs}(\bar{f}\,)\left(f_k^{rs} - \bar{f}_k^{rs}\right) - \sum_{rs,k} \pi^{rs}\left(f_k^{rs} - \bar{f}_k^{rs}\right) \geq 0, \tag{2.46}$$

$$\sum_{rs,k} c_k^{rs}(\bar{f}\,)\left(f_k^{rs} - \bar{f}_k^{rs}\right) - \sum_{rs} \pi^{rs} \sum_k \left(f_k^{rs} - \bar{f}_k^{rs}\right) \geq 0. \tag{2.47}$$

From flow conservation constraint (2.38) the latter term vanishes, therefore

$$\sum_{rs,k} c_k^{rs}(\bar{f}\,)\left(f_k^{rs} - \bar{f}_k^{rs}\right) \geq 0. \tag{2.48}$$

This is the Variational Inequality formulation, with respect to $f$, namely

$$c(\bar{f})^T (f - \bar{f}) \geq 0, \quad \forall \text{ feasible } f. \tag{2.49}$$

For the proof that: $\bar{x}$ is solution of VI → $\bar{x}$ is solution of Beckmann transformation, we refer to Florian and Hearn (1995). From Theorem 1a and Theorem 2 we know that if $\bar{x}$ is solution of Beckmann transformation, then $\bar{x}$ is UE. This completes the proof.

■

## 2.3 Existence and Uniqueness

In this section, the TAP is considered with respect to the existence and uniqueness of solutions.

In the case of TAP with separable costs, Beckmann et al. (1956) showed that if the cost function $c_a$ is monotonically increasing function of $x_a$, the optimal solution is unique, and is obtained by solving an optimization problem as in equations (2.7) – (2.10).

When adding junction delays to the TAP, the cost functions become non-separable. Considering the Jacobian of the cost function,

$$J = \left[\frac{\partial c_i}{\partial x_j}\right], \tag{2.50}$$

non-separable costs yields non-zero off-diagonal entries. The Jacobian has several properties, $J$ can be symmetrical, that is

$$\frac{\partial c_i}{\partial x_j} = \frac{\partial c_j}{\partial x_i}, \qquad \forall\, i, j \tag{2.51}$$

and $J$ can be diagonally dominant, that is

$$\left|\frac{\partial c_i}{\partial x_i}\right| \geq \sum_{j \neq i} \left|\frac{\partial c_i}{\partial x_j}\right|, \quad \forall i \quad \text{(row dominance)}, \tag{2.52}$$

$$\text{and} \quad \left|\frac{\partial c_i}{\partial x_i}\right| \geq \sum_{j \neq i} \left|\frac{\partial c_j}{\partial x_i}\right|, \quad \forall i \quad \text{(column dominance)}. \tag{2.53}$$

Note that a diagonal dominant Jacobian implies a positive semi-definite Jacobian. A matrix $A$ is positive semi-definite if

$$x^T A x \geq 0, \qquad \forall\, x \neq 0. \tag{2.54}$$

A positive semi-definite Jacobian is a sufficient condition for convexity of the problem. Dafermos (1971) showed that if the Jacobian is symmetric and positive definite, the TAP has a unique solution which is obtained by a minimization problem. Later Dafermos (1980) showed that the TAP can also be expressed as a VIP, and she showed that also in the case of an asymmetric Jacobian the solution of the VIP is globally unique, under the assumption of a globally positive definite Jacobian. Diagonally dominance is a stronger criterion for the existence of a unique solution.

Bestaat er eigenlijk wel altijd een oplossing voor de toedeling, en is die oplossing uniek of kunnen er meerdere verschillende oplossingen bestaan? Als de kostenfuncties seperabel zijn, dus als de reistijd van een weg alleen afhangt van het verkeer op de weg zelf, dan bestaat er een oplossing, en die is uniek. Dat heeft Beckmann gelijk bewezen na het beschrijven van het optimalisatieprobleem. Als de kostenfuncties niet-seperabel zijn, dus als de reistijd ook afhangt van andere (bijvoorbeeld kruisende) wegen, dan ligt dit anders. Er zijn twee mogelijkheden. Als de kostenfunctie 'diagonaal dominant' is, dat betekent dat de reistijd voorál afhangt van het verkeer op de eigen weg, dan gaat alles goed: er bestaat een unieke oplossing. Maar als de kostenfunctie niet-diagonaal dominant is, dan is een unieke oplossing niet meer gegarandeerd.

In a realistic transportation network, taken junctions into account, in general the cost functions are non-separable, the Jacobian is asymmetric and the Jacobian may be non-diagonally dominant. Consider for example a priority junction. The influence of a major road (priority) on a minor road (no priority) is not equal to the influence of a minor road on a major road, and therefore the cost functions are asymmetric. Further, the cost function on a turn from a minor road crossing a major road can be dominated by the load on the major road instead of the load on the minor road itself. In this situation the cost function is non-diagonally dominant.

Therefore, when junction delays are modelled realistic, non-diagonally dominant cost functions exist, and a unique solution is not guaranteed. This means that, in reality, multiple equilibrium solutions may exist, which corresponds to different traffic flow patterns.

Als we kijken naar de reistijd (of beter gezegd: vertraging) die opgelopen wordt op kruispunten in de realiteit, dan wordt deze enerzijds beïnvloed door de hoeveelheid verkeer op de weg waar je zelf op rijdt. Hoe drukker het is in 'jouw' verkeersstroom, hoe langer je bij een kruistpunt staat te wachten. Anderzijds wordt de vertraging ook bepaald door de hoeveelheid verkeer op andere wegen. Hoe drukker het is op de kruisende verkeersstromen, hoe langer je staat te wachten. Sterker nog, er zijn voldoende situaties denkbaar, bijvoorbeeld op voorrangskruispunten, waarbij de kruisende verkeersstroom een grotere invloed heeft op de vertraging dan je eigen verkeersstroom. Dit betekent in wiskundige termen dat de kostenfunctie niet-diagonaal dominant is.
Dit betekent dat er in de realiteit geen unieke oplossing is gegarandeerd, en kunnen meerdere gebruikersevenwichten bestaan, verschillende verkeerspatronen waarin toch niemand zijn reistijd kan verkleinen.

One can question the goal of modelling traffic. Should the model approach reality as close as possible, even if it adopts the existence of several equilibria? That would imply that the model could result in different flow patterns, depending on for example its initialization or the solving method. In that case the given solution is not necessarily the same as the real situation, since there are more solutions. Also when comparing different scenarios, a fair comparison could be problematic. For these reasons, one can state it is better for the model to always converge to the same unique solution. But on the other hand, when a unique solution is required, convexity of the problem is needed. That implies no non-diagonal dominant turn costs are accepted, and that is not always realistic. Concluding, if one states the turn costs must be as accurate as possible, one has to take the existence of several local minima for granted.

Als men besluit een verkeersmodel zoveel mogelijk waarheidsgetrouw te houden, en dus ook de kruispuntvertragingen realistisch in het model op te nemen, dan is de kostenfunctie die de kruispuntvertraging berekent dus niet (altijd) diagonaal dominant. Dit leidt automatisch tot een situatie waarin er meerdere oplossingen kunnen bestaan. Dit kan een groot probleem zijn als het verkeersmodel gebruikt wordt om scenarios te vergelijken. Een gemeente kan bijvoorbeeld twee modellen maken, één zonder brug en één met brug. De oplossingen (verkeerspatronen) die in beide situaties berekent worden, worden dan met elkaar vergeleken om te kijken wat de invloed is van de brug. De gemeente wil er natuurlijk zeker van zijn dat de gevonden verschillen in de verkeerstromen veroorzaakt worden door de extra brug, en niet doordat er 'toevallig' twee oplossing zijn gevonden die ver uit elkaar liggen, terwijl er misschien ook twee mogelijke oplossingen zijn die meer op elkaar lijken. Om de verschillen toe te kunnen schrijven aan de interventie (in dit geval de brug) en om een eerlijk vergelijk te kunnen maken, is een unieke oplossing vereist.

## 2.4 OMNITRANS

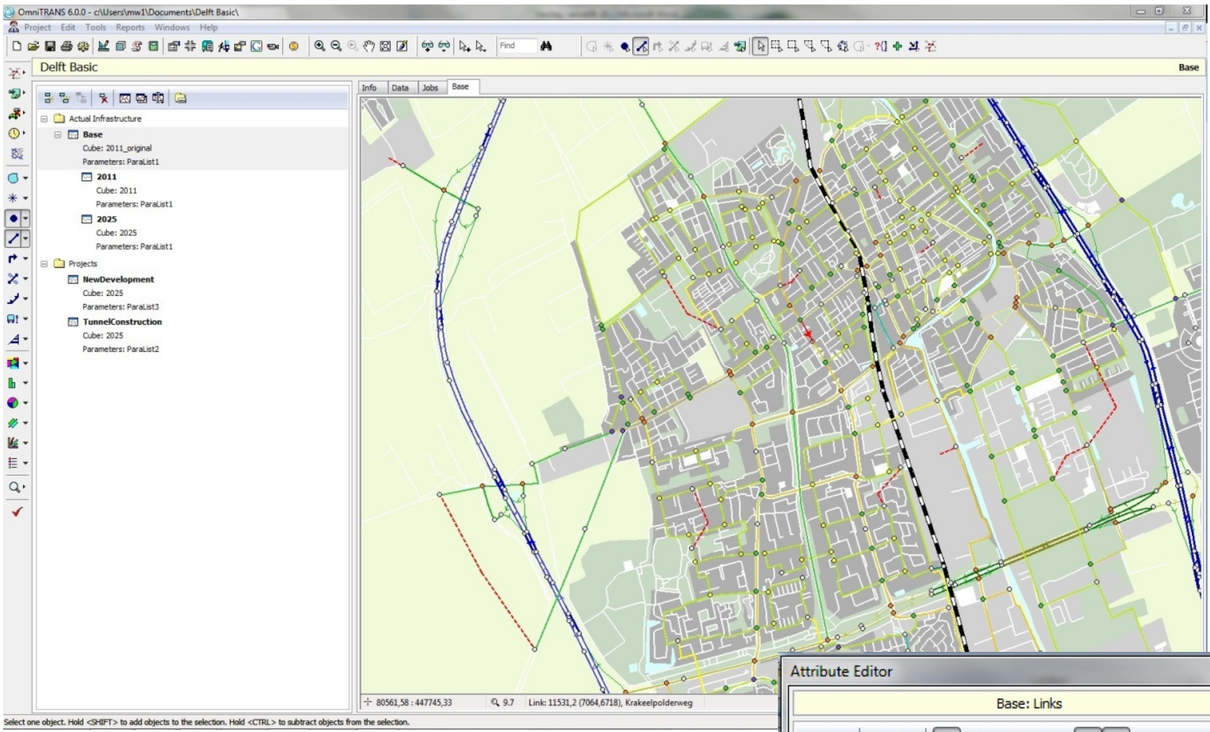



**FIGURE 6: OMNITRANS SCREENSHOT**

Before zooming in on the implementation of the junction modelling in OmniTRANS, we will first discuss OmniTRANS in general.

Omnitrans International is a software company specialised in traffic and transport. Besides consultancy projects, Omnitrans International develops and maintains their main product OmniTRANS. OmniTRANS is a transport planning application, see Figure 6 for a screenshot. Transportation networks can be implemented, and properties of nodes and links (representing junctions and roads) can be set. In Figure 7 the attribute editor of a link is shown, where the speed, capacity and other properties can be set. Also junctions can be defined, for example the setting of lanes, turn possibilities and the junction type can be specified, see Figure 8.

In OmniTRANS an assignment can be executed. The solution can be visualized, by 'plotting' the load on the network. An example is shown in Figure 9. The colour and the thickness of the links represent selected information, for example the load, the costs or the load / capacity ratio. Using this visualization, congestion and bottlenecks can easily be examined.

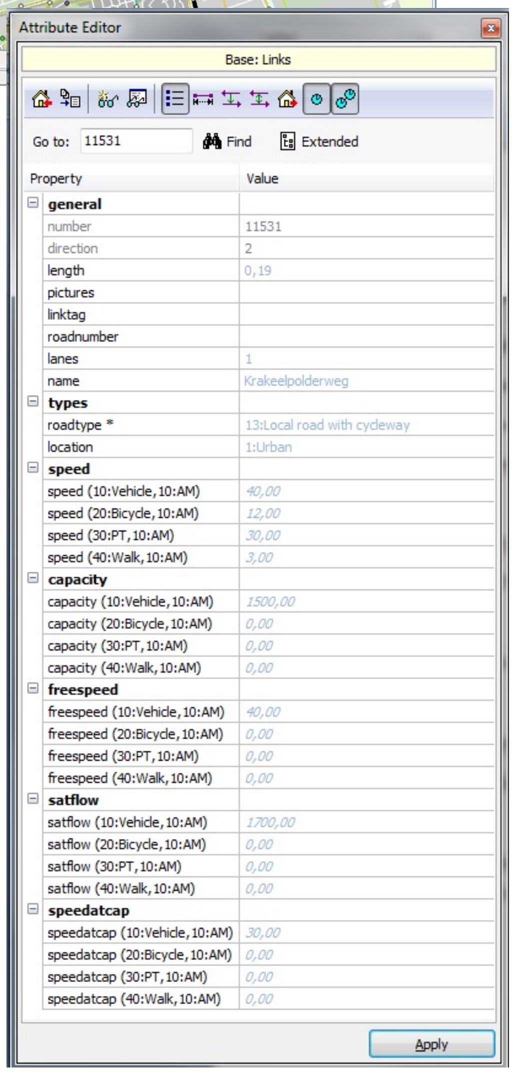



**FIGURE 7: ATTRIBUTE EDITOR**

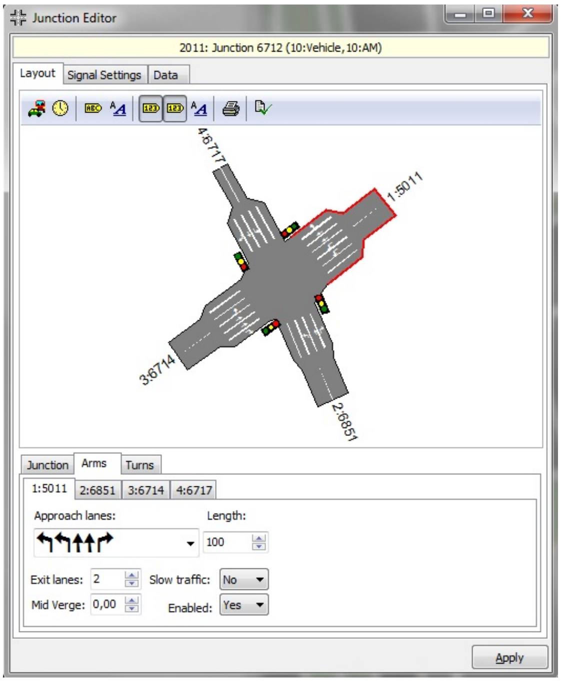



FIGURE 8: JUNCTION EDITOR

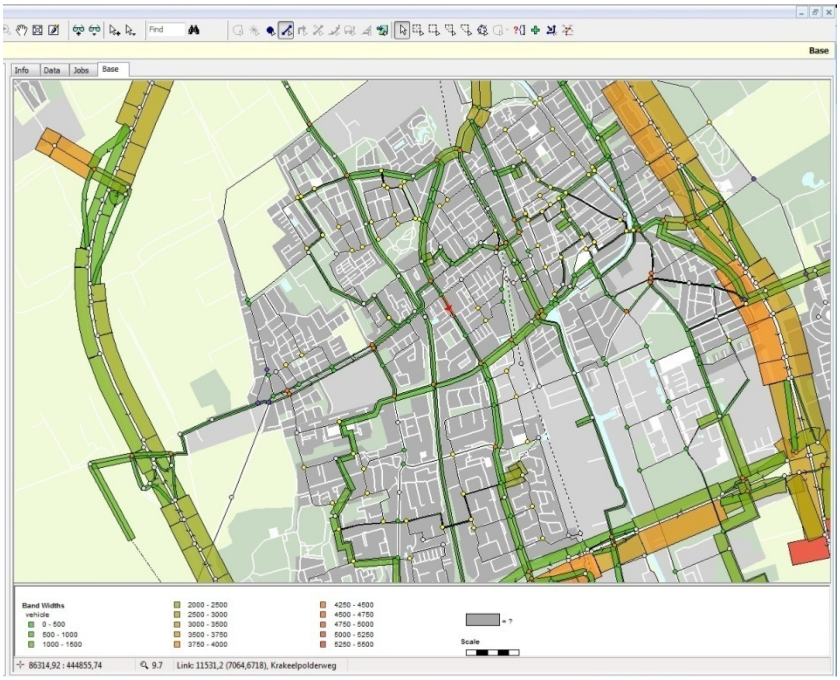



FIGURE 9: VISUALIZATION OF CONGESTION

## 2.5 Junction Modelling

### 2.5.1 Turns in the Network

In congested urban networks, relatively much of the time spent on a journey is incurred by queuing and turning at junctions. In most traffic models junction delays are ignored, or average delays are used. For highway modelling, still accurate solutions are obtained, but in urban networks the lack of accurate junction delays in the model can lead to significant errors. To illustrate the importance of the contribution of junction delays to the total travel time, see Figure 10. No one would easily ignore the delay at this junction when calculating the travel time of a route passing this junction.

**FIGURE 10: CONGESTED JUNCTION**

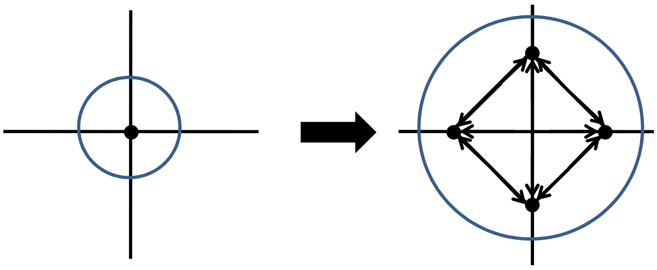


**FIGURE 11: EXPANDED JUNCTION**

A common way to model junctions in a network, is to expand the junction nodes. All possible turns become extra links and all branches of the junction get a node. An example of an expanded junction with four branches is shown in Figure 11. On each turn a cost function is defined, which is a function of the load on the turn itself and the load on conflicting turns. In the next section an explanation is given of the specific junction modelling in OmniTRANS.

Bij de toedeling spelen reistijden een belangrijke rol. We 'zoeken' immers naar een situatie waarin alle reizigers een minimale reistijd ervaren van hun herkomst naar hun bestemming. Een goede en waarheidsgetrouwe berekening van de reistijden is dus van groot belang. In stedelijke verkeersnetwerken wordt een groot deel van de reistijd over de hele route opgelopen bij kruispunten. Het is daarom belangrijk dat de kruispuntvertragingen accuraat meegenomen worden in de berekening van de reistijden.
In de meeste verkeersmodellen wordt de kruispuntvertraging genegeerd, of wordt er gewerkt met vaste waardes voor kruispuntvertragingen, ongeacht de drukte op een kruispunt. Echter, OmniTRANS gebruikt een hele uitgebreide module voor de kruispuntmodellering, die gegeven de hoeveelheid verkeer op de eigen en kruisende verkeersstromen een vertraging berekent in seconden. De berekeningen die hiervoor gemaakt worden staan beschreven in paragraaf 2.5.2.
Hoe 'zitten' deze kruispuntvertragingen eigenlijk verwerkt in het verkeersmodel? We hebben eerder gezien dat er aan wegen een kostenfunctie werd 'gehangen', die gegeven de hoeveelheid verkeer de reistijd berekende. Hetzelfde doen we voor 'turns', dat zijn afslagbewegingen op een kruispunt. Turns worden gerepresenteerd als een lijn in het netwerk, zie Figuur 11. Aan elke turn wordt ook een kostenfunctie 'gehangen', die de vertraging voor die turn geeft.

### 2.5.2 JUNCTION MODELLING IN OMNITRANS

OmniTRANS contains an extensive junction modelling module, where all types of junctions can be defined, namely equal junctions, priority junctions, signalized junctions and roundabouts. Also, a number of lanes on every branch of the junction and possible turning movements on those lanes can be defined, see Figure 12 for some examples. Calculations are made per lane, per turn, or per lane group. A lane is a road section used for a turn or a combination of turns. Turns are movements of one branch to another, possible turns are left, through and right. Lane groups are circled in Figure 12. Note that if more turning movements are possible from one lane, the turning movements are in the same lane group. Also, if the same turning movement can be made from different lanes, those lanes are in the same lane group.

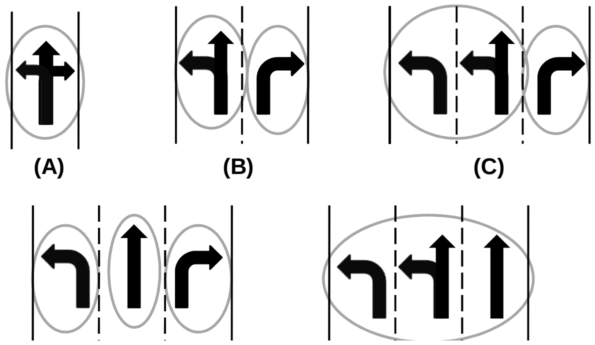



**FIGURE 12: ARRANGEMENTS OF TURNINGMOVEMENTS ON LANES, LANEGROUPS ARE CIRCLED**

Generally, the capacity is calculated per lane group and the delay is calculated per lane. In the formulas the lanes are denoted by $l$, the lane groups are denoted by $g$ and the turns are denoted by $t$. The calculation of the capacity and the delay differs per junction type, the main differences are between signalized and unsignalized junctions. Besides capacity and delay, also the setting of traffic lights are (optionally) calculated in OmniTRANS. In the next sections, the main calculations are given.

For clearness, these calculations are simplified in a sense that all the parameters are omitted. We maintain the ´structure´ of the formulas, in a way that it is workable and relevant for our purposes. For a complete overview of the calculations in junction modelling in OmniTRANS, see the documentation of OmniTRANS 'Explanation of Junction Modelling' (Brandt & Schilpzand, 2007). These calculations are partially based on the formulas for capacity and delay in the Highway Capacity Manual (Transportation Research Board, 2000).

Next to omitting parameters, also a variable is omitted, namely the 'apparent conflict' variable. Apparent conflicts are situations where a driver unnecessary waits for another driver. This could happen for example when driver A wants to enter a roundabout, but is waiting for driver B, who is going to exit the roundabout. Driver B forgot his turning signal, so actually there is no conflict, but still driver A experiences a conflict situation and is waiting unnecessarily for driver B. In this study no apparent conflicts, but only real conflicts are taken into account.

First the calculations of capacity and delay of unsignalized junctions are discussed. Thereafter, the calculation of a signalized junction, including the setting of traffic lights, is discussed. Finally, a general cost function of a turn is given.

#### 2.5.2.1 UNSIGNALIZED JUNCTIONS

***CAPACITY***

The calculation of the capacity of a lane group at an unsignalized junction is as follows

$$q_g = \max\left(\sigma_g - \sum_{b \in Y_g} x_b \ , q_{\min}\right), \qquad (2.55)$$

where

$q_g$ is capacity of lane group $g$;
$\sigma_g$ is saturation flow;
$x_b$ is load on lane group $b$;
$Y_g$ is set of lane groups conflicting with lane group $g$ and
$q_{\min}$ is minimal capacity of lane group $g$.

The saturation flow is the capacity when there are no conflicting movements. The minimal capacity is used to avoid total congestion.

The calculation of the capacity at a priority junction is extended with some extra terms. Those terms provide the decrease of the capacity as a result of the difficulty of crossing a priority junction for a minor road. Those terms are fixed values for every junction, and therefore omitted.

The delay depends on the capacity. The delay is calculated per lane, whereas the capacity is calculated per lane group. We can obtain the capacity per lane by dividing the capacity of the lane group proportionally over the lanes. For the complete calculation, see Brandt and Schilpzand (2007).

***DELAY***

The general formula for the delay (cost function) at an unsignalized junction is as follows

$$c_l = \min\left(c_{1,l} + c_{2,l} + c_{3,l}, c_{max,l}\right), \qquad (2.56)$$

where

$c_l$ is average delay on lane $l$;
$c_{1,l}$ is uniform delay on lane $l$;
$c_{2,l}$ is incremental delay on lane $l$;
$c_{3,l}$ is geometric delay on lane $l$ and
$c_{max,l}$ is maximal delay on lane $l$.

$c_{max,l}$ is used to avoid total congestion.

The uniform delay $c_{1,l}$ and the incremental delay $c_{2,l}$ are calculated as follows

$$c_{1,l} = \frac{1}{q_l}, \qquad (2.57)$$

$$c_{2,l} = \begin{cases} \left(\frac{x_l}{q_l} - 1\right) + \sqrt{\left(\frac{x_l}{q_l} - 1\right)^2 + \frac{x_l}{(q_l)^2}}, & \text{if } \frac{x_l}{q_l} > \alpha; \\ 0, & \text{if } \frac{x_l}{q_l} \le \alpha, \end{cases} \qquad (2.58)$$

where

$x_l$ is load on lane $l$;

$q_l$ is capacity of lane $l$ and

$\alpha$ is a parameter usually set on 0.5.

Both uniform delay and incremental delay are proportional with the load on the lane $x_l$, and inversely proportional with the capacity $q_l$. Note that the incremental delay is only taken into account when the ratio $x_l/q_l$ is greater than $\alpha$, which is usually 0,5.

The geometric delay $c_{3,l}$ is calculated per junction type as follows:

equal junction:
$$c_{3,l} = \begin{cases} 1, & \text{if } x_l > 0; \\ 0, & \text{if } x_l = 0, \end{cases} \tag{2.59}$$

priority junction:
$$c_{3,l} = \begin{cases} 1, & \text{if } x_l > 0 \text{ and branch is major road;} \\ \frac{x_{\text{adjusted}}}{x_l}, & \text{if } x_l > 0 \text{ and branch is minor road;} \\ \beta, & \text{if } x_l = 0, \end{cases} \tag{2.60}$$

unsignalized roundabout:
$$c_{3,l} = \beta, \tag{2.61}$$

where

$x_{\text{adjusted}} = \beta x_{\text{left}} + x_{\text{through}} + \beta x_{\text{right}}$, with

$x_{\text{left}}$ is load on left turn movements,

$x_{\text{through}}$ is load on through movements,

$x_{\text{right}}$ is load on right turn movements;

$x_l$ is load on lane $l$;

$\beta$ is usually set on 7.

Note that in the adjusted load $x_{\text{adjusted}}$ the load on left and right movements have a greater weight than the trough movement.

Considering the Traffic Assignment Problem, the cost function of a turning movement at a junction is relevant. Generally, the cost function is a monotonically increasing function of the load on its own turning movement and the load on conflicting movements. The influence of the load on the conflicting movements is via the capacity: when load on conflicting movements increases, the capacity decreases, and therefore the cost function increases.

#### 2.5.2.2 Signalized Junction

In OmniTRANS there are three possibilities to specify a signalized junction. One option is ‘manual’, in which the signal cycle time and the green times can be set by hand in the junction editor. Another option is ‘automated’. In that case the Junction Modelling Module in OmniTRANS will optimize the setting of the traffic lights during the assignment, such that given the demand at a junction the total travel time is minimized. This roughly corresponds to the real situation where the traffic lights interact with the demand, using induction loops in the roads. The last option in ‘actuated’, which is a combination of manual and automated. Some boundaries are set by hand, for example minimum and maximum green times per lane group, and the actual setting is done automated by the Junction Modelling Module.

***Capacity and Settings of Traffic Lights***

The calculation of the capacity at a signalized junction is done as follows. First, the capacity of a turn is set on the saturation flow,

$$q'_t = \sigma_t \, , \tag{2.62}$$

where

$q'_t$ is base capacity of turn $t$ and
$\sigma_t$ is saturation flow of turn $t$.

Then the capacity per lane is calculated from the capacity per turn as follows,

$$q'_l = \frac{\sum_{t \text{ on lane } l} q'_t}{h_l} \, , \tag{2.63}$$

where

$q'_l$ is base capacity of lane $l$ and
$h_l$ is number of turns on lane $l$.

Naturally the final capacity of a lane depends on the green time, which is the period the traffic light is green for that lane, so that the travellers can pass the junction. The green time is the same for all lanes in one lane group. For calculating the green time, information is needed about the conflicting lane groups. A conflict matrix is used, where conflicts between all pairs of lane groups are given. Then 'maximum conflict groups' are obtained, these are maximum groups consisting of lane groups which are in conflict with all the other lane groups in the group. For every conflict group, the signal cycle time is calculated, it is set on the minimum time period such that the junction can 'digest' all the traffic in the conflict group. The normative conflict group is the conflict group with the highest signal cycle time. This cycle time is used for the junction. This is limited by a maximum cycle time, to avoid a very high value. Then the green times of the lane groups are calculated. First the green times of the normative conflict group are calculated, thereafter the green times of the other conflict groups. For a more specific explanation of the calculation of conflict groups, cycle times and green times, see Brandt and Schilpzand (2007).

The final capacity of a lane is the base capacity times the fraction of green time the lane gets, that is

$$q_l = \theta_l q'_l \, , \tag{2.64}$$

where

$q_l$ is capacity of lane $l$;
$\theta_l$ is fraction green time of lane $l$ of total cycle time and
$q'_l$ is base capacity of lane $l$.

***Delay***

The delay at a signalized junction is calculated with the general formula for the delay, where only the uniform delay $c_{1,l}$ is differs significant from the delay at an unsignalized junction. The general formula for the delay is

$$c_l = \min\left(c_{1,l} + c_{2,l} + c_{3,l}, c_{max,l}\right), \quad (2.65)$$

where

$c_l$ is average delay on lane $l$;
$c_{1,l}$ is uniform delay on lane $l$;
$c_{2,l}$ is incremental delay on lane $l$;
$c_{3,l}$ is geometric delay on lane $l$ and
$c_{max,l}$ is maximal delay on lane $l$.

At a signalized junction uniform delay $c_{1,l}$ is calculated as follows

$$c_{1,l} = \tau \cdot \frac{(1-\theta_l)^2}{1-\min\left(1, \frac{x_l}{q_l}\right) \cdot \theta_l}, \quad (2.66)$$

where

$\theta_l$ is fraction green time of lane $l$ of total cycle time;
$\tau$ is cycle time;
$x_l$ is load on lane $l$ and
$q_l$ is capacity of lane $l$.

For explaining how $\theta_l$ is calculated, we need to introduce phases, which we denote by $p$. The cycle time consists of phases, and a phase is a time period where a set of lanes gets green. $G_p$ is the set of lanes that get green in phase $p$. The fraction of green time of lane $l$ is calculated as follows

$$\theta_l = \frac{\max\limits_{l \in G_p} \frac{x_l}{q'_l}}{\sum_p \left(\max\limits_{l \in G_p} \frac{x_l}{q'_l}\right)}, \quad (2.67)$$

where

$x_l$ is load on lane $l$;
$q'_l$ is base capacity of lane $l$ and
$G_p$ is the set of lanes that get green in phase $p$.

This can be interpreted as follows. In a certain phase the lane with the highest load / capacity ratio is obtained. The proportion of this load / capacity ratio of the sum of all maximum load / capacity ratios of all phases, is the proportion of green time it gets. So green times are proportionally divided based on the maximum load / capacity ratio of all the phases.

The incremental delay $c_{2,l}$ is calculated in the same manner as at an unsignalized junction, so

$$c_{2,l} = \begin{cases} \left(\frac{x_l}{q_l} - 1\right) + \sqrt{\left(\frac{x_l}{q_l} - 1\right)^2 + \frac{x_l}{(q_l)^2}}, & \text{if } \frac{x_l}{q_l} > \alpha; \\ 0, & \text{if } \frac{x_l}{q_l} \leq \alpha, \end{cases} \quad (2.68)$$

where

$x_l$ is load on lane $l$;
$q_l$ is capacity of lane $l$ and
$\alpha$ is usually set on 0.5.

The geometrical delay $c_{3,l}$ is calculated roughly the same way as at an unsignalized junction,

$$c_{3,l} = \begin{cases} \frac{x_{\text{adjusted}}}{x_l}, & \text{if } x_l > 0; \\ \beta, & \text{if } x_l = 0, \end{cases} \tag{2.69}$$

where

$x_{\text{adjusted}} = \beta x_{\text{left}} + x_{\text{through}} + \beta x_{\text{right}}$, with
$x_{\text{left}}$ is load on left turn movements,
$x_{\text{through}}$ is load on through movements,
$x_{\text{right}}$ is load on right turn movements;
$x_l$ is load on lane $l$ and
$\beta$ is usually set on 7.

Concluding, as at an unsignalized junction, also at a signalized junction the cost function is a monotonically increasing function of the load on its own turning movement and the load on conflicting movements, although the cost function is slightly different.

#### 2.5.2.3 Cost Functions for a Turn at a Junction

For simplicity, we assume that every lane corresponds to one turn, see Figure 12(D), we omit the minimal capacity, the maximal delay and the geometric delay ($c_{3,l}$).

Wat we vooral van deze berekeningen kunnen leren is dat de kruispuntvertragingen heel precies worden berekend. En hieronder staat dan dé kostenfunctie voor een ongeregeld kruispunt (vergelijking (2.70)) en voor een geregeld kruispunt (vergelijking (2.71)). Hoewel deze sterk vereenvoudigd zijn, is de ‘structuur’ van de functies behouden, zodat we deze exemplaren kunnen gebruiken om te kijken of ze voldoen aan de eisen voor een goede toedeling.

In total, the simplified cost function for a turn on a unsignalized junction is as follows.

$$\begin{aligned} c_t &= c_{1,t} + c_{2,t} \\ &= \frac{1}{q_t} + \left(\frac{x_t}{q_t} - 1\right) + \sqrt{\left(\frac{x_t}{q_t} - 1\right)^2 + \frac{x_t}{(q_t)^2}} \\ &= \frac{1}{\sigma_t - \sum_{b \in Y_g} x_b} + \left(\frac{x_t}{\sigma_t - \sum_{b \in Y_g} x_b} - 1\right) + \sqrt{\left(\frac{x_t}{\sigma_t - \sum_{b \in Y_g} x_b} - 1\right)^2 + \frac{x_t}{\left(\sigma_t - \sum_{b \in Y_g} x_b\right)^2}}. \end{aligned} \tag{2.70}$$

The simplified cost function for a turn on a signalized junction is

$$\begin{aligned} c_t &= c_{1,t} + c_{2,t} + c_{3,t} \\ &= \tau \cdot \frac{(1-\theta_l)^2}{1 - \min\left(1, \frac{x_t}{q_t}\right) \cdot r_t} + \left(\frac{x_t}{q_t} - 1\right) + \sqrt{\left(\frac{x_t}{q_t} - 1\right)^2 + \frac{x_t}{(q_t)^2}} + 0 \end{aligned}$$

$$= \tau \cdot \frac{(1-\theta_l)^2}{1-\min\left(1,\frac{x_t}{\theta_l\sigma_t}\right)\cdot\theta_l} + \left(\frac{x_t}{\theta_l\sigma_t}-1\right) + \sqrt{\left(\frac{x_t}{\theta_l\sigma_t}-1\right)^2 + \frac{x_t}{(\theta_l\sigma_t)^2}}. \quad (2.71)$$

where

$c_t$ is cost function (delay) on turn $t$;
$c_{1,t}$ is uniform delay on turn $t$;
$c_{2,t}$ is incremental delay on turn $t$;
$c_{3,t}$ is geometric delay on turn $t$;
$q_t$ is capacity on turn $t$;
$\sigma_t$ is saturation flow on turn $t$;
$x_t$ is load on turn $t$;
$Y_t$ is set of turns conflicting with turn $t$;
$\theta_t$ is fraction of green time of turn $t$ of total cycle time and
$\tau$ is cycle time.

Note that when the signalized junctions are specified 'manual', $\theta_t$ and $\tau$ are fixed. When the signalized junctions are specified 'automated' or 'actuated', $\theta_t$ is a monotonically decreasing function of $y_t$, because the more load on the conflicting movements, the less percentage of green time the total turn $t$ gets:

$$\theta_t = \theta_l = \frac{\max\limits_{l\in G_p}\frac{x_l}{q'_l}}{\sum_p\left(\max\limits_{l\in G_p}\frac{x_l}{q'_l}\right)} = \frac{\max\limits_{l\in G_p}\frac{x_l}{q'_l}}{\max\limits_{l\in G_p}\frac{x_l}{q'_l} + \sum_{p\neq p \text{ where } l \text{ gets green}}\left(\max\limits_{m\in G_p}\frac{x_m}{q'_m}\right)}, \quad (2.72)$$

where

$x_l$ is load on lane $l$;
$q'_l$ is base capacity of lane $l$;
$G_p$ is the set of lanes that get green in phase $p$ and
lanes are denoted by $l$ and $m$.

Because all lanes $m$ get green in another phase as lane $l$, lane $m$ is in conflict with lane $l$ by definition. So load $x_m$, where $m \in Y_l$, is in conflict with lane $l$, where $Y_l$ is the set of conflicting loads with lane $l$.

Concluding, the cost function of a turn at a junction is, at both unsignalized and signalized junctions, a monotonically increasing function of the load on its own turning movement $x_t$ and the load on conflicting movements $x_m$, where $m \in Y_t$.

In this study, we assume that the signalized junctions are specified 'manual', so $\theta_l$ and $\tau$ are fixed. Later on, we will inspect the consequences of that assumption to the generalization of the conclusions. This assumption means that the cost function of a signalized junction is a monotonically increasing function only of the load on its own turning movement, so the cost function is separable. Only for an unsignalized junction the cost function is non-separable.

## 2.6 TURNS IN A NETWORK

There are several interpretations when adding turns, with its load and cost, to a network. As we have seen in Section 2.5 in OmniTRANS the addition of turns is implemented as an extension of the network. The junctions are expanded, all turns become extra links and every branch of a junction gets a node, see Figure 11. Implemented in this manner, the solution of the assignment directly provides turn loads and costs.

Considering the addition of turns to the network, naturally a question rises, namely can we construct the turn loads from the link loads? Although this is not relevant for OmniTRANS, it is still an interesting issue.

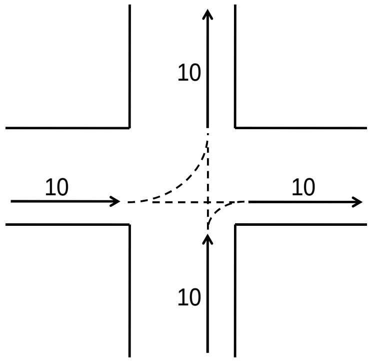



**FIGURE 13: JUNCTION WITH UNKNOWN TURN LOADS**

For example, in the junction in the Figure 13, the link loads are given. Ten travellers are approaching the junction from the West, ten from the South. Furthermore, ten travellers are leaving the junction to the North, and ten to the East. Can we determine how many travellers making specific turns, such that load fits with the link loads? There are multiple feasible solutions to this problem, two solutions of this instance are shown in a schematic representation in Figure 14.

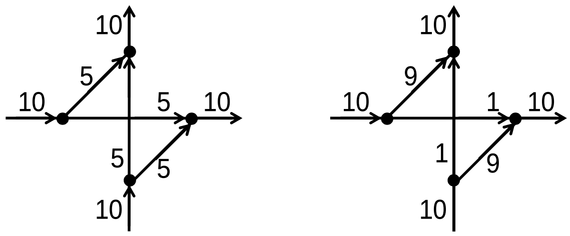



**FIGURE 14: POSSIBLE SOLUTIONS FOR TURN LOADS**

A way to construct turn loads from link loads is via the path solution. In a path solution, the variables are paths instead of links and so the loads and costs are calculated per path. Therefore, the portion of travellers from one link to another is known, and the turns loads can be extracted.

### 2.6.1 FROM A LINK SOLUTION TO A PATH SOLUTION

As discussed in Section 2.2.1, the link solution and path solution are related by the link-path incidence relationship

$$x_a = \sum_{r,s,k} \delta_{a,k}^{rs} f_k^{rs}, \qquad (2.73)$$

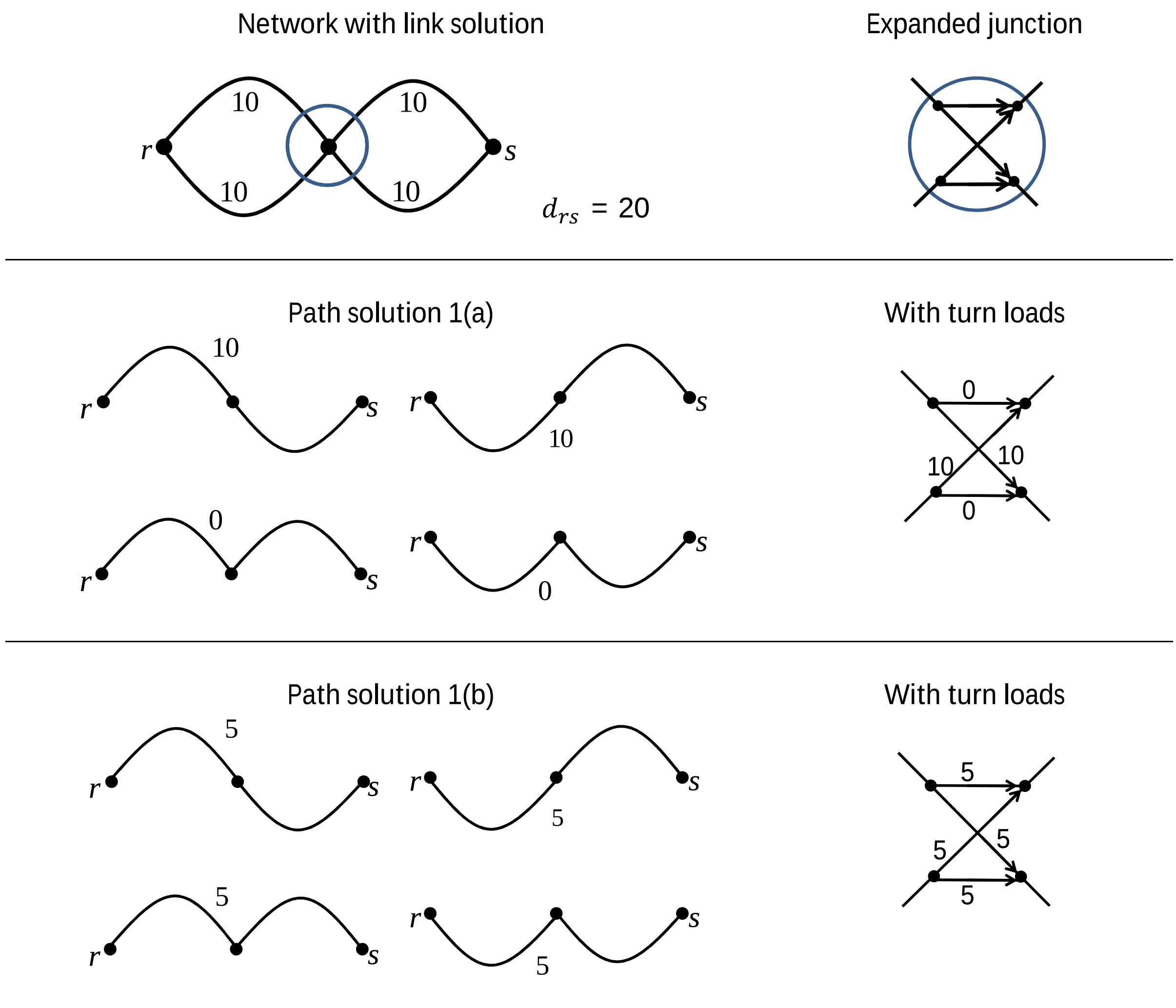
where

$x_a$ is the load on link $a$;
$f_k^{rs}$ is the load of path $k$ connecting $r$ and $s$;
$$\delta_{a,k}^{rs} = \begin{cases} 1, \text{ if link } a \text{ is on path } k \text{ connecting } r \text{ and } s \text{ and} \\ 0, \text{ otherwise.} \end{cases}$$

Although a path solution uniquely determines a link solution, this does not hold for the reverse. Given a link solution, the path solution is not unique. And also, as shown in Figure 14, given a link solution the turn loads are not uniquely defined. This becomes more clear in the following two examples.

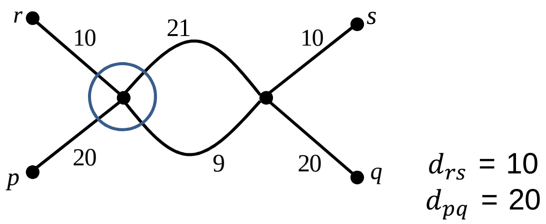
In example 1, a network is shown with one OD-pair, one junction and four possible routes. Given the link solution, two possible path solutions are given, in such a way that the link-path incidence relationship holds. Next to the path solution, the resulting turn loads on the junction are given. Example 2 contains an other network, with two OD-pairs, two junctions and two possible routes per OD-pair. Also two path solutions and the resulting turn loads on the left junction are given.

**EXAMPLE 1: MORE PATH SOLUTIONS GIVEN A LINK SOLUTION**

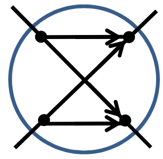


#### Example 2: More Path Solutions Given a Link Solution

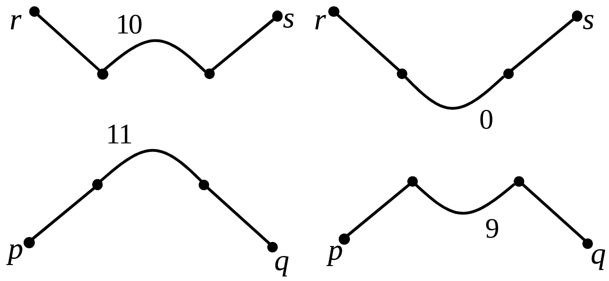
Network with link solution



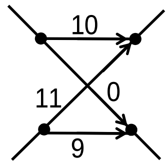

Expanded (left) junction



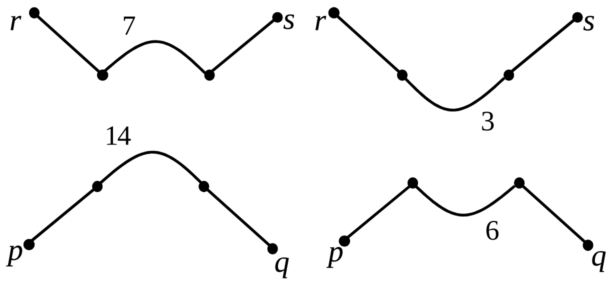
Flow solution 2(a)



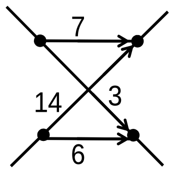

With turn loads on left junction




Flow solution 2(b)




With turn loads on left junction




Considering the fact that a link solution does not uniquely determine a path solution, and thus turn loads, we can state that, given a link solution, a path solution has to be chosen according to some strategy. Naturally, we search for the 'most likely' path solution, given the link solution.

### 2.6.2 Most Likely Path Solution

Larsson, Lundgren, Rydergren and Patriksson (2001) studied most likely path flows, given a link flow solution. They state that travellers are indifferent to which route they use, among all equal cost routes. Therefore, all route choices are equally probable. Using general principles from information theory, the most likely path solution is the one with the highest entropy.

The entropy value is, by definition,

$$-\sum_{rs}\sum_{k\in K_{rs}} f_k^{rs} \ln f_k^{rs}, \qquad (2.74)$$

where $f_k^{rs}$ is the load of path $k$ connecting $r$ and $s$.

A solution with a maximum, or 'sufficiently high', entropy value is also referred to as a path proportional solution.

To illustrate the meaning of the entropy value, the entropy value is calculated for the two examples above. In example 1, the entropy value for path solution 1(a) is

$$-\sum_{rs}\sum_{k\in K_{rs}} f_k^{rs}\ln f_k^{rs} = 2(10\cdot\ln 10) + 2(0\cdot\ln 0) \approx -46.05,$$

and the entropy value for path solution is 1(b) is

$$-\sum_{rs}\sum_{k\in K_{rs}} f_k^{rs}\ln f_k^{rs} = 4(5\cdot\ln 5) \approx -32.19.$$

In example 2, the entropy value for path solution 2(a) is

$$-\sum_{rs}\sum_{k\in K_{rs}} f_k^{rs}\ln f_k^{rs} = 10\cdot\ln 10 + 0\cdot\ln 0 + 11\cdot\ln 11 + 9\cdot\ln 9 \approx -69.18,$$

and the entropy value for path solution is 2(b) is

$$-\sum_{rs}\sum_{k\in K_{rs}} f_k^{rs}\ln f_k^{rs} = 7\cdot\ln 7 + 3\cdot\ln 3 + 14\cdot\ln 14 + 6\cdot\ln 6 \approx -64.61.$$

The path solution of 1(b) and 2(b) has a higher entropy value than respectively the path solution of 1(a) and 2(a). These path solutions are indeed more likely path solutions, because the loads are more equally distributed over all possible paths.

The problem of finding the most likely path solution given a link solution, is the maximum entropy problem, which is as follows

$$\max \quad -\sum_{rs}\sum_{k\in K_{rs}} f_k^{rs}\ln f_k^{rs}, \tag{2.75}$$

$$\text{subject to} \quad \sum_{k\in K_{rs}} f_k^{rs} = d_{rs}, \qquad \forall rs; \tag{2.76}$$

$$\sum_{r,s,k} \delta_{a,k}^{rs} f_k^{rs} = x_a, \qquad \forall a; \tag{2.77}$$

$$f_k^{rs} \geq 0, \qquad \forall rs,\ \forall k \in K_{rs}, \tag{2.78}$$

where

$x_a$ is the load on link $a$;
$f_k^{rs}$ is the load of path $k$ connecting $r$ and $s$;
$\delta_{a,k}^{rs} = \begin{cases} 1, \text{ if link } a \text{ is on path } k \text{ connecting } r \text{ and } s; \\ 0, \text{ otherwise, and} \end{cases}$
$d_{rs}$ is the demand from $r$ to $s$.

In the maximum entropy problem the entropy value is maximized under the equilibrium constraints and the link-path incidence relationship. This is a strictly convex problem. Note that $x_a$ is the given link solution, and therefore is an input parameter.

Several solution methods of the maximum entropy problem are given by Larsson et al. (2001). Also Freund and Saxena (1984) give a solution method, the complexity of this algorithm is in order $O(n)$, where $n$ is the number of paths.

Note that the maximum entropy problem is only an issue when one wants to obtain turn loads from a link solution. Considering the implementation of turns in OmniTRANS, the turn loads are automatically generated in the assignment. Furthermore, the entropy value is of interest when the output of the assignment is a path solution instead of a link solution. This is not the case in OmniTRANS. For these reasons, the entropy value of a solution is not relevant for this study.

## 2.7 FINAL PROBLEM FORMULATION

Our problem is the static user equilibrium-based Traffic Assignment Problem with deterministic route choice, which we referred to as TAP, expanded with junction delays. In this section an overview is given of all implications of the addition of junction delays, as implemented in OmniTRANS, to the TAP.

We hebben in Hoofdstuk 2 bekeken wat voor soort probleem de toedeling eigenlijk is, hoe kruispuntmodellering dat kan beïnvloeden, en hoe kruispunten gemodelleerd zijn in OmniTRANS. In deze laatste paragraaf komt dat samen en presenteren we een uiteindelijke probleem formulering.

**(NON-)SEPARABLE COST FUNCTIONS**

When adding junction delays to the TAP, the nodes are expanded, and all turns become extra links, as we have seen in Section 2.5. On all links, both 'normal' links (representing roads) and 'turn' links (representing turns) a cost function is defined. The cost function of a 'normal' link is the BPR function

$$c_a(x_a) = \frac{L_a}{v_a^{\max}}\left(1 + \alpha_a\left(\frac{x_a}{q_a}\right)^{\beta_a}\right), \tag{2.79}$$

the simplified cost function of a 'turn' link is, in the case of an unsignalized junction

$$c_a(x) = \frac{1}{\sigma_a - \sum_{b\in Y_a} x_b} + \left(\frac{x_a}{\sigma_a - \sum_{b\in Y_a} x_b} - 1\right) + \sqrt{\left(\frac{x_a}{\sigma_a - \sum_{b\in Y_a} x_b} - 1\right)^2 + \frac{x_a}{\left(\sigma_a - \sum_{b\in Y_a} x_b\right)^2}}, \tag{2.80}$$

and in the case of a signalized junction

$$c_a(x_a) = \tau \cdot \frac{(1-\theta_a)^2}{1 - \min\left(1, \frac{x_a}{\theta_a\sigma_a}\right)\cdot\theta_a} + \left(\frac{x_a}{\theta_a\sigma_a} - 1\right) + \sqrt{\left(\frac{x_a}{\theta_a\sigma_a} - 1\right)^2 + \frac{x_a}{(\theta_a\sigma_a)^2}}, \tag{2.81}$$

where

$c_a$ is the cost function (delay) on link $a$;
$L_a$ is the length of link $a$;
$v_a^{\max}$ is the maximum speed on link $a$;
$x_i$ is the load on link $i$;
$Y_a$ is the set with links conflicting with $a$;
$q_a$ is the capacity of link $a$;
$\alpha_a$ and $\beta_a$ are constants defined for every link;
$\sigma_a$ is the saturation flow;
$y_a$ is the load on conflicting movements with link $a$;
$\theta_a$ is fraction green time of link $a$ of total cycle time (we assume this to be fixed) and
$\tau$ is cycle time (we assume this to be fixed).

Als kruispuntenvertragingen worden opgenomen in hetverkeersmodel, wordt de reistijd niet alleen bepaald door reizen over wegen, maar ook door de vertaging die opgelopen wordt bij een kruispunt. Deze reistijden en vertragingen worden berekend door de kostenfuncties, die staan beschreven in vergelijkingen (2.79) – (2.81). Er is een kostenfunctie voor een normale weg, voor een turn op een ongeregeld en op een geregeld kruispunt. Het bijzondere aan de kostenfuncties voor turns is dat de vertraging niet alleen afhankelijk is van de eigen verkeersstroom, maar ook van de conflicterende verkeersstromen. We noemen deze kostenfuncties niet-seperabel.

The cost function of 'normal' links and the cost function of a turn at a signalized junction (assuming a fixed setting of traffic lights) are separable. Only the cost function of a turn at an unsignalized junction is non-separable, meaning that the delay depends on the load on the turn itself and also on the loads of conflicting turns.

**DIAGONAL DOMINANCE**

We know from Section 2.3 that, if the cost function is global diagonally dominant, we can guarantee there exists a globally unique equilibrium solution.

We zagen al eerder in paragraaf 2.3 dat als de kostenfunctie niet-diagonaal dominant is, dat we geen unieke oplossing kunnen garanderen. Ook constateerden we dat een realistische kruispuntvertraging inderdaad niet-diagonaal dominant is, omdat er situaties denkbaar zijn waarin de conflicterende verkeersstroom meer invloed heeft op de vertraging dan de eigen verkeersstroom. We kunnen nu verder nog stellen dat de kruispuntmodellering zoals die is geïmplementeerd in OmniTRANS ook niet perse diagonaal dominant is, en dat we het bestaan van meerdere gebruikersevenwichten in de verkeersmodellen in OmniTRANS dus niet kunnen uitsluiten.

Recall that diagonal dominance means

$$\left|\frac{\partial c_i}{\partial x_i}\right| \geq \sum_{j\neq i}\left|\frac{\partial c_i}{\partial x_j}\right|, \quad \forall i \quad \text{(row dominance)}, \tag{2.82}$$

$$\text{and} \quad \left|\frac{\partial c_i}{\partial x_i}\right| \geq \sum_{j\neq i}\left|\frac{\partial c_j}{\partial x_i}\right|, \quad \forall i \quad \text{(column dominance)}. \tag{2.83}$$

For inspecting the cost function of an unsignalized junction with respect to its diagonal dominance, we first extend the function with all the (relevant) parameters,

$$c_a(x_a) = \frac{3600}{0{,}8\left(\sigma_a - 0{,}99\sum_{b\in Y_a} x_b\right)} + 900\left(\left(\frac{x_a}{0{,}8\left(\sigma_a - 0{,}99\sum_{b\in Y_a} x_b\right)} - 1\right) + \sqrt{\left(\frac{x_a}{0{,}8\left(\sigma_a - 0{,}99\sum_{b\in Y_a} x_b\right)} - 1\right)^2 + 0{,}5\left(\frac{x_a}{0{,}8\left(\sigma_a - 0{,}99\sum_{b\in Y_a} x_b\right)} \middle/ 0{,}8\left(\sigma_a - 0{,}99\sum_{b\in Y_a} x_b\right)\right)}\right). \tag{2.84}$$

This function is plotted in Figure 15, the $x$ and $y$ axis are respectively the load on the turn itself $x_a$ and the load on all the conflicting turns $\sum_{b\in Y_a} x_b$. The load is measured in vehicles per hour, and $\sigma_a$ = 1000 vehicles per hour.

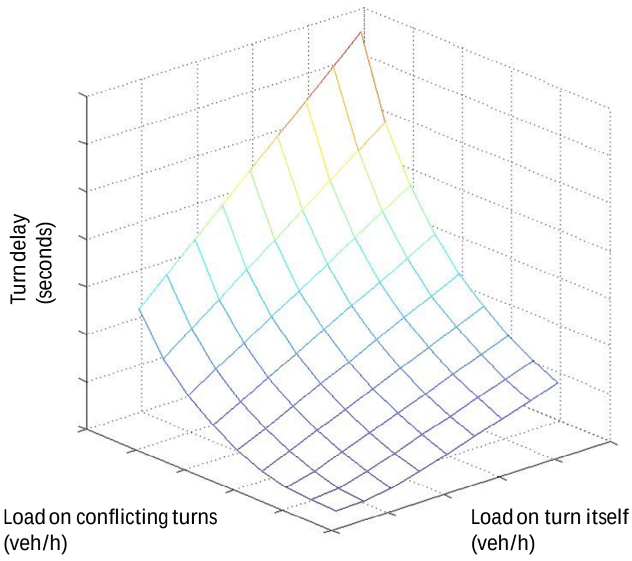



**FIGURE 15: PLOT OF TURN COST FUNCTION**

Note that diagonal dominance means in particular that the slope in the $x$ direction must be greater than the slope in the $y$ direction, in every point. This is not the case, so the cost function is not diagonally dominant. Therefore, in the current situation, we cannot guarantee a unique solution exists, even locally uniqueness is not guaranteed.

The formulas of $\frac{\partial c_i}{\partial x_i}$ and $\frac{\partial c_i}{\partial x_j}$ are also added in Appendix II.

**ASYMMETRICAL COST FUNCTION**

In Section 2.2.2, we stated that if the cost function is asymmetric, the Beckmann formulation is not defined, so a VI formulation of the TAP is needed. Recall asymmetry of the cost function means, $\exists\ i, j$ such that

$$\frac{\partial c_i}{\partial x_j} \neq \frac{\partial c_j}{\partial x_i}. \tag{2.85}$$

The cost function of a turn at an unsignalized junction in OmniTRANS, differentiated with respect to a conflicting turn is added in Appendix II as equation (A.2). It shows that this differential contains the saturation flow of the turn $\sigma_a$, and this is a value specified per turn. This means that symmetry is not guaranteed. Therefore, we cannot use the Beckmann formulation anymore and we need to switch to the VI formulation.

De kostenfunctie voor een turn zoals geïmplementeerd in OmniTRANS blijkt asymmetrisch te zijn, dus we kunnen de toedeling niet meer omschrijven als een optimalisatieprobleem. Voor een formulering van de toedeling zullen we daarom over moeten stappen naar de variationale ongelijkheid.

The problem formulation becomes: find a feasible $\bar{x}$ such that

$$c_a(\bar{x})(x - \bar{x}) \geq 0\,, \qquad \forall x \in \text{feasible set}, \tag{2.86}$$

where $c_a(x)$ is defined as in equations (2.79) - (2.81) above.

**DUALITY GAP**

Running an assignment method that converges to an equilibrium solution, when do we state the method has reached it? For this stopping criterion, the duality gap is used. We state the equilibrium solution is reached when the duality gap ($DG$) is small enough, namely $DG \leq \alpha$. Conventionally, $\alpha$ is set on $10^{-6}$.

The duality gap is defined as follows

$$DG = \frac{\sum_a c_a x_a}{\sum_{rs} \pi_{rs} d_{rs}} - 1 , \qquad (2.87)$$

where

$c_a$ is the cost of link $a$;
$x_a$ is the load on link $a$;
$\pi_{rs}$ is the optimal cost from $r$ to $s$ and
$d_{rs}$ is the demand from $r$ to $s$.

Note that when equilibrium is reached, all travellers, that is, the total demand, experience the costs of the shortest path, and so $\sum_a c_a x_a = \sum_{rs} \pi_{rs} d_{rs}$. This leads to a duality gap of 0. $\pi_{rs}$ is obtained from the last iteration, it is the cost of the shortest path at that moment from $r$ to $s$.

# 3 METHODS IN OMNITRANS

In the previous chapter implications are listed of the addition of junction modelling to the TAP. In this chapter the assignment methods implemented in OmniTRANS are discussed. In Chapter 4, we will discuss their limitations and suggest some adaptations. Later on, in Chapter 5, new possibilities for the assignment in OmniTRANS are proposed.

Currently in OmniTRANS, there are five methods for a static assignment. Thereof, two methods obtain user equilibrium, namely Frank-Wolfe (FW) algorithm and the Method of Successive Averages (MSA). Furthermore, two heuristics are implemented, namely Incremental assignment and All-Or-Nothing (AON) assignment. AON can be used as a standalone assignment, but it is also used as a module in other assignment techniques. Finally, OmniTRANS contains a system optimum assignment.

In OmniTRANS zijn er verschillende algoritmes (oplossingsmethoden) geïmplementeerd die de toedeling doen. Twee hiervan geven een gebruikersevenwicht-oplossing, dat zijn Method of Succesive Averages (MSA) en het Frank-Wolfe (FW) algoritme. In dit hoofdstuk worden de algoritmes besproken, met nadruk op het belangrijkste FW algoritme.

MSA is the most used technique by OmniTRANS users in practice. However, the FW algorithm is a widely used technique in general, not only in traffic engineering. Both methods are much alike, only one step, the line search, is different. In this study the focus is on the FW algorithm, but MSA will also be discussed.

First we will discuss the basic AON assignment, next we will zoom in on the two user equilibrium assignment techniques, the FW assignment and MSA, and we will discuss some limitations.

## 3.1 ALL-OR-NOTHING ASSIGNMENT

Many algorithms, including the FW algorithm and MSA, use the basic assignment technique All-Or-Nothing (AON) for initialization, or as a module in each iteration, in the direction finding step. AON can also be used as a standalone assignment.

The principal of this technique is the following. The shortest path for an OD-pair is calculated. The total demand is loaded on that path. This is done for every OD-pair. Thus, a path gets either all flow, or nothing. An example is given below.

#### EXAMPLE AON ASSIGNMENT

In the example network in Figure 16, the link costs are given in blue. Assume the demand is $d_{qr}$ = 20, $d_{qs}$ = 10 and $d_{rs}$ = 30. As seen in the figure, the shortest path between q and r is via t, the shortest path between q and s is via t, and the shortest path between r and s is via a direct link. When performing an AON assignment, the total demand of every OD-pair is loaded on the shortest paths, and the flow pattern in Figure 16 is obtained.

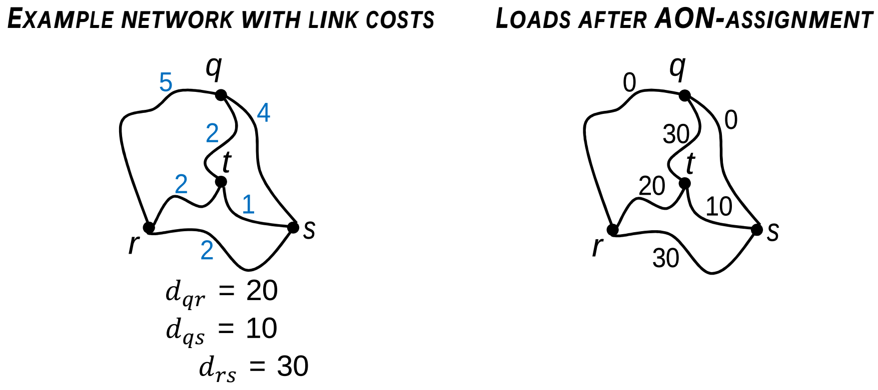



**FIGURE 16: EXAMPLE OF AON ASSIGNMENT**

## 3.2 FRANK-WOLFE ALGORITHM

One possible method to obtain a user equilibrium solution in OmniTRANS, is via the Frank-Wolfe (FW) algorithm, also known as convex combination algorithm. This is historically the conventional way of solving the TAP. It was developed by Frank and Wolfe in 1956. Many other algorithms are derivatives of the FW algorithm.

For a schematic representation of the initialization and iterations of the FW algorithm, see Figure 17. A technical description of the FW algorithm is shown in the text box. The idea of the Frank-Wolfe algorithm is as follows. In the initialization the network is loaded by an AON assignment. The costs are based on zero load. Then the first iteration starts. Every iteration consists of the following steps. First, the costs are recalculated based on the load from the previous iteration. Thereafter, an AON assignment is performed based on the new costs, this yields new loads. This is also referred to as direction finding, meaning there will be some load shifted in this 'direction'. Via a line search it is determined how much load is to be shifted in this new direction, this is also referred to as step size determination. In this line search, the objective function is minimized over all linear combinations of both the old and the new obtained load. The linear combination which gives the minimal objective function is the final flow at the end of the iteration, and the move is made. This process is repeated until a stop criterion is met.

This corresponds to a linearization at the current solution of the convex objective set in each iteration.

Het FW algoritme en MSA werken iteratief, ze komen stap voor stap bij de oplossing. In de eerste stap wordt al het verkeer op de korste route gezet. Vervolgens wordt de reistijd herberekend. De gekozen routes zullen nu dus waarschijnlijk niet meer de korste zijn, omdat er veel verkeer op rijdt. In de volgende stap wordt opnieuw de korste route gezocht, gegeven het verkeer wat nu op het netwerk staat. Dan wordt er een bepaalde hoeveelheid verkeer verplaatst naar de nieuwe korste routes. Hoeveel verkeer er wordt versplaatst is bij MSA een vaste hoeveelheid, wat vooraf is bepaald. Bij het FW algoritme wordt deze hoeveelheid berekend door een 'line search'. Dat betekent dat er verschillende hoeveelheden 'geprobeerd' worden, en degene met de minimale waarde van de doelfunctie van het optimalisatieprobleem van Beckmann wordt gekozen. Merk op dat hier direct gebruik wordt gemaakt van de formulering van het optimalisatieprobleem van Beckmann. We weten dat de oplossing waar we naar zoeken het minimum is van de doelfunctie, en we 'wandelen' daar zo stap voor stap naar toe.

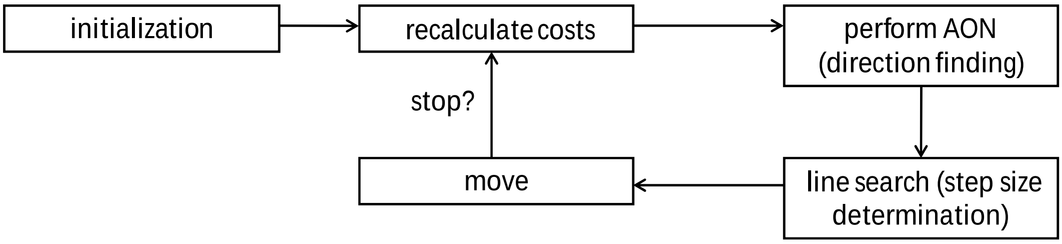



**FIGURE 17: FLOW CHART FW ALGORITHM**

**TEXTBOX 2: TECHNICAL EXPLANATION OF FW ALGORITHM**

Frank-Wolfe algorithm:

0. Initialization:
   Perform AON assignment based on ${c_a}^0 = c_a(0),\ \forall a$.
   This yields link loads $\{{x_a}^0\}$.
   Set $n = 1$.
1. Update costs on links:
   ${c_a}^n = c_a({x_a}^{n-1}),\ \forall a$.
2. Generate search direction:
   Perform AON assignment based on $\{{c_a}^n\}$.
   This yields link loads $\{{w_a}^n\}$.
3. Line search:
   Find $\lambda_n$ that solves:
   $\min_\lambda z((1-\lambda)x^{n-1} + \lambda\, w^n)$,
   with $0 \leq \lambda \leq 1$.
4. Move:
   $x^n = (1-\lambda_n)x^{n-1} + \lambda_n\, w^n$.
5. Convergence test:
   If stop criterion is met then stop,
   else set $n = n + 1$ and return to step 1.

Note that the addition of junction delays to the standard TAP, influences the FW assignment in the following manner. The links, with their loads and costs, are not only 'normal' links but also turn links. This means that the obtained loads are also loads on turns, and when the costs are recalculated, this is also done for the turn costs. The costs are, besides in the recalculation step, also in the objective function $z(x)$ used in the line search, namely

$$z(x) = \sum_a \int_0^{x_a} c_a(\omega)d\omega. \quad (3.1)$$

When adding the turn costs, this becomes rather problematic. We will come back to this in Section 3.4.

## 3.3 METHOD OF SUCCESSIVE AVERAGES

The Method of Successive Averages (MSA) is much like the FW algorithm, it only differs in the way it calculates the step size during the line search. Where the FW algorithm calculates the step size by minimizing the objective function, in MSA the step size $\lambda$ is fixed, it is set on $1/n$, where $n$ is the iteration number.

A generalization of the MSA is Volume Averaging, where $\lambda$ is any fixed value.

The convergence of MSA tends to be slow, because of its predetermined step sizes.

## 3.4 LINE SEARCH

In deze paragraaf wordt er dieper ingegaan op de line search, een belangrijke stap in het algoritme. De doelfunctie waarvan we het minimum willen vinden (want dat is gebruikersevenwicht volgens het optimalisatieprobleem van Beckmann) kunnen we zien als een 'berglandschap', zie Figuur 18. Tijdens de toedeling, dus tijdens het uitvoeren van een algoritme, 'wandel' je over dit berglandschap. Je bepaalt een dalrichting (door middel van het nieuwe korste pad te berekenen in het netwerk), in die richting moet in ieder geval een stap gezet worden. Een stap zetten correspondeerd met een bepaalde hoeveelheid verkeer verplaatsen naar het nieuwe korste pad. Om te bepalen hoe groot die stap moet zijn, bereken je voor verschillende oplossingen op de 'lijn' de waarde van de doelfunctie. Je neemt het minimum, want dan kom je het snelst in de buurt van het 'dal' van de doelfunctie.

In both the FW algorithm and MSA, the line search is an elementary step. It considers how many traffic is shifted to the new obtained shortest paths. In other words, it considers the determination of the step size, after the search direction is found. This process is illustrated in Figure 18.

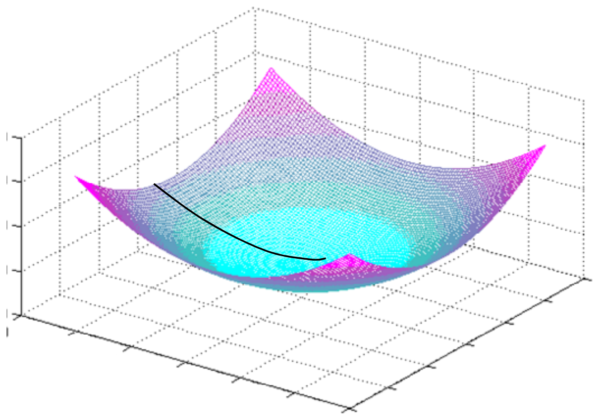


**FIGURE 18: VISUALIZATION OF LINE SEARCH IN PROBLEM SPACE**

In Figure 18, let every flow pattern $x$, that is, a set of feasible link loads, represent a point on the horizontal plane. On the vertical axis the objective function $z(x)$ is presented. The old flow pattern is the starting point of the line search, and the new flow pattern, obtained by an AON assignment, is the search direction. A move is made along the black line from the old flow pattern to the new flow pattern, this line is also represented in Figure 19.

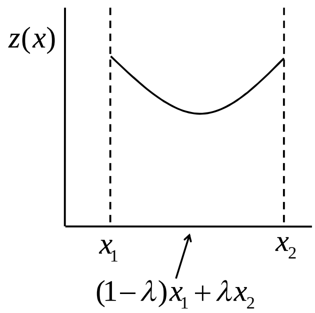



**FIGURE 19: GRAPH LINE SEARCH**

In MSA, the step size $\lambda$ is set on $1/n$, where $n$ is the iteration number. In the FW algorithm the step size is determined by minimizing the objective function. There are several ways to execute the line minimization, for example the bisection method and Newton's method. In OmniTRANS, a golden section method is implemented.

The definition of the objective function is as follows

$$z(x) = \sum_a \int_0^{x_a} c_a(\omega) d\omega, \tag{3.2}$$

and in the case of the TAP extended with junction modelling, the objective function can be separated in 'normal' links and 'turn' links as follows

$$z(x) = \sum_{\text{'normal' links } a} \int_0^{x_a} c_a(\omega) d\omega + \text{"} \sum_{\text{turns } a} \int_0^{x_a} c_a(\omega) d\omega \text{"}. \tag{3.3}$$

Note that the turns only depend on other turns and not on other 'normal' links, so the above separation can be done. The quotes are used because the latter term is rather problematic, which we will explain below.

The first part of the objective function, considering the 'normal' links, uses the integral of the BPR function. The BPR function and its integral are shown below.

$$c_a(x_a) = \frac{L_a}{v_a^{\text{max}}} \left(1 + \alpha_a \left(\frac{x_a}{q_a}\right)^{\beta_a}\right) \tag{3.4}$$

$$\int_0^{x_a} c_a(\omega) d\omega = \frac{L_a}{v_a^{\text{max}}} \left(x_a + \frac{\alpha_a}{(\beta_a + 1) {q_a}^{\beta_a}} {x_a}^{\beta_a + 1}\right) \tag{3.5}$$

The second part of the objective function, considering the turns, contains the integral of the turn cost function. From Section 2.7 we know the cost function of turns can be asymmetric and, as seen in Section 2.2.2, then there is not a function $z$ such that $\nabla z = c$, so this 'integral' in the latter part of equation (3.3) does not exist. In OmniTRANS an 'approximation' of the 'objective function' is calculated as follows

$$\tilde{z}_1(x) = \sum_{\text{'normal' links } a} \int_0^{x_a} c_a(\omega) d\omega + \sum_{\text{turns } a} c_a(x). \tag{3.6}$$

Naturally, when the ‘objective function’ does not exist, we cannot calculate or even approximate it accurately. We will discuss this limitation further in Section 4.1.

We constateerden al eerder dat het optimalisatieprobleem van Beckmann, waar het FW algoritme direct gebruik van maakt, niet meer gebruikt kan worden met de kostenfunctie van de turns, omdat die niet geïntegreerd kunnen worden. De doelfunctie, waar we overheen willen ‘wandelen’ naar het minimum toe, bestaat niet meer. We zullen dieper ingaan op deze beperking in Paragraaf 4.1.

# 4 LIMITATIONS AND ADAPTATIONS

The methods in OmniTRANS have some limitations. Firstly, the FW algorithm uses the objective function in the line search, which does not exist in the asymmetric case. Therefore, the FW algorithm does not necessarily converge to equilibrium. Secondly, the turn costs may not be diagonally dominant, so there may exist multiple equilibria. This is usually not desirable, for example when comparing scenarios. These two limitations are discussed in this chapter, including some suggestions for adaptations to (partly) overcome these limitations.

For showing results in this chapter, two networks are used, namely the network of Dutch city Delft and the network of the Belgium city Leuven. Both cities have approximately 100,000 citizens. The network of Leuven is implemented in OmniTRANS with more surroundings than the network of Delft, which can be seen in Table 2.

**TABLE 2: CHARACTERISTICS OF THE NETWORK**

| | # links | # nodes | # centroids |
|---|---|---|---|
| Delft | 1378 | 470 | 25 |
| Leuven | 5148 | 1733 | 430 |

## 4.1 OBJECTIVE FUNCTION NOT DEFINED

As discussed in Section 3.4, the FW algorithm uses an approximation of the 'objective function' of the Beckmann formulation in the line search, because the 'objective function' does not exist. This influences the convergence of the FW algorithm, as will be discussed in this section.

When performing an FW assignment with junction modelling in OmniTRANS, it shows the method returns a solution with a duality gap not close to zero. We know that at user equilibrium, the duality gap is zero. Recall that the duality gap is defined as follows

$$DG = \frac{\sum_a c_a x_a}{\sum_{rs} \pi_{rs} \, d_{rs}} - 1 \, , \tag{4.1}$$

where

$c_a$ is the cost of link $a$;
$x_a$ is the load on link $a$;
$\pi_{rs}$ is the optimal cost from $r$ to $s$ and
$d_{rs}$ is the demand from $r$ to $s$.

At user equilibrium, all used routes have the same travel time, and all unused routes are at least as long as the used routes. So, every traveller is experiencing the optimal travel cost. This implies $\sum_a c_a x_a = \sum_{rs} \pi_{rs} d_{rs}$, and so the duality gap will be zero. Without loss of generality, this holds for local equilibrium solutions. So even when convexity of the problem is not guaranteed, we know a (local) equilibrium solution corresponds to a duality gap of zero. Knowing a solution has a non-zero duality gap, means that user equilibrium is not yet reached.

Omdat de kostenfunctie van een turn nu eenmaal asymmetrisch is zoals deze in OmniTRANS (realistisch) is geïmplementeerd, kunnen we de toedeling niet meer beschouwen als een optimalisatieprobleem. De meeste algoritmes zijn wel gebaseerd op het optimalisatieprobleem, ze gebruiken direct de doelfunctie en 'wandelen' naar het minimum. Het FW algoritme in OmniTRANS is daar ook op gebaseerd. Het andere algoritme voor de toedeling, MSA, gebruikt de doelfunctie niet, en werkt dus prima. Bij het FW algoritme wordt er, bij gebrek aan de doelfunctie, een 'benadering' gebruikt voor de doelfunctie. Het blijft een benadering van iets wat niet bestaat, en dat kan natuurlijk nooit helemaal kloppen. Daardoor worden er oplossingen gevonden die geen gebruikersevenwicht zijn, met andere woorden, het FW algoritme levert niet de juiste oplossing.

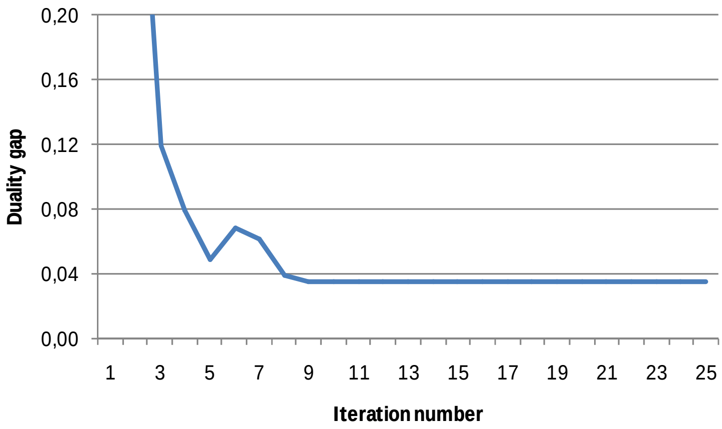



**FIGURE 20: CONVERGENCE OF FW ALGORITHM IN DELFT**

For example, in the network of Delft, the FW assignment with junction modelling returns a solution with a duality gap of 0.0350, see Figure 20. It shows that convergence of the FW algorithm is not guaranteed. At some point in the determination of the search direction or in the line search, the FW algorithm fails to approach equilibrium. As discussed in Section 3.4, in the implementation of the FW algorithm in OmniTRANS, the 'objective function' is approximated by

$$\tilde{z}_1(x) = \sum_{\text{'normal' links } a} \int_0^{x_a} c_a(\omega)d\omega + \sum_{\text{turns } a} c_a(x). \quad (4.2)$$

Naturally, this is not accurate, but at least it gives an explicit value. This $\tilde{z}_1(x)$ is used in the line search, for the step size determination. Graphs of the line search, during some iterations of the FW assignment in the network of Delft, are presented in Figure 21.

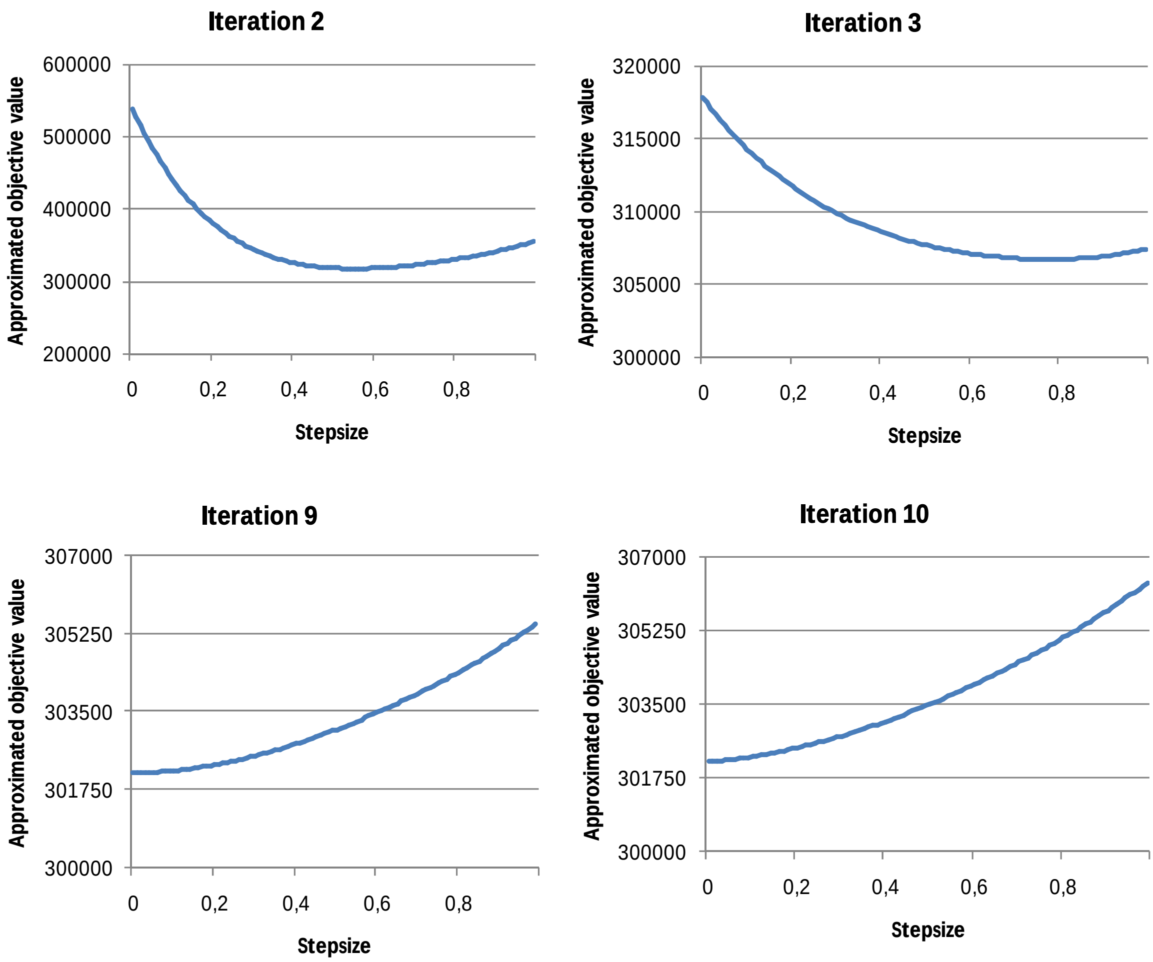



**FIGURE 21: GRAPHS OF LINE SEARCH DURING FW ASSIGNMENT IN DELFT**

Figure 21 shows that in early iterations, the minimum is ´halfway´ the line, and leads to a significant move in the search direction. But at the end of the process the minimum coincides with the old solution, so the move will be zero. In practice a negligible small move is made, for numerical reasons.

Assuming the cost functions to be symmetric, then the objective function is defined, although not always explicitly available. With that assumption we can proof the FW algorithm always finds a descent direction, when equilibrium is not yet reached. Even when the problem is not convex, this theorem holds. For the theorem and proof, see Textbox 3.

**TEXTBOX 3: THEOREM FW FINDS DESCENT DIRECTION**

***THEOREM***
Let $\nabla z(x) = c(x)$, and let $\nabla c(x)$ be symmetric, but not necessarily positive semi definite. Assume equilibrium is not yet reached. Then the FW algorithm finds a descent direction.

***PROOF***
For the determination of the search direction in the FW algorithm, an All-Or-Nothing assignment is performed. Let $w^n$ be the result of the All-Or-Nothing assignment, then the search direction is $w^n - x^{n-1}$.

This corresponds to solving the linear program

$$\min \quad c(x^{n-1})^T(w^n - x^{n-1}),$$
$$\text{subject to} \quad w^n \text{ is feasible.}$$

Since $x^{n-1}$ is not yet equilibrium, $w^n$ must satisfy the Variational Inequality

$$c(x^{n-1})^T(w^n - x^{n-1}) < 0.$$

Using Taylor series

$$f(x+h) \approx f(x) + \nabla f(x) \cdot h + O(\nabla^2 f(x))$$

and using the gradient of the objective function

$$\nabla z(x) = c(x)$$

we get

$$\begin{aligned} & z\left(x^{n-1} + \lambda(w^n - x^{n-1})\right) \\ & \approx z(x^{n-1}) + \nabla z(x^{n-1})\lambda(w^n - x^{n-1}) \\ & = z(x^{n-1}) + \lambda c(x^{n-1})^T(w^n - x^{n-1}). \end{aligned}$$

So

$$z\left(x^{n-1} + \lambda(w^n - x^{n-1})\right) < z(x^{n-1})$$

for $\lambda$ small enough. So the new solution corresponds to a smaller objective value then the old solution, so the search direction is a descent direction.

■

It is not possible to make statements about the descent direction and a minimum of a function that does not exist. But it is certain that the method stops at a point that is not yet equilibrium. This can be caused by at least two reasons. On the one hand, the search direction may not be a direction towards equilibrium. On the other hand, the current approximation of the 'objective value' $\tilde{z}_1(x)$ may lead to inaccuracy with respect to the step size determination. This is plausible, since in the end of the process equilibrium is not yet reached, and still no move is made.

A suggestion for an improved calculation which approximates the 'objective function' is given in the next section.

### 4.1.1 ADAPTATIONS

We improved the approximation of the 'objective function'. It remains an approximation of a function that does not exist, therefore the returned solution is not accurate. But with this adaptation the returned solution will have a lower duality gap, and is therefore 'closer to' an equilibrium solution.

This adaptation can be easily implemented (adding a few lines in the code of OmniTRANS, see Appendix III). For totally accurate methods that converge to an equilibrium solution, we refer to Chapter 5. Note that the methods in Chapter 5 need rigorous new implementations.

We changed the current calculation of the objective function

$$\tilde{z}_1(x) = \sum_{\text{'normal' links } a} \int_0^{x_a} c_a(\omega)d\omega + \sum_{\text{turns } a} c_a(x) \quad (4.3)$$

to

$$\tilde{z}_2(x) = \sum_{\text{'normal' links } a} \int_0^{x_a} c_a(\omega)d\omega + \sum_{\text{turns } a} c_a(x) \cdot x_a. \quad (4.4)$$

That is, we multiplied the turn costs by the load. The blue surface in Figure 22(A) is a visualisation of the 'real' value of the integral, which does not exist. Note that the cost function is plotted on one dimension, namely $x_a$, where actually it is a function of all links. The current calculation $\tilde{z}_1(x)$ as in equation (4.3) and the suggested improved calculation $\tilde{z}_2(x)$ as in equation (4.4) are visualized as surfaces in respectively Figure 22(B) and (C).

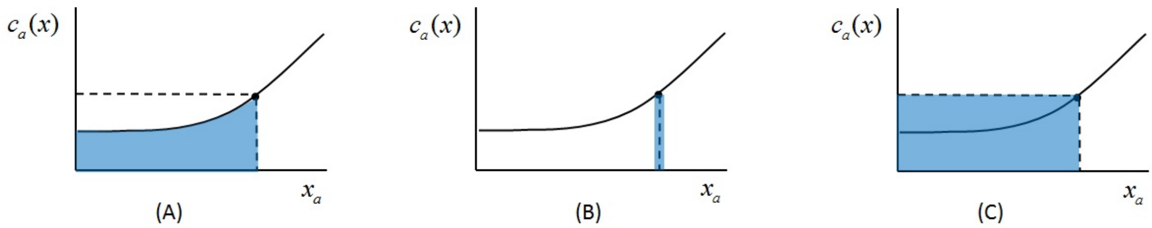



**FIGURE 22: (A) VISUALISATION OF REAL INTEGRAL (B) CURRENT CALCUATION (C) IMPROVED CALCULATION**

De doelfunctie uit het optimalisatieprobleem bestaat niet meer. Als hij wel had bestaan, had hij de waarde gehad van de oppervlakte onder de grafiek van de kostenfunctie, zoals weergegeven in Figuur 22(A). Momenteel heeft OmniTRANS het 'niet-bestaan' van de integraal opgelost door de waarde van de functie zelf te nemen. Dit correspondeert met de oppervlakte van het blauwe vlakje zoals weergegeven in Figuur 22(B). Het lijkt aannemelijk dat een berekening die correspondeert met de oppervlakte van het blauwe vlak in Figuur 22(C) een betere benadering is van de 'echte' waarde van de doelfunctie. Dit is geïmplementeerd, en het blijkt dat er met deze alternatieve berekening inderdaad oplossingen gevonden worden die 'dichterbij' gebruikersevenwicht liggen. Dit is in ieder geval een vooruitgang ten opzichte van de huidige implementatie in OmniTRANS, maar nog steeds zijn de oplossingen nog niet precies gebruikersevenwicht. Dit is te zien in Figuren 23 en 24.

In the current calculation the turn costs are negligible small compared to the integrated link costs. With this improvement, the turn costs are at least 'participating' in the line search, although not in a totally accurate way. Note that the improved calculation is more accurate when the load is small, because the function is almost constant for small loads. Also note that the turn delays actually have an upper bound, so the worst case of this approximation is within limits. The implementation in OmniTRANS of this adaptation is explained in Appendix III.

This adaption is tested in the networks of Delft and Leuven, in situations with different total demands. The FW assignment with junction delays is performed with the current and the improved calculation of the objective value during the line search. Results of the networks of Delft are shown in Figure 23 and results of the network of Leuven are shown in Figure 24.

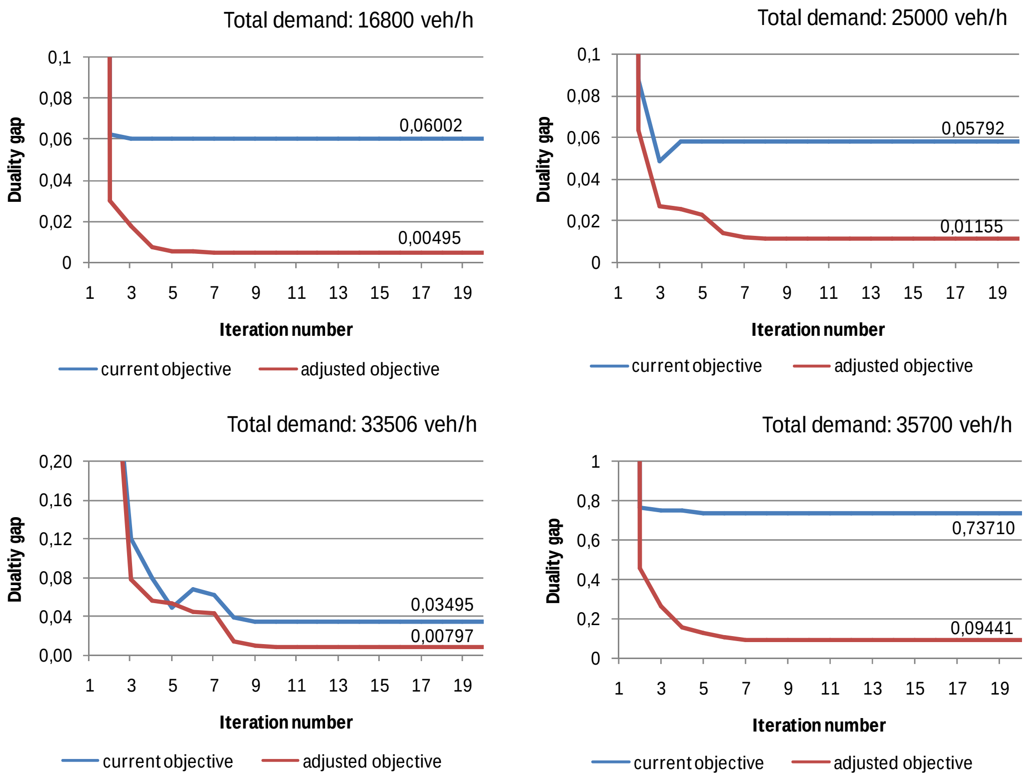



**FIGURE 23: RESULTS OF ADJUSTED OBJECTIVE IN DELFT**

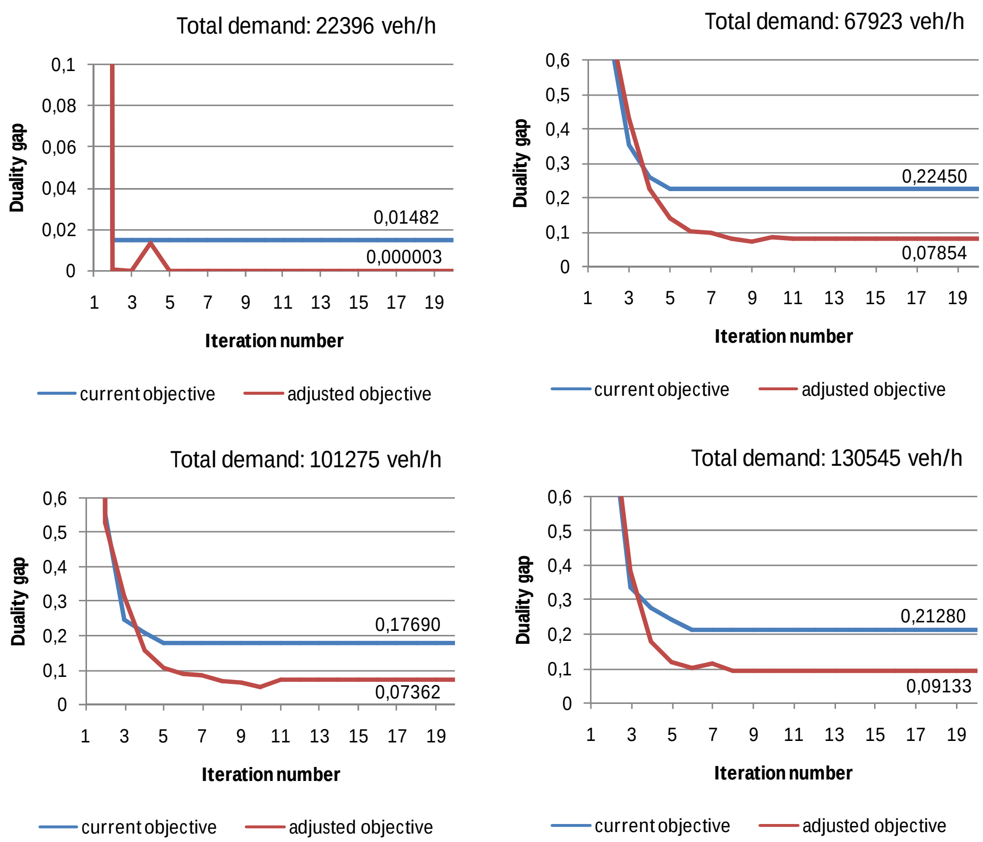



**FIGURE 24: RESULTS OF ADJUSTED OBJECTIVE IN LEUVEN**

In Figure 23 and Figure 24 one can see that the FW algorithm with the improved approximation of the 'objective function' returns solutions with a lower duality gap compared to the current approximation of the 'objective function'. Note that the improvement of the duality gap of the solution increases with a lower demand, see Table 3.

**TABLE 3: IMPROVEMENT OF THE DUALITY GAP OF SOLUTION AFTER ADJUSTMENT OBJECTIVE**

| ***Delft*** | | ***Leuven*** | |
|---|---|---|---|
| **Total demand** | **Improvement of Duality Gap** | **Total demand** | **Improvement of Duality Gap** |
| 16800 veh/h | 0.06002/0.00495 ≈ 12.1 | 2396 veh/h | 0.01482/0.000003 ≈ 4940 |
| 25000 veh/h | 0.05792/0.01155 ≈ 5.0 | 67923 veh/h | 0.22450/0.07854 ≈ 2.9 |
| 33506 veh/h | 0.03498/0.00797 ≈ 4.4 | 101275 veh/h | 0.17690/0.07362 ≈ 2.4 |
| 35700 veh/h | 0.73710/0.09441 ≈ 7.8 | 130545 veh/h | 0.21280/0.09133 ≈ 2.3 |

This can be accounted to the fact that the improved approximation of the 'objective function' is more accurate with a lower demand, as can be seen in Figure 22(C). There is one exception to this pattern, namely in Delft, with a high total demand of 35700 vehicles per hour. Here the improvement has increased. The demand of 35700 vehicles per hour is very high, since a normal morning peek hour has a total demand of 33506 vehicles per hour. When there is a high amount of congestion, the network can be in a 'gridlock' state, and several effects occur. For example, the BPR-function returns unrealistically high costs when the intensity / capacity ratio is higher than one. Also, the junction delay may be set to the maximum value. A combination of these effects makes the improvement of the duality gap harder to evaluate. Still, with realistic demands we observe that the lower the demand, the better the improvement.

Note that, when the turn costs would be a constant value, the improved approximation of the objective value would be accurate. This is shown in Figure 25 and Figure 26.

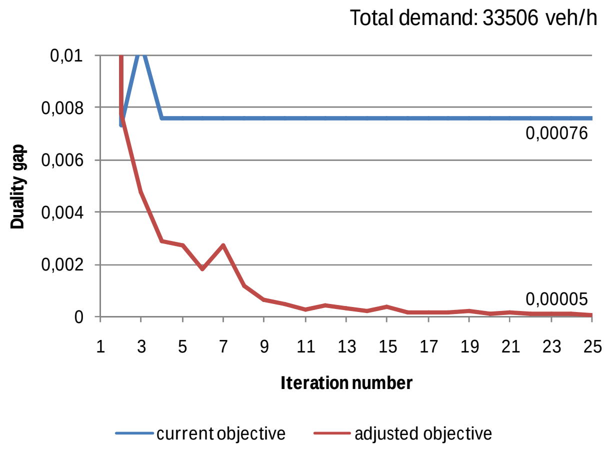



**FIGURE 25: FW ASSIGNMENT WITH FIXED JUNCTION DELAYS IN DELFT**

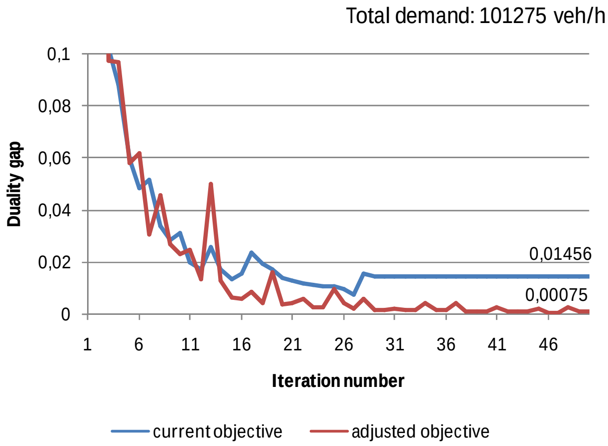



**FIGURE 26: FW ASSIGNMENT WITH FIXED JUNCTION DELAYS IN LEUVEN**

## 4.2 NON-DIAGONALLY DOMINANT TURN COSTS

Multiple equilibria may exist, because the turn costs may be non-diagonally dominant, as stated in Section 2.7. This is not desirable, especially when comparing different scenarios with the traffic model. For example, a government may be interested in the influence of construction work on a specific road, and it wants to compare the traffic flow in the normal situation with the traffic flow when certain roads are closed. Then it is important that the difference in the traffic flows are due to the construction work, and not to the fact that the obtained solutions are far apart 'by coincidence', and that there may be other pairs of solutions which are more similar. To make an honest comparison, a unique minimum is required.

Theoretically we know that there may be situations where the turn costs are not diagonally dominant. In order to examine the urge of this problem, we investigate turns in a real network with realistic loads, to find out if there exist turn cost functions which indeed appear to be non diagonally dominant, and to what extent.

We hebben al eerder geconstateerd dat we een unieke oplossing niet kunnen garanderen in de verkeersmodellen van OmniTRANS. Dit komt omdat de kostenfunctie op een turn niet altijd diagonaal dominant is. Het is de vraag op hoeveel turns de kostenfunctie niet voldoen aan deze diagonaal dominantie. Het blijkt dat een schatting van het percentage niet-diagonaal dominante kostenfuncties 41% is, in een realistisch netwerk met realistische verkeersstromen. Dat betekent dat het aannemelijk is dat in realistische netwerken meerdere oplossingen bestaan.

For this investigation, the network of Delft is used. Recall this network contains 1378 links and 470 nodes. On most nodes junctions are defined, all types of junctions are present and in total 1923 turns are defined in these junctions. Recall the definition of (row-)diagonal dominance is $\frac{\partial c_a}{\partial x_a} \geq \sum_{b \neq a} \frac{\partial c_a}{\partial x_b}$. These differentials are not easily obtained as such, so the following approximation is used

$$\frac{c_a(x_1, \dots, x_a + 10, \dots, x_n) - c_a(x_1, \dots, x_a, \dots, x_n)}{10} \approx \tag{4.5}$$

$$\frac{\partial c_a}{\partial x_a} \geq \sum_{b \neq a} \frac{\partial c_a}{\partial x_b} \tag{4.6}$$

$$\approx \sum_{b \neq a} \frac{c_a(x_1, \dots, x_b + 10, \dots, x_n) - c_a(x_1, \dots, x_b, \dots, x_n)}{10} \tag{4.7}$$

$$= (n-1) \cdot \frac{c_a(x_1, \dots, x_b + 10, \dots, x_n) - c_a(x_1, \dots, x_b, \dots, x_n)}{10} \tag{4.8}$$

$$\geq \frac{c_a(x_1, \dots, x_b + 10, \dots, x_n) - c_a(x_1, \dots, x_b, \dots, x_n)}{10}, \tag{4.9}$$

where $n$ is the number of turns which influence the cost of turn $a$, that is, the turn itself and its conflicting turns.

Equations (4.5) and (4.7) are linear approximations of the derivative. Equations (4.7) and (4.8) are the same, because the derivatives $\frac{\partial c_a}{\partial x_b}$ are all the same for $\forall b \neq a$.

The load is incremented with 10 instead of 1, because an increase of 1 does not always influence the cost enough to note the effect. Note that the load $x_a$ usually has a value between 0 and 1500 on urban roads and a value of 1000 to 5000 on highways. This means an increase of 10 is a relatively small step.

Concluding, the number of turn cost functions that fulfil

$$\frac{c_a(x_1, \dots, x_a + 10, \dots, x_n) - c_a(x_1, \dots, x_a, \dots, x_n)}{10} \geq \frac{c_a(x_1, \dots, x_b + 10, \dots, x_n) - c_a(x_1, \dots, x_b, \dots, x_n)}{10} \quad (4.10)$$

is an approximation of an upper bound for the number diagonally dominant turns. Automatically, the number of turn cost functions that fulfil

$$\frac{c_a(x_1, \dots, x_a + 10, \dots, x_n) - c_a(x_1, \dots, x_a, \dots, x_n)}{10} < \frac{c_a(x_1, \dots, x_b + 10, \dots, x_n) - c_a(x_1, \dots, x_b, \dots, x_n)}{10} \quad (4.11)$$

is an approximation of a lower bound for the number of non-diagonally dominant turns.

All turns in the network of Delft are checked on this condition, either it will fulfil equation (4.10) or it will fulfil equation (4.11). The results are shown in the Table 4.

**TABLE 4: RESULTS OF TURNS IN DELFT**

| | |
|---|---|
| **APPROXIMATION OF UPPER BOUND OF DIAGONAL DOMINANT TURNS:** | |
| 1131 (out of 1923) | Percentage of total: 59% |
| From the turns in this category, at 812 turns the terms in equation (3.10) are equal, and at 319 turns there is a non zero difference. | |
| **APPROXIMATION OF LOWER BOUND OF NON-DIAGONAL DOMINANT TURNS:** | |
| 792 (out of 1923) | Percentage of total: 41% |

As seen in Table 4, an approximation of the lower bound of the percentage of non-diagonally dominant turns is 41%. This means that in practice a reasonable amount of turns is non-diagonally dominant. Hence, it is plausible that there exist more local minima.

Note that we only considered row-diagonal dominance, and not column-diagonal dominance. But since the row-diagonally dominance condition is harmed, this is enough evidence for the possible existence of more local minima.

In fact, in large realistic networks, the existence of more local minima can be shown. The results are given in the next section.

#### 4.2.1 THE EXISTENCE OF DIFFERENT EQUILIBRIA

The existence of different equilibria is demonstrated is two realistic networks, namely in the networks of Delft and Leuven. The equilibria are obtained by an MSA assignment. In advance, note that, when MSA is performed with a large number of iterations, MSA reaches an equilibrium satisfying the Wardrop conditions. To be sure the obtained solutions are equilibria, the number of iterations is set on 10000.

The different equilibria are obtained by different initializations. When performing an assignment in OmniTRANS, an initial load can be set. Normally, the initial load is zero, but it is possible to set the initial load to the result of another assignment. This initial load will only influence the first iteration, and the initial load will not remain in the network.

We have studied the results of MSA with different initializations, in both the networks of Delft and Leuven. Different equilibria are 'easily' obtained. For example, in Delft ten different initializations are used, and this resulted in six different equilibria. So some initializations resulted in the same equilibrium. In Leuven four different initializations are used, this resulted in four different equilibria.

Niet alleen is het aannemlijk dat er verschillende oplossingen bestaan in realistische netwerken, sterker nog, er zijn vrij makkelijk meerdere oplossingen te vinden. Dit is getest in de netwerken van Delft en Leuven. De verschillende oplossingen werden gevonden door verschillende initialisaties (startpunten) van het algoritme. Meer dan de helft van de verschillende initialisaties resulteerde in een andere oplossing.
Om de verschillen tussen de oplossingen te bekijken, worden de verschillen 'geplot' in het netwerk, zie figuren 28 tot en met 31 en 33 tot en met 35. Om de verschillen goed te kunnen onderzoeken zijn de verschillen uitgedrukt in voertuigen per uur, relatieve verschillen en de maat 'RSE'.
De oplossingen die worden vergeleken zijn de meeste verschillende van alle gevonden oplossingen, dus dit zijn de 'extremen'. Het is aan de eindgebruikers van de verkeersmodellen om te bepalen of deze verschillen groot, en dus erg, zijn.

Of all obtained equilibria, the two equilibria with the largest difference are compared and examined. The differences are examined in three ways:

- the difference in load, measured in vehicles per hour:

$$x_a^j - x_a^i, \tag{4.12}$$

- the relative difference, measured in percentage with respect to one of both solutions:

$$\frac{x_a^j - x_a^i}{x_a^i}, \tag{4.13}$$

- the relative squared error (RSE), which is basically the product of the absolute and the relative difference:

$$\text{RSE} = \frac{\left(x_a^j - x_a^i\right)^2}{x_a^i}, \tag{4.14}$$

where $i$ and $j$ are the solutions to compare, and $x_a^i$ is the load on link $a$ in solution $i$. Note that the links include 'normal' links and turn links.

### Multiple Equilibria in Delft

To give an idea of the order of magnitude of the loads in Delft, a plot of the loads is given in Figure 27. These loads are obtained from a 'normal' MSA assignment, without an initial load. The loads in vehicles per hour are plotted on the links, expressed in bandwidths and colours. As the load increases, the bandwidth increases, and also the colour changes from yellow to red to blue, see the legend.

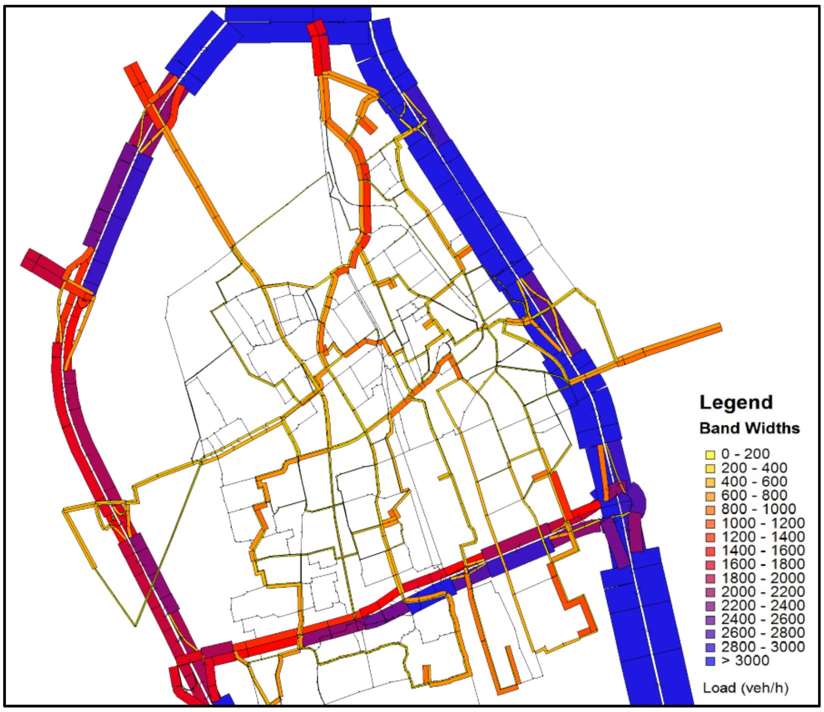



**FIGURE 27: LOADS IN DELFT**

The equilibrium solutions $i$ and $j$, based on different initializations, are to be compared. A compare plot of these solutions is shown in Figure 28. The bandwidths and the values at the links represent the difference of the loads $x_a^j - x_a^i$, in vehicles per hour. So a positive value corresponds to a higher load of solution j and has a blue colour, and a negative value corresponds to a higher load of solution $i$ and has a red colour.

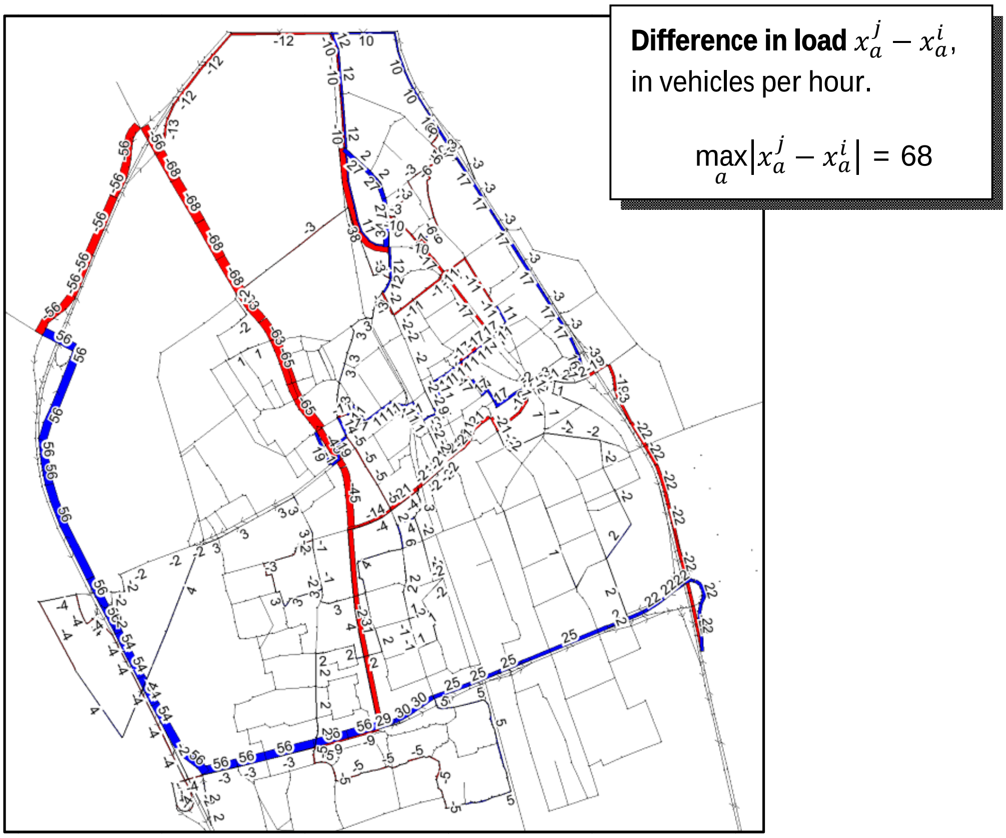



**FIGURE 28: COMPARE PLOT, DIFFERENCE IN LOAD**

As seen in Figure 28, the maximum difference is 68 vehicles per hour. To state if this is a large difference, we have to see this in perspective of the total load on that link. For example, relative to a total load of 100 vehicles per hour, a difference of 68 vehicles per hour is large, but relative to a total load of 3000 vehicles per hour, one can state the difference is negligibly small. So for that purpose, we show the same compare plot expressed in relative difference. In Figure 29, the bandwidths represent the difference in load in vehicles per hour, but the labels at the links express the relative difference.

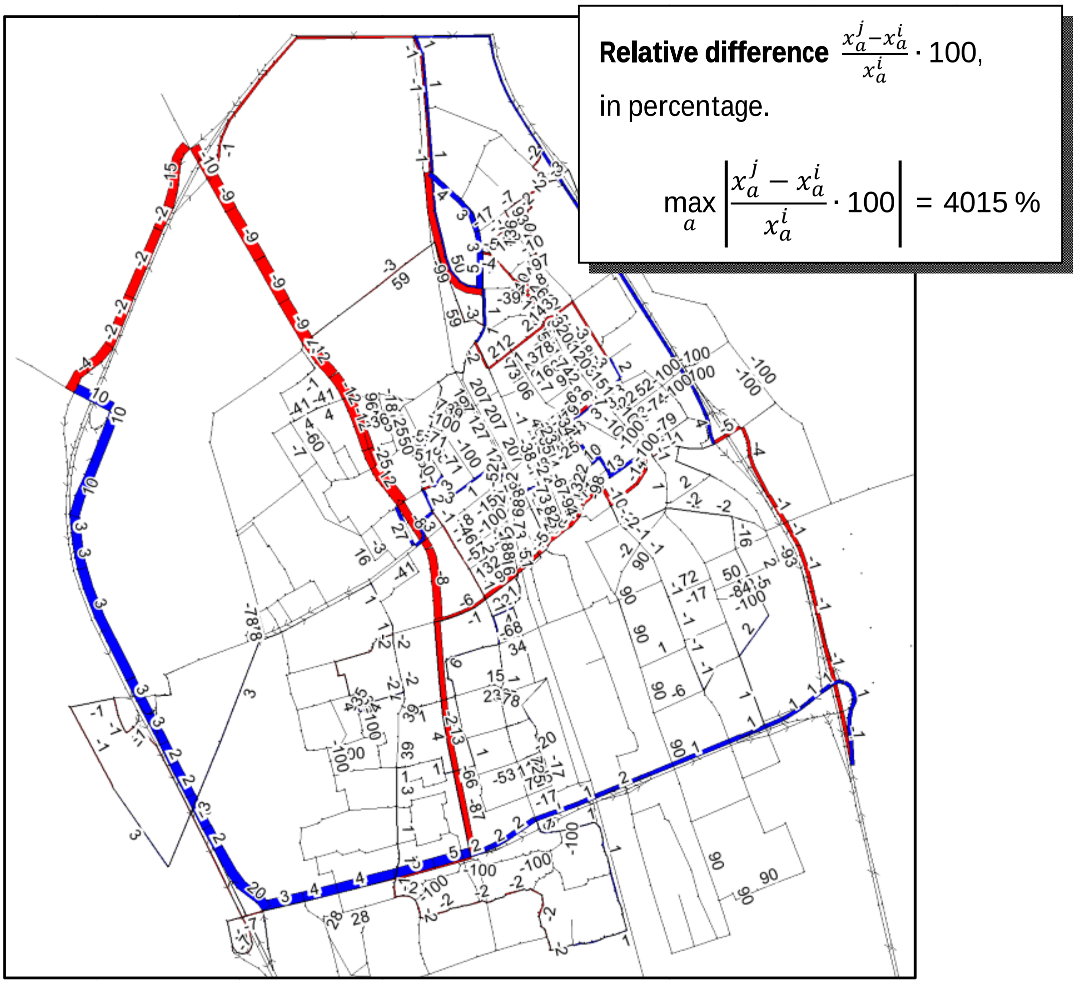



**FIGURE 29: COMPARE PLOT, RELATIVE DIFFERENCE**

This gives a relative picture of the differences. In the centre of the network, large differences are found. In Figure 30 we have zoomed in on this area.

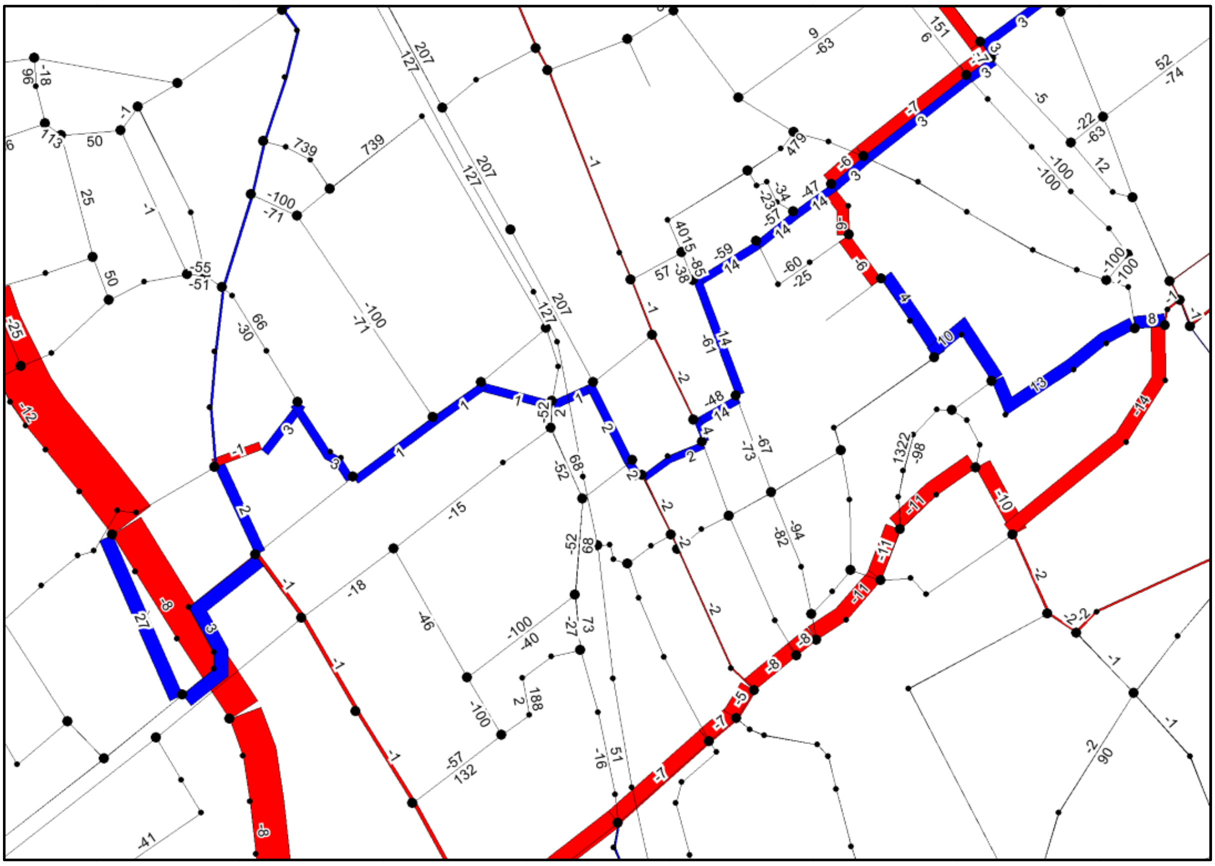


**FIGURE 30: COMPARE PLOT, RELATIVE DIFFERENCE, CENTRE OF DELFT**

In Figure 30 it becomes clear that very large relative differences are found, up to a maximum of 4015%. These large values are not necessarily striking differences, for example, a difference between 0.1 vehicles per hour and 4 vehicles per hour is not very large, although it leads to an excessive relative difference.

Concluding, both the (absolute) difference in loads and the relative difference have their advantages and disadvantages. The relative squared error (RSE)

$$\frac{\left(x_a^j - x_a^i\right)^2}{x_a^i}$$

is a measurement that combines 'the best of both worlds'. It gives a high value when the difference is high in both absolute and relative terms. In Figure 31 the RSE is plotted on the links of the network of Delft.

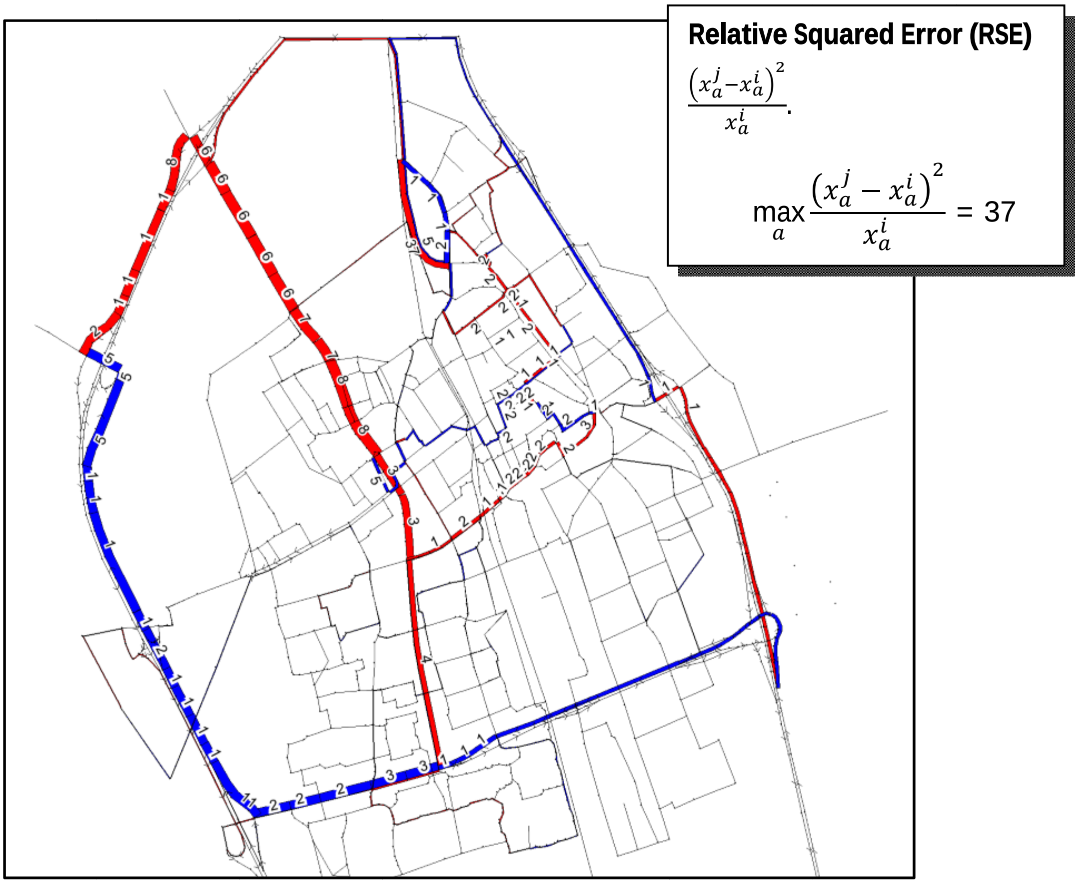



**FIGURE 31: COMPARE PLOT, RSE**

The RSE can be used to examine if the differences between the solutions are significant. For example, one can choose to set a threshold, and when it exceeds that threshold the differences are notable.

It is interesting to compare the equilibrium solutions also with respect to travel time for every OD-pair. For this a 'skim' matrix is generated for both solutions, where the optimal travel time is given for every OD-pair. No major differences are found. The mean difference found was approximately 1 second and the maximum difference was approximately 9 seconds, where the mean travel time was about 10 minutes. Concluding, the differences with respect to travel time between the two equilibrium solutions are negligibly small.

### MULTIPLE EQUILIBRIA IN LEUVEN

In the network of Leuven a comparison of solutions is made in the same manner as in the network of Delft. So plots are given of the differences in load, the relative differences and the RSE.

First, a plot of the loads of a ´normal´ MSA assignment is given in Figure 32, to give an idea of the order of magnitude of the loads in Leuven. The loads in vehicles per hour are plotted on the links, expressed in bandwidths and colours. As the load increases, the bandwidth increases, and also the colour changes from yellow to red to blue, see the legend.


**FIGURE 32: LOADS IN LEUVEN**

The equilibrium solutions $i$ and $j$, based on different initializations, are to be compared. A compare plot of these solutions is shown in Figure 33. The bandwidths and the values at the links represent the difference of the loads $x_a^j - x_a^i$, in vehicles per hour. So a positive value corresponds to a higher load of solution j and has a blue colour, and a negative value corresponds to a higher load of solution $i$ and has a red colour.

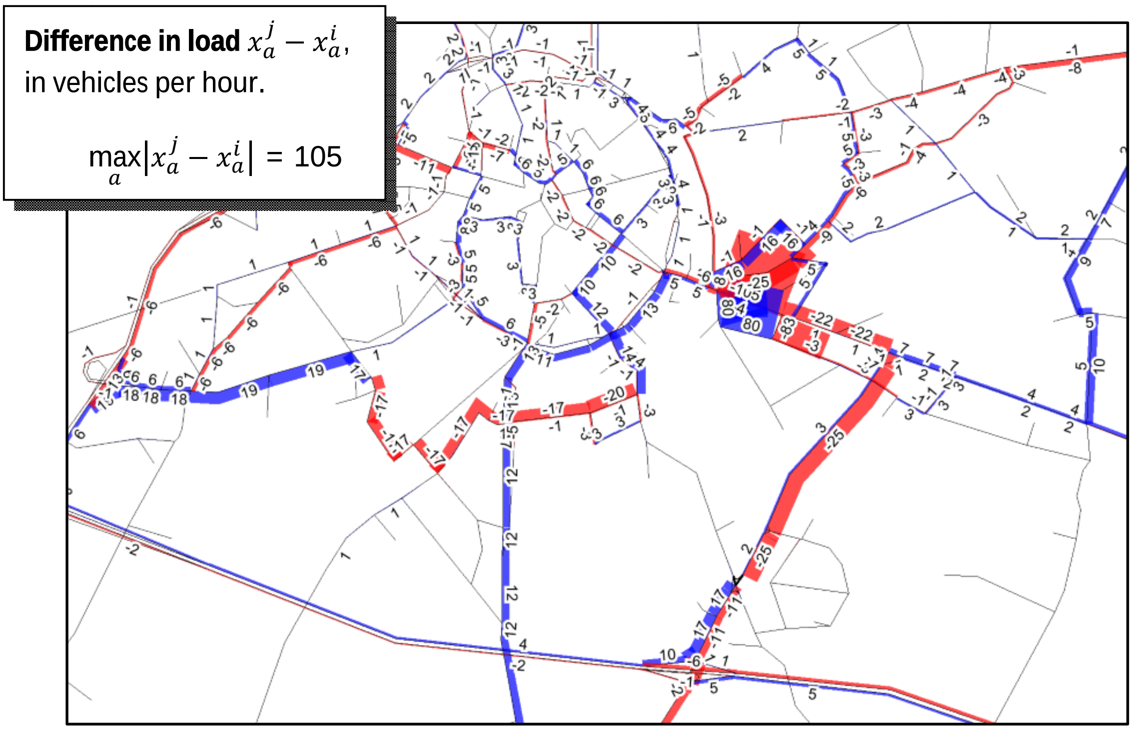



**FIGURE 33: COMPARE PLOT, DIFFERENCE IN LOAD**

Again, as discussed in the previous section, the difference in load in vehicles per hour does not give sufficient information. Therefore, also the relative differences are given in Figure 34. The best expression of the difference is in terms of the RSE, as plotted in Figure 35.

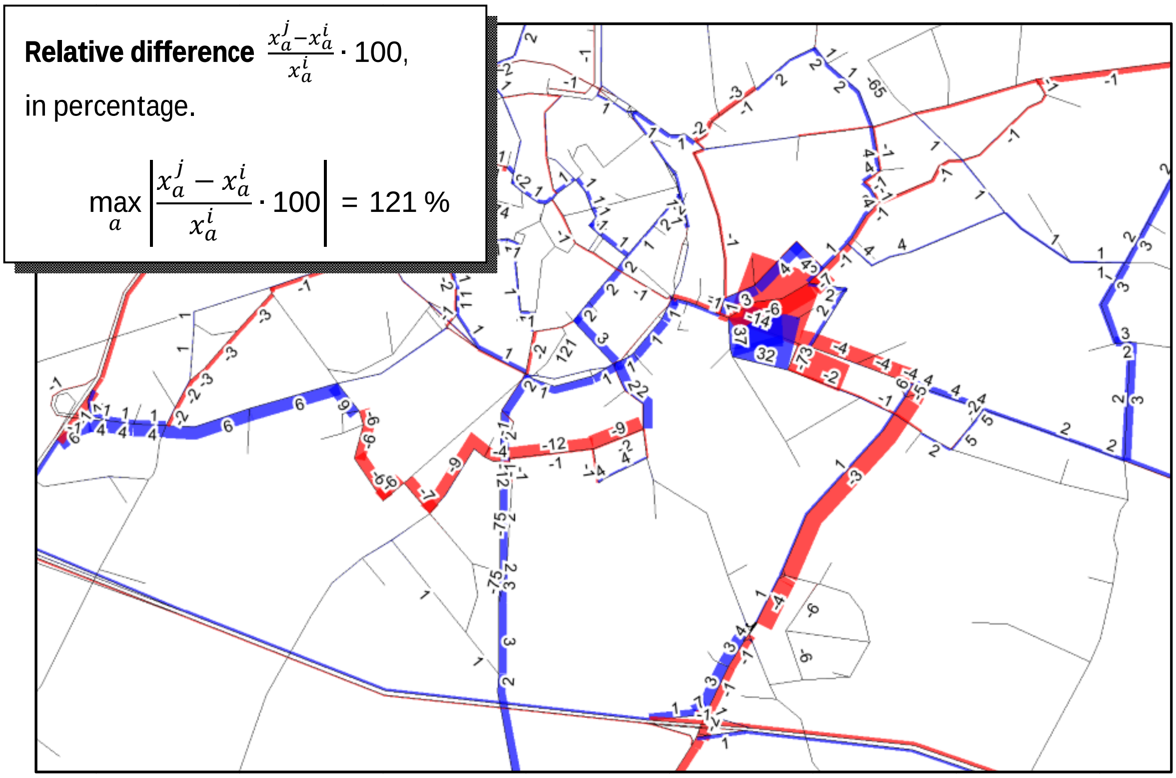



FIGURE 34: COMPARE PLOT, RELATIVE DIFFERENCE

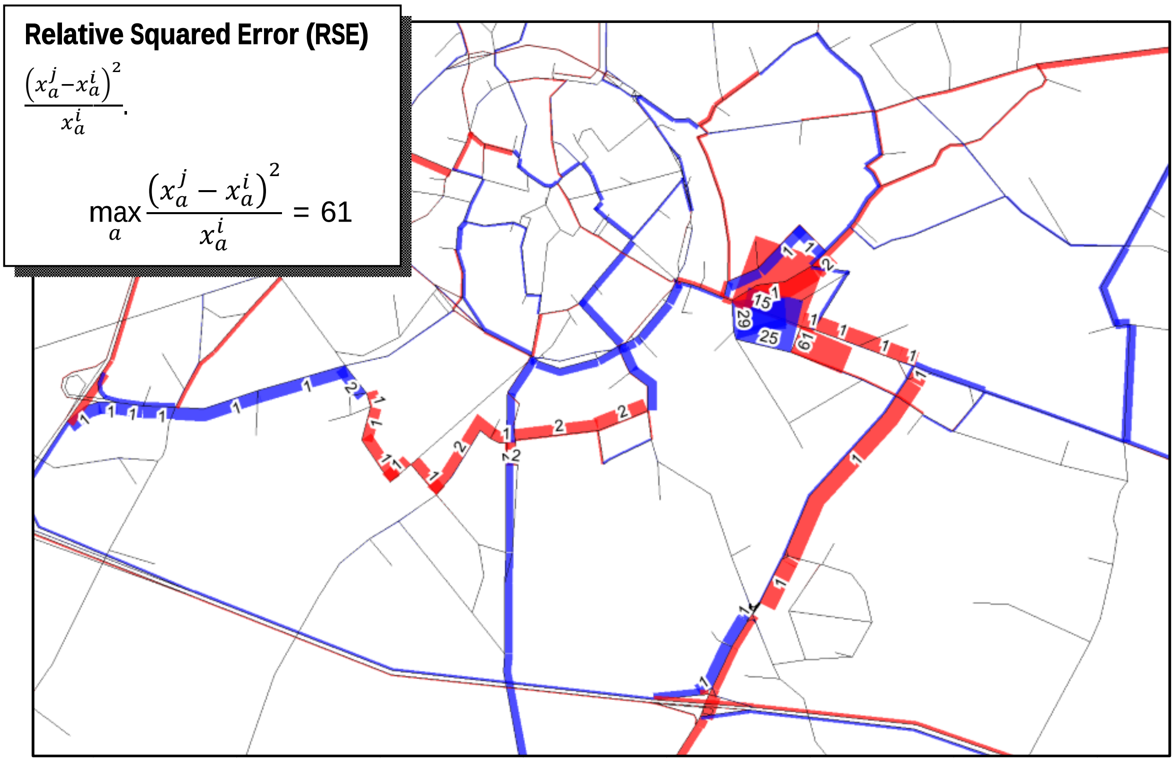



FIGURE 35: COMPARE PLOT, RSE

The results show that in realistic networks multiple equilibria exist. The differences between these equilibria can be significant, although it is up to the user of the traffic model to judge these differences.

### 4.2.2 ADAPTATIONS

In the previous section the results show that there indeed exist several local minima in realistic networks. This can be a problem, especially when the aim is to compare scenarios with the traffic model.

If one values a realistic junction modelling when performing an assignment, the existence of several local minima remains a problem. When one requires a unique solution, there are several options. One option is drop the junction modelling completely, but then obviously the advantages of the junction modelling are eliminated. Another option is to use fixed junction delays, based on an assignment on an 'average' model. The last option is to use a heuristic, which 'forces' the assignment to pick a certain minimum. When this is done with a consistent strategy, it allows one to make a 'honest' comparison of solutions for comparing scenarios. The latter two options are discussed below.

#### FIXED JUNCTION DELAYS

When the junction delays are fixed, the delay has always the same value, independent of the load on the junction. The values for the junction delays can be obtained from junction delays in a solution of an assignment on an 'average' model. With fixed junction delays, the turn cost functions become separable, therefore the objective function of the Beckmann transformation is existent and defined, and the problem becomes convex. Because of the convexity, the obtained solution is unique.

In practice, fixed junction delays can lead to unrealistic junction delays. For example when comparing scenarios, the values for the delays on junctions near the intervention, for example near an extra road, may be unrealistic. This problem can be overcome by making the delays on junctions near the intervention variable, by using the normal calculation of the junction delays. Note that, in that case, the uniqueness of the solution can again not be guaranteed.

#### 'DIRECTING' THE ASSIGNMENT TO A CERTAIN MINIMUM

The problem space can be temporarily 'made' convex in the beginning of the process. In this heuristic the turn costs are added increasingly to the objective function, in such a way that in early iterations the turn costs 'participate' a little, and in later iterations the factor of the turn costs is growing, until the turn costs 'participate' totally. This means that in early iterations the problem space is convex, and in the final iterations the problem space is realistic, and may be not convex. An example of this changing problem space is visualized in Figure 36.

This way, the obtained solution will be forced to stay close to the minimum of the convex situation. It is still questionable if the obtained solution of this heuristic is the 'good' minimum. There may be for example two minima both close to the solution of the convex situation, and it may be still arbitrary which of these two minima will be 'picked'. But on the other hand, this heuristic 'directs' to a certain minimum, and when this is done consistent in both scenarios, at least a more honest comparison can be made than without this heuristic.

This heuristic can be implemented by increasingly add the turn costs to the objective function during the assignment. For example, let parameter $\mu$ grow from zero to one, by setting $\mu = \frac{n}{N}$, where $n$ is the iteration number, and $N$ is the total number of iterations. The 'objective function' will be calculated by

$$\tilde{z}_3^m(x) = \sum_{\text{'normal' links } a} \int_0^{x_a} c_a(\omega)d\omega + \mu \cdot \sum_{\text{turns } a} \int_0^{x_a} \tilde{c}_a^m(\omega)d\omega. \tag{A.1}$$

Recall from Section 4.1 that the normal objective function does not exist, so we use this 'approximation'. We use the 'diagonalized' cost function of the turns, where the load on the conflicting turns is fixed, based on an previous iteration $m$. This can be for example the last iteration. This 'diagonalization' will be extensively explained in Section 5.1. The diagonalized costs are defined as

$$\tilde{c}_a^m(x_a) = c_a\left(x_1^m, x_2^m, \dots, x_{a-1}^m, x_a, x_{a+1}^m, \dots, x_{|A|}^m\right). \tag{A.2}$$

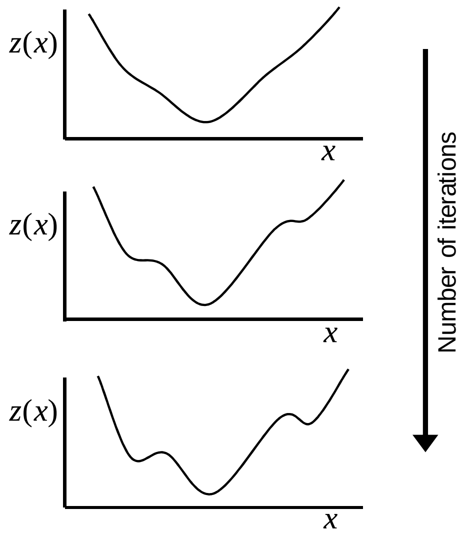



**FIGURE 36: CHANGING PROBLEM SPACE**

# 5 New Solution Methods for the TAP with Junction Modelling

In Chapter 4 we have seen that the current methods in OmniTRANS, for solving the TAP with junction delays, have their limitations. Convergence of the FW algorithm is not guaranteed. Furthermore, MSA tends to converge slowly, because of its predetermined step sizes. In this chapter new possibilities for methods are discussed, which will converge accurately to local user equilibrium and can cope with junction modelling.

In general we can state an algorithm is needed, which can cope with non-separable asymmetrical costs. This implies the objective function of the Beckmann formulation can no longer be used as such.

First, we will propose a modification of the FW algorithm, which is accurately converging to equilibrium, because the cost functions are 'diagonalized'. After, we will list several other methods which have potential for solving the TAP with junction modelling accurately, for example based on the Variational Inequality formulation.

## 5.1 Diagonalization

The first method we will propose is the Diagonalization Algorithm (DA) (Florian & Spiess, 1982). The DA is a variant on the FW algorithm, where the off-diagonal elements of the cost function are fixed in the sub problem of minimizing the objective function. Because the off-diagonal elements are fixed, the cost function becomes separable, and the objective function is explicitly defined. The idea of the Diagonalization Algorithm (DA) is as follows.

Het Diagonalisatie Algoritme is een oplossingsmethode voor de toedeling die wel is gebaseerd op het optimalisatieprobleem, maar toch geen 'last' heeft van de doelfunctie die niet bestaat. Dit komt omdat dit algoritme voorafgaand aan de 'line search' waarin die doelfunctie wordt gebruikt, de conflicterende verkeersstromen 'vastzet'. Daardoor worden de kostenfuncties seperabel, en kunnen ze gewoon geïntegreerd worden in de doelfunctie. De doelfunctie bestaat op deze manier gewoon, en kan gebruikt worden. Je maakt hiermee natuurlijk een kleine fout, door de conflicterende verkeersstromen als vast te beschouwen, maar na elke stap 'update' je deze verkeersstromen weer. Zo kom je met deze oplossingsmethode toch mooi in gebruikersevenwicht terecht.

First the off-diagonal elements are fixed, based on a feasible flow at an iteration $m$. This means the influence of the conflicting turns is fixed, so when at iteration $m$ the load on a conflicting turn is 1000 vehicles per hour, it remains that way. Let

$$\tilde{c}_a^m(x_a) = c_a\left(x_1^m, x_2^m, \dots, x_{a-1}^m, x_a x_{a+1}^m, \dots, x_{|A|}^m\right). \tag{5.1}$$

This results in a new cost function, where the costs solely depend on the load on the turn itself. The (fixed) loads of iteration $m$ are used for the loads on the conflicting turns. The new cost function can be formulated as a vector of all cost functions per link

$$c(x, x^m) = (\tilde{c}_a^m(x_a), a \in A). \tag{5.2}$$

Because the off-diagonal elements are fixed, the cost function becomes separable and symmetrical, since the Jacobian of $c(x, x^m)$ has only zeros in the off-diagonal entries. The objective function of this diagonalized problem is explicitly known, and also the gradient of the objective function is given by

$$\nabla z(x) = \tilde{c}_a^m(x_a). \tag{5.3}$$

The steps of the Diagonalization Algorithm are explained in Textbox 4.

**TEXTBOX 4: TECHNICAL EXPLANATION OF THE DIAGONALIZATION ALGORITHM**

Diagonalization Algorithm:

0. Initialization:
   Perform AON assignment based on ${c_a}^0 = c_a(0)$ , $\forall a$ .
   This yields link loads $\{{x_a}^0\}$ .
   Set = 1 .
1. Diagonalize:
   Construct $c(x, x^{n-1})$ as given in equation (5.2).
2. Solve the diagonal problem:
   For example using the FW algorithm.
   This returns new loads $x^n$.
3. Stopping criterion:
   Terminate, or return to step 1.

Dafermos (1980) proposed, amongst others, the idea of fixing the symmetric part during solving the TAP with asymmetric costs. Florian and Spiess (1982) as well as Dupuis and Darveau (1986) proved convergence of the DA.

In step 2 of the algorithm, the diagonal problem is solved. This is originally done extensively with many iterations, but can also be done with a few iterations. Sheffi (1985) presented a proof of convergence, when the diagonal problem is solved with only one iteration. He named this version of the Diagonalization Algorithm the ‘streamlined diagonalization algorithm’.

The implementation in OmniTRANS of the ´streamlined´ version of the Diagonalization Algorithm of Sheffi, is not very difficult to implement. In the two modules in OmniTRANS, ´Traffic´ and ´JunctionModelling´, some adaptations are needed. These adaptations are explained in Appendix III.

Diagonalization is a technique which can be transferred to other methods.

## 5.2 OTHER POSSIBILITIES

There are many solution methods for the TAP, most of them are variations on the FW algorithm. For an explanation of several main algorithms, see Ton (2011). A distinction can be made between link-based and path-based algorithms. Since Bar-Gera presented his Origin Based Algorithm (OBA) in 2002, a new series of algorithms is developed, called ‘bush’-based algorithms.

In this section we will first discuss the link- and path-based algorithms in general, with respect to dealing with models with junction modelling. Afterwards, we will zoom in on one path-based algorithm with high potential for OmniTRANS. Finally we will discuss the bush-based algorithms in general with respect to dealing with junction modelling.

### 5.2.1 Link-based and Path-based Algorithms

Most of the link-based and path-based algorithms are variations on the FW algorithm. Variation can be made on several topics. For example, the search direction can be optimized, or a different strategy for the minimization can be used. We have seen that the objective function in the Beckmann formulation is no longer of use, unless it is used in a 'diagonalized' problem, as explained in Section 5.1. This diagonalization may be applied to other algorithms as well. Another possibility to avoid the use of the non-existing objective function is to use the Variational Inequality formulation of the TAP. Still, the Variational Inequality is only a criterion for equilibrium, it tells us when equilibrium is reached, but it does not provide a method to find the equilibrium. For converging to the equilibrium, where usually the objective function is used, now for example the cost functions and their gradient can be used. Then, the flow is shifted from the 'expensive paths' to 'cheaper paths'.

In literature, some solving methods are proposed for solving the asymmetric TAP (Dafermos, 1971, Dafermos, 1980, Fisk & Nguyen, 1982, Smith, 1983b, Nguyen & Dupuis, 1984). Most of these algorithms are described superficially and rather theoretically, and therefore are not always suitable for direct implementation. They are mostly based on general techniques from mathematical programming, for example 'cutting planes' or 'column generation'.

Other algorithms are practically described, most of them are implemented and tested, but not always suitable for the asymmetric TAP. The considered algorithms include some variations on Frank-Wolfe, for example the (Bi-)Conjugate Frank-Wolfe algorithm (Zhou & Martimo, 2009), Projected Gradient (Florian, Constantin & Florian, 2009), Simplicial Decomposition (Lawphongpanich & Hearn, 1984) and some variations, namely Restricted Simplicial Decomposition (Hearn, Lawphongpanich & Ventura, 1985) and Nonlinear Simplicial Decomposition (Larsson, Patriksson & Rydergren, 1997), and Gradient Projection (Bertsekas, 1976, Jayakrishnan, Tsai, Prashker and Rajadhyaksha, 1994). Of these algorithms, the only one that is directly suitable for the asymmetrical TAP is the specially adapted version of Simplicial Decomposition (SD).

#### Simplicial Decomposition

Originally, SD was based on the Beckmann formulation, but the version of Lawphongpanich and Hearn (1984) is adapted to the asymmetrical TAP, by using the Variational Inequality formulation. The SD algorithm generates 'extreme points' by performing an AON assignment, which represent extreme flow patterns. In every iteration the minimum is searched over the convex combination of the extreme points. It uses the VI notation for its stopping criterion. When the solution is not optimal yet, a new extreme point is added to the set of extreme points, and in the next iteration it contributes to the convex hull of extreme points where we search our solution in. So in other words, we add and possibly delete extreme flows, depending on if we need those flows to 'construct' the optimal solution. This is based on the idea of column generation, a well known technique in mathematical programming.

Note that the minimization problem is formulated as a Variational Inequality. The minimization can be executed by, for example, the Gradient Projection (GP) method (Bertsekas, 1976). Jayakrishnan, Tsai, Prashker and Rajadhyakska (1994) presented an implementation of the GP method for the TAP. In the GP method the flow is shifted from 'expensive paths' to 'cheaper paths', until equilibrium is reached.

Er zijn zeker mogelijkheden voor een volledig nieuw algoritme om de toedeling te doen in modellen met kruispuntmodellering, die goed werken en waarschijnlijk sneller zijn dan het FW algoritme of het Diagonalisatie Algoritme. De voorwaarde waar zo'n nieuw algoritme aan moet voldoen is dat het geen gebruik mag maken van de doelfunctie, tenzij dat is in een 'gediagonaliseerd' probleem zoals bij het Diagonalisatie Algoritme. Het mag dus ook geen gebruik maken van de afgeleide of het minimum van de doelfunctie, want als de functie niet bestaat, bestaan deze kenmerken ook niet. Alternatief 'gereedschap' voor een algoritme is de variationele ongelijkheid die gebruikt kan worden om te constateren of gebruikersevenwicht is bereikt. Verder kunnen de kostenfuncties zelf gebruikt worden, en hun afgeleide. Dit kan informatie geven over dure en goedkopere routes, zodat verkeer van dure naar goedkopere routes verplaatst kan worden. Dit is een andere manier om 'naar het gebruikersevenwicht te wandelen'. Een voorbeeld van een algoritme wat deze technieken gebruikt, en veelbelovend is als het gaat om snelheid, is Simplicial Decomposition.

### 5.2.2 BUSH-BASED ALGORITHMS

In 2002 Bar-Gera presented his origin-based algorithm (OBA), which initialized a new series of algorithms, namely the bush-based algorithms. Of these algorithms, little is known of their applicability to models with junction modelling. Bar-Gera states: "In the definition of the algorithm only link-cost function and their derivatives are used; therefore, the algorithm can be applied as is to any general cost structure including non-separable asymmetric costs. Theoretical convergence in the latter case, however, is not necessarily guaranteed." (Bar-Gera, 2002, p. 399) Since the bush-based algorithms are relatively new, little is known about convergence in the asymmetrical case.

Other main bush-based algorithms used for solving the TAP, are Algorithm B (Dial, 2006, Slavin, Brandon & Rabinowicz, 2006), Linear User Cost Equilibrium (LUCE) (Gentile, 2009) and Traffic Assignment by Paired Alternative Segments (TAPAS) (Bar-Gera, 2010).
Nie (2010) investigated the class of bush-based algorithms and distinguished two possibilities for solving the restricted master problem, 'all-at-once' or 'one-at-a-time' (Nie, 2010, p. 82). The restricted master problem is the assignment problem for one 'bush', that is for example for the demand from one origin. In Figure 37 'bushes' are coloured in an example network. If the restricted master problem is solved 'one-at-a-time', then first one bush is solved to equilibrium state, then the next, etcetera. It never returns to a bush when it is once solved. This is naturally not applicable to the TAP with non-separable costs, because the costs in one bush may be influenced by load from another bush, namely at junctions.

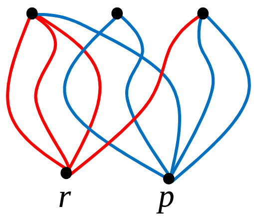



FIGURE 37: 'BUSHES' IN A NETWORK

When the restricted master problems are solved 'all-at-once', the solving process iterates over the bushes, so first in one bush one or a few iterations are performed, then in the next bush one or a few iterations are performed, etcetera. When all bushes are ‘visited’, it repeats from bush one, and it maintains iterating over all bushes until equilibrium over the total network is reached.

When using a bush-based algorithm in a model with junction modelling, at least the ‘one-at-a-time’ strategy for solving the restricted master problem will not work properly. Iterating over all bushes is necessary for convergence, so the ‘all-at-once’ strategy is to be used.

Numerical results show that bush-based algorithms perform well on efficiency in terms of calculation time. Still, one has to keep in mind non-separable costs are not taken into account in this studies, and non-separable costs may influence the efficiency rigorously.

# 6 CONCLUSION

In this study we considered the influence of the addition of junction modelling on the standard Traffic Assignment Problem (TAP), in particular with respect to the implementation in the transport modelling application OmniTRANS. The standard TAP, which we assumed to be user equilibrium based, static and deterministic, can be formulated as an optimization problem, as in equations (2.7) – (2.10), or as a Variational Inequality, as in equation (2.12). It concerns the traffic flow on a network, given the demand for all OD-pairs. We assume user equilibrium to be the goal of the TAP. This is according to Wardrops first principle (Wardrop, 1952), that is, the route choice of travellers results in a state such that for all OD-pairs all used routes have equal travel times, and all unused routes have higher (or equal) travel times. In this state, no individual can reduce his travel time, which is a realistic situation.

In urban areas, a significant portion of the travel time is incurred at junctions. Therefore, junction modelling is of great importance when calculating realistic travel times during the assignment.

When adding junction delays to the model, this changes the model and also the TAP in an essential manner. The junction nodes are expanded, and all turns become extra links. On these new links a cost function is defined as in equations (2.70) – (2.71), which gives the turn delay. This cost function is non-separable, meaning that the costs depend not only on the load on the turn itself, but also on the load on the conflicting turns. The cost functions, as implemented in OmniTRANS, show to be asymmetric and may sometimes be non-diagonal dominant. This asymmetry and the non-diagonal dominance of the cost functions influences the TAP, both effects have been investigated.

First, a significant number of cost functions appear to be non-diagonally dominant in realistic networks, and therefore the monotonicity condition is harmed, and a unique solution cannot be guaranteed. The existence of multiple solutions can be problematic when one aims to compare scenarios using the traffic model. For example when examining the traffic flow with and without a road closure, one wants to be sure the differences are due to this road closure, instead of a 'coincidence' which one of multiple possible equilibria is obtained. For making a fair comparison between two scenarios, a unique solution is needed. Actually, multiple solutions are shown in realistic networks, and they are easily obtained by different initializations. So, in realistic situations, a unique solution is certainly not obvious.

There are several possibilities to deal with this problem. Besides dropping the junction modelling, one can use fixed junction delays which do not depend on the load on the conflicting turns. A disadvantage of this approach is that it can lead to inaccurate junction delays. Also one can increasingly add the turn delays to the objective function, so that the problem space is convex in the beginning of the process, and later on, when the turn delays are fully ´participating´, the problem space is realistic but not convex. The obtained solution is then forced to be close to the minimum from the convex problem. It remains a trade off, either the junction delays are realistic but there may exist multiple solutions, or a unique solution is obtained but the junction delays are not realistic.

The second main effect of the addition of junction delays to the TAP is due to the asymmetry of the cost function. This asymmetry implies the objective function of the Beckmann formulation is no more valid. The common Frank-Wolfe algorithm as implemented in OmniTRANS, which uses an ‘approximation’ of the non-existing objective function, is not an appropriate solution method. Results show that the Frank-Wolfe algorithm obtains solutions in models with junction delays which are not equilibrium solutions. An improvement for the ‘approximation’ of the non-existing objective function is given, and although this leads to ‘better’ solutions closer to equilibrium, still no equilibrium solutions are obtained. However, the Method of Succesive Averages, implemented and named ‘Volume Averaging’ in OmniTRANS, is working properly with asymmetrical costs. That is because of its predetermined step sizes, and therefore it does not need the objective function.

Alternatives for the Frank-Wolfe algorithm are proposed. These new methods are either based on ‘diagonalized’ cost functions, or on the Variational Inequality formulation of the TAP. One of the possibilities is the Diagonalization Algorithm, which is a variation on the Frank-Wolfe algorithm. In the Diagonalization Algorithm the cost function is diagonalized, meaning that the influence of the load on the conflicting turns is temporarily fixed, and the cost function becomes separable. This sub problem can be solved using normal solving methods based on the Beckmann formulation. During the solving process the load on the conflicting turns are regularly ´updated´. This algorithm converges to an equilibrium solution, and is a good alternative for the Frank-Wolfe algorithm. Other new possibilities are based on the Variational Inequality formulation, which can be used to determine if equilibrium is reached. For the convergence towards this equilibrium the cost functions and their derivatives are used. This is done by shifting flow from ‘expensive paths’ to ‘cheaper paths’, until equilibrium is reached. A specific method that is based on these techniques is Simplicial Decomposition.

Recalling the research question, as stated in Section 1.2,

***“In the static user equilibrium-based Traffic Assignment Problem with deterministic route choice, expanded with junction delays, which algorithm converges the fastest to an accurate solution, within limited memory capacity?”***

we can conclude that the expansion of the TAP with junction delays had certain important effects. We have examined and discussed those effects extensively. For example, unexpectedly the existence of multiple solutions of the TAP with junction delays became clear. Also, the effects of junction modelling resulted in certain conditions for the algorithm to meet. A couple of possible new solving algorithms have been proposed. The only topic that has not extensively been considered is the efficiency of these algorithms. That remains open for further research, which will be discussed further is Section 7.1.

Overall, the significance of the findings is mainly on the practical results in realistic networks. Some topics were known theoretically. Only the statements were limited to ‘... if these conditions are not meet, we cannot guarantee ...’. This study has taken a step further, it shows that in models with realistic and extensive junction modelling, these conditions are indeed not met, and the effects are easily visible in realistic networks. Furthermore, this study provides a descent argumentation for Omnitrans International to help them choose a new algorithm for their static assignment.

# 7 DISCUSSION

In this section, the main results and conclusions are discussed. For example, we will discuss the assumptions we have made, and the generalization of the results to other models and situations is discussed.

We studied traffic modelling, and therefore we have to see the results and conclusions in the perspective of modelling. With a model one attempts to approximate reality, so it may be tempting to interpret the model as a realistic situation. But at first, the model gives an average picture, and does not take incidents into account. Secondly, a model is always based on assumptions, and that may lead to errors. For example, it is well known that errors are made in the estimation of the origin-destination matrix. Despite of these possible errors, the model still gives a good general picture of the traffic flows in a network, which can support traffic policy makers in making decisions.

A static model is also based on some assumptions. In a static model, the traffic flow on a road can exceed the capacity of the road, which naturally is not realistic. Due to this fact some specific loads and travel times may be unrealistic. When the flow exceeds the capacity the BPR-function provides very large travel times, to force travellers to avoid using that route in the next iteration.

Furthermore, for inspecting the turn cost function with respect to some characteristics, we assumed a simplified version of the turn cost function. For example, the configuration of turning movements on entry lanes can vary, which results in different capacities. We assumed every turning movement corresponds to exactly one lane. Also, we omitted ‘apparent’ conflicts, some parameters, and the geometric delay. Still, the ‘structure’ of the turn cost function is the same, so the characteristics of this function can be inspected accurately.

Also, on signalized junctions we assumed the traffic light settings are fixed, including green times and the cycle time. In practice, as well as in the traffic models in OmniTRANS, most of the signalized junctions have variable traffic light settings, anticipating on the traffic flow on the junction. In this situation the turn cost function also becomes non-separable and is not necessarily diagonal dominant and symmetric. Since this is the same situation as for unsignalized junctions, we state that the same conclusions hold.

For improving the Frank-Wolfe algorithm, one of the suggestions we have made was an improvement of the approximation of the ´objective function´. In the networks of Delft and Leuven this led indeed to a better solution. However, it is not guaranteed that this will always be an improvement, in all networks and with all amounts of congestion. Still, in two representative networks it was an improvement, so it is plausible to state this a better way of approximating the ´objective value´ in similar networks. Note that in non-urban models the effect is small, because junction delays contribute less to the total travel time.

Multiple equilibria, as obtained in the networks of Delft and Leuven, are likely to exist in other networks as well. That is because we have obtained these equilibria easily, most of the initializations we tried resulted in a different equilibrium. Actually, it is not the most important to state that there always will exist multiple equilibria, but that there can (and often will) exist multiple equilibria. That fact is enough to question the uniqueness of a solution.

On the other hand, we have to see the 'hurt' of the multiple equilibria in perspective of the practical use. In practice, an assignment algorithm is often performed with only ten iterations, no matter the type of network or the accuracy of the solution. Then, equilibrium is often not reached, so the importance of accuracy may be overrated.

It is hard to split the effects of the existence of multiple equilibria on the one hand, and the non-existing objective function on the other hand. Those effects are not mutually exclusive, they are related. Most of the algorithms are based on the existence of a convex solution space and a unique solution, so new algorithms (for the asymmetric problem) are not necessarily useful in a non-convex solution space. Also, the suggestions for dealing with both problems coincide. The diagonalization algorithm that can deal with asymmetric costs is also a possible strategy to deal with the existence of multiple equilibria.

## 7.1 Recommendations for Further Research

This study covers the possibilities for algorithms to perform an accurate assignment in models with junction delays. No extensive results are given about the efficiency of these algorithms. In further research, for example several algorithms can be implemented and numerical results can be compared.

Instead of static models, recently dynamic models are more and more used. In dynamic models the definition and existence of equilibria is not as extensively studied as in static models. It might be interesting to do further research to the effect of junction modeling on the TAP and equilibria in dynamic models.

The setting of traffic lights can be controlled. For example, a government can decide to decrease the green time on a specific turn, to make travelling across that turn less attractive. Those decisions influence the traffic flows, and therefore influence the TAP. Among others, Yang and Yagar (1995), Gartner and Al-Malik (1996) and Chen and Hu (2009) studied this topic. Further research to implications of this topic to traffic models might be interesting.

## 7.2 Recommendations for Omnitrans International

Although the last decade a number of new solution methods for the TAP were presented, these new methods were not necessarily able to cope with traffic models with junction delays. For the choice of a new algorithm for the static assignment in OmniTRANS that can deal with junction delays, one has to keep in mind that the objective function of the Beckmann formulation is not of use any more, unless in a 'diagonalized' problem as in the Diagonalization Algorithm. Also the gradient of the objective function and its minimum are not of use, since the objective function simply does not exist. Instead of the Beckmann formulation the Variatonal Inequality can be used. The Variational Inequality holds in equilibrium, so it can be used as a stopping criterion. Also the cost functions and its derivatives can be used in the iterations. The idea is that load is shifted from 'expensive' to 'cheaper' routes.

Two specific algorithms are discussed in Chapter 5, which are recommended to use for the static assignment in OmniTRANS. The first is the Diagonalization Algorithm, which is the easiest to implement. It is a variation on the Frank-Wolfe algorithm as already implemented in OmniTRANS. The elements of the algorithm are already there, only some adaptations are needed. They are explained in Appendix III. The second suggestion for a new algorithm is Simplicial Decomposition. It is a path-based algorithm, which can deal with asymmetric costs, and so it is applicable to models with junction delays. The implementation of Simplicial Decomposition in OmniTRANS is far more radical that the Diagonalization Algorithm. Numerical results show Simplicial Decomposition is fast in terms of calculation time, although it uses a large amount of memory.

Furthermore, it is very important to realize that realistic junction modelling directly implies the existence of multiple solutions. One has to be aware of this fact when interpreting solutions, especially when solutions of different scenarios are to be compared. When one wants to be sure an honest comparison of solutions can be made, a unique minimum is required. Only in that case it is guaranteed that the differences in the solutions are due to the differences in the scenarios, and not to a ´coincidence´ which one of multiple possible solutions is found. But when one requires a unique minimum, one has to release the realistic junction modelling. This remains a trade-off, and it is important to inform and help the users of OmniTRANS in making a good consideration and decision with respect to this topic.

Bedankt voor het lezen van mijn verslag! De originele conclusie en discussie moet goed te lezen zijn voor niet-wiskundigen, dus daarop volgt geen aanvullende uitleg. Wat ik nog wel wil benadrukken is dat ik hoop dat de niet-wiskundige lezer heeft gezien dat de wiskunde aan de basis staat van hele begrijpelijke en praktische processen. De wiskunde zelf is misschien niet altijd toegankelijk, daarentegen zou de praktische betekenis ervan dat wel altijd moeten zijn!

## REFERENCES


Bar-Gera, H. (2002). Origin-based Algorithm for the Transportation Assignment Problem. Transportation Science, 36, 398-417.

Bar-Gera, H. (2010). Traffic Assignment by Paired alternative segments. Transportation Research Part B, 44(8-9), 1022-1046.

Beckmann, M., McGuire, C. B. & Winsten, C. B. (1956). Studies in the Economics of Transportation. Yale University Press, New Haven, CT.

Bertsekas, D. P. (1976). On the Goldstein – Levitin – Polyak Gradient Projection Method. IEEE Transactions on Automatic Control, 21(2), 174-184.

Brandt, F. & Schilpzand, M. (2007). Explanation of Junction Modelling, the code in OmniTRANS. Version 1.0. Omnitrans International.

Chen, L. & Hu, T. (2009). Dynamic Equilibrium for Combined Signal Setting and Time-dependent Traffic Assignment. Proceedings of the Eastern Asia Society for Transportation Studies, Volume 6.

Dafermos, S. (1971). An Extended Traffic Assignment Model with Applications to Two-way Traffic. Transportation Science, 5(4), 366-389.

Dafermos, S. (1980). Traffic Equilibrium and Variational Inequalities. Transportation Science, 14, 42-54.

Dial, R. B. (2006). A Path-based User-equilibrium Traffic Assignment Algorithm that Obviates Path Storage and Enumeration. Transportation Research Part B, 40, 917-936.

Dupuis, C. & Darveau, J. (1986). The Convergence Conditions of Diagonalization and Projection Methods for Fixed Demand Asymmetric Network Equilibrium Problems. Operations Research Letters, 5(3), 149-155.

Fisk, C. & Nguyen, S. (1982). Solution Algorithms for Network Equilibrium Models with Asymmetric User Costs. Transportation Science, 16(3), 361-381.

Florian, M. & Hearn, D. (1995). Network Equilibrium Models and Algorithms. In M.O. Ball et al. (eds.), Handbooks in Operations Research & Management Science, volume 8 (pp. 485-550).

Florian, M. & Spiess, H. (1982). The Convergence of Diagonalization Algorithms for Asymmetric Network Equilibrium Problems. Transportation Research Part B, 16B(6), 477-483.

Florian, M. A., Constantin, I. & Florian, D. (2009). A New Look at Projected Gradient Method for Equilibrium Assignment. Transportation Research Record: Journal of the Transportation Research Board, 2090, 10-16.

Frank, M. & Wolfe, P. (1956). An Algorithm for Quadratic Programming. Naval Research Logistics Quarterly, 3(1-2), 95-110.

Freund, D. & Saxena, U. (1984). An Algorithm for a Class of Discrete Maximum Entropy Problems. Operations Research, 32(1), 210-215.

Gentile, G. (2009). Linear User Cost Equilibrium: a New Algorithm for Traffic Assignment. Roma, Italy.

Gartner, N. H. & Al-Malik, M. (1996). Combined Model for Signal Control and Route Choice in Urban Traffic Networks. Transportation Research Record, 1554, 27-35.

Hearn, D. W., Lawphongpanich, S. & Ventura, J. A. (1985). Finiteness in Restricted Simplicial Decomposition. Operations Research Letters, 4(3), 125-130.

Jayakrishnan, R., Trai, W. K., Prashker, J. N. & Rajadhyaksha, S. (1994). A Faster Path-based Algorithm for Traffic Assignment. In Proceedings of the Transportation Research Board Annual Meeting 1994, Washington, DC.

Larsson, T., Lundgren, J. T., Rydergren, C. & Patriksson, M. (2001). Most Likely Traffic Equilibrium Route Flows Analysis and Computation. In F. Giannessi et al (eds.), Equilibrium Problems: Non-smooth Optimization and Variational Inequality Models (pp. 129-159). Kluwer Academic Publishers, the Netherlands.

Larsson, T., Patriksson, M. & Rydergren, C. (1997). Applications of Simplicial Decomposition with Nonlinear Column Generation to Nonlinear Network Flows. In P. M. Pardalos, D. W. Hearn, & W. W. Hager (eds.), Network Optimization (pp. 346-373). Springer, Berlin.

Lawphongpanich, S. & Hearn, D. W. (1984). Simplicial Decomposition of the Asymmetric Traffic Assignment problem. Transportation Research Part B, 18(2), 123-133.

Nguyen, S. & Dupuis, C. (1984). An Efficient Method for Computing Traffic Equilibria in Networks with Asymmetric Transportation Costs. Transportation Science, 18(2), 185-202.

Nie, Y. (2010). A Class of Bush-based Algorithms for the Traffic Assignment Problem. Transportation Research Part B, 44, 73-89.

Sheffi, Y. (1985). Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods. Englewood Cliffs, NJ: Prentice-Hall.

Slavin, H., Brandon, J. & Rabinowicz, A. (2006). An Empirical Comparison of Alternative User Equilibrium Traffic Assignment Methods. In Proceedings of the European Transport Conference 2006.

Smith, M. J. (1983a). The Existence and Calculation of Traffic Equilibria. Transportation Research Part B, 17B(4), 291-303.

Smith, M. J. (1983b). An Algorithm for Solving Asymmetric Equilibrium Problems with a Continuous Cost-flow Function. Transportation Research Part B, 17B(5), 365-371.

Ton, S. W. (2011). Static User Equilibrium, a Search for an Optimal Algorithm. Master Thesis, Omnitrans International, University of Twente.

Transportation Research Board (2000). Highway Capacity Manual. National Research Council, Washington D.C.

Wardrop, J. (1952). Some theoretical aspects of road traffic research. Proceedings of the Institution of Civil Engineers, Part II, 1, 325-378.

Yang, H. & Yagar, S. (1995). Traffic Assingment and Signal Control in Saturated Road Networks. Transportation Research Part A, 29A(2), 125-139.

Zhou, Z. & Martimo, M. (2009). Computational Study of Alternative Methods for Static Traffic Equilibrium Assignment. Citilabs Inc, Alameda, California.

## APPENDIX I: NOTATION

| | |
|---|---|
| $a, b$ | link |
| $A$ | set of links |
| $c_a$ | cost function of link $a$ |
| $c$ | vector of cost functions, $c = (c_a, \forall a \in A)$ |
| $c_{\max}$ | maximal delay on link or turn |
| $c_k$ | cost function of path $k$ |
| $c_k^{rs}$ | cost function of path $k$ connecting $r$ and $s$ |
| $\tilde{c}_a^n$ | 'diagonalized' cost function of link $a$, loads fixed with respect to iteration $n$ |
| $d_{rs}$ | demand from $r$ to $s$ |
| $D$ | vector of the demand, $D = \{d_{rs}, \forall rs \in OD\}$ |
| $DG$ | duality gap |
| $\Delta$ | link-path incidence matrix, where an element $\delta_{a,k}^{rs} = \begin{cases} 1, \text{ if } a \text{ is on } k \text{ connecting } r \text{ and } s \\ 0, \text{ otherwise} \end{cases}$ |
| $e$ | node |
| $E$ | set of nodes |
| $f_k$ | load on path $k$ |
| $f_k^{rs}$ | load on path $k$ connecting $r$ and $s$ |
| $f$ | vector of all path loads, $f = \{f_k, \forall k \in K\}$ |
| $g$ | lane group |
| $G_p$ | set of lanes that get green in phase $p$ |
| $h_l$ | number of turns on lane $l$ |
| $\theta_l$ | fraction of green time of lane $l$ |
| $i, j$ | solutions of assignment algorithm |
| $J$ | Jacobian of the cost function, $J = \left[\frac{\partial c_i}{\partial x_j}\right]$ |
| $k^{rs}$ | path from $r$ to $s$ |
| $K^{rs}$ | set of paths from $r$ to $s$ |
| $K$ | set of all paths |
| $L_a$ | length of link $a$ |
| $\lambda$ | stepsize |
| $\lambda_n$ | stepsize in iteration $n$ |
| $l$ | lane |
| $\Lambda$ | path-OD matrix , where an element $\Lambda_{rs,k} = \begin{cases} 1, \text{ if path } k \in K_{rs}; \\ 0, \text{ otherwise;} \end{cases}$ |
| $L$ | Lagrangian |
| $\mu$ | increasing parameter, $\mu = \frac{n}{N}$ |
| $n, m$ | iteration number |

$N$ total number of iterations

$OD$-pair a pair of one origin and one destination

$OD$ set of all $OD$-pairs

$p$ phase

$\pi^{rs}$ optimal travel time from $r$ to $s$

$q$ capacity

$q'$ base capacity

$q_{\min}$ minimal capacity

$r$ origin

$R$ set of origins

$RSE$ Relative Squared Error, $RSE = \frac{\left(x_a^j - x_a^i\right)^2}{x_a^i}$

$S$ set of destinations

$s$ destination

$\sigma$ saturation flow

$t$ turn

$\tau$ signal cycle time

$u^1, u^2, u^3$ Lagrangian multipliers

$v_a^{\max}$ maximum speed on link $a$

$w_a^n$ new load on link $a$ obtained in $n^{\text{th}}$ iteration

$x_a$ load on link $a$

$x_a^i$ load on link $a$ in solution $i$

$x$ vector of all link loads, $x = \{x_a, \forall a \in A\}$;

$Y_t$ set of conflicting turns with turn $t$

$z$ objective function of Beckmann formulation

$\tilde{z}_1$ current 'approximation of objective function'

$\tilde{z}_2$ improved 'approximation of objective function'

$\tilde{z}_3$ 'approximation of objective function' with increasing turn costs

## Appendix II: Differentials of Cost Function

The cost function is differentiated with respect to the load on its own turn in equation (A.1) and with respect to the load on a (arbitrary) conflicting turn in equation (A.2).

$$\frac{\partial c_a}{\partial x_a} = \frac{900}{0.8\left(\sigma_a - 0{,}99\sum_{b\in Y_a} x_b\right)} + \frac{450\left(\frac{4}{\left(0.8\left(\sigma_a - 0{,}99\sum_{b\in Y_a} x_b\right)\right)^2} + \frac{2\left(\frac{x_a}{0.8\left(\sigma_a - 0{,}99\sum_{b\in Y_a} x_b\right)} - 1\right)}{0.8\left(\sigma_a - 0{,}99\sum_{b\in Y_a} x_b\right)}\right)}{\sqrt{\left(\frac{x_a}{0.8\left(\sigma_a - 0{,}99\sum_{b\in Y_a} x_b\right)} - 1\right)^2 + \frac{4\left(\frac{x_a}{0.8\left(\sigma_a - 0{,}99\sum_{b\in Y_a} x_b\right)} - \frac{1}{2}\right)}{0.8\left(\sigma_a - 0{,}99\sum_{b\in Y_a} x_b\right)}}} \tag{A.1}$$

$$\frac{\partial c_a}{\partial x_b} = \frac{3564x_a}{5\left(0.8(\sigma_a - 0{,}99x_b)\right)^2} + \frac{14256}{5\left(0.8(\sigma_a - 0{,}99x_b)\right)^2} + \frac{450\left(\frac{396x_a}{125\left(0.8(\sigma_a - 0{,}99x_b)\right)^3} + \frac{396\left(\frac{x_a}{0.8(\sigma_a - 0{,}99x_b)} - \frac{1}{2}\right)}{125\left(0.8(\sigma_a - 0{,}99x_b)\right)^2} + \frac{198x\left(\frac{x_a}{0.8(\sigma_a - 0{,}99x_b)} - 1\right)}{125\left(0.8(\sigma_a - 0{,}99x_b)\right)^2}\right)}{\sqrt{\left(\frac{x_a}{0.8(\sigma_a - 0{,}99x_b)} - 1\right)^2 + \frac{4\left(\frac{x_a}{0.8(\sigma_a - 0{,}99x_b)} - \frac{1}{2}\right)}{0.8(\sigma_a - 0{,}99x_b)}}} \tag{A.2}$$

# APPENDIX III: IMPLEMENTATIONS IN OMNITRANS

In this appendix some implementations in OmniTRANS are explained. First, the implementation of the improved approximation of the 'objective function' is explained, as discussed in Section 4.1.1. Second, the implementation of the Diagonalization Algorithm, as discussed in Section 5.1, is explained. Finally, a suggestion for the implementation of the fixed values for junction delays is given, as discussed in Section 4.2.2.

### NEW APPROXIMATION OF 'OBJECTIVE FUNCTION'

In Section 4.1.1, we have proposed a new approximation of the 'objective function', which actually does not exist. The current calculation of the 'objective function' is

$$\tilde{z}_1(x) = \sum_{\text{'normal' links } a} \int_0^{x_a} c_a(\omega)d\omega + \sum_{\text{turns } a} c_a(x). \qquad \text{(A.3)}$$

The suggested new calculation is

$$\tilde{z}_2(x) = \sum_{\text{'normal' links } a} \int_0^{x_a} c_a(\omega)d\omega + \sum_{\text{turns } a} c_a(x) \cdot x_a. \qquad \text{(A.4)}$$

where

$x_a$ is the load on link $a$,
$x$ is a vector of all loads, $x = (x_a, a \in A)$;
$c_a$ is the cost of link $a$.

This calculation is used during the line search in the Frank Wolfe algorithm, called the 'User Equilibrium' assignment in OmniTRANS. For this new calculation of the 'objective function' an adaptation is needed in the module *Traffic*, in the file *TOtTrafficMinimiseAlphasEquilibrium.h*.

In this file line 134 and line 135 correspond to the first and last black line below, the blue lines in between are to be added.

```
totalTravelCost = linkCostList.sum();

TurnLinkList& turnlist = assignment->propNetwork->getTurnLinkList();
for  (TurnLinkList::iterator  iter=turnlist.begin();  iter!=turnlist.end();
     ++iter)
{
IDType turnId = (*iter)->getID();
linkCostList[turnId] = linkAVGLoadList[turnId].sum()*linkCostList[turnId];
}
assignment->listTurns(1,assignment->propNetwork-
     >getTurnLinkList(),linkAVGLoadList,linkCostList);
assignment->listLinks(0,linkAVGLoadList,linkCostList);

return totalTravelCost;
```

### DIAGONALIZATION

In Section 5.1, the Diagonalization Algorithm (DA) is proposed as a solution algorithm for the Traffic Assignment Problem with junction delays. The steps of DA have been explained, but the implementation in OmniTRANS is discussed in this section.

In both the modules *Traffic* and *JunctionModelling* adaptations are needed. Before explaining the adaptations, let us declare some notation for different loads during the process. Let $x^{n-1}$ be the vector with loads resulting from the last iteration. Let $w^n$ be the vector with loads resulting from the All-Or-Nothing assignment in the beginning of iteration $n$, this is the 'search direction'. Let $x^n$ be the vector with the final loads after iteration $n$, which is actually a linear combination of the 'old' and the 'new' load, namely $x^n = (1-\lambda)x^{n-1} + \lambda\, w^n$. The step size $\lambda$ is chosen according to minimize the objective function: $\min_\lambda z((1-\lambda)x^{n-1} + \lambda\, w^n)$. To minimize this objective function, different 'temporary' loads are used, which we refer to as $\tilde{x}^n$. These temporary loads are 'attempts' to find the minimum. This occurs in the line search process.

The adaptations that are needed are the following. In *Traffic*, before the line search starts, the 'old' load $x^{n-1}$ needs to be stored. During the line search, for different temporary loads $\tilde{x}^n$ the costs are calculated. This is done by the *minimizeByBrent* routine. The turn costs are also calculated, for which *JunctionModelling* is called. *Traffic* needs to give both the old load $x^{n-1}$ and the temporary new load $\tilde{x}^n$ to *JunctionModelling*. *JunctionModelling* needs to use these both sets of loads for the calculation of the turn delay. The old load $x^{n-1}$ is to be used for the load on conflicting turns, which is only used in the calculation of the capacity. The temporary new load $\tilde{x}^n$ is to be used for the load on the turn itself.

When this is implemented, the ´UE´ assignment in OmniTRANS becomes the ´streamlined version´ of the Diagonalization Algorithm, as described by Sheffi (1985).

#### FIXED JUNCTION DELAYS

One of the solutions for guaranteeing a unique equilibrium is to set fixed values for the junction delays, as discussed in Section 4.2.2. For this study, a temporary implementation is used for setting fixed junction delays, which is a series of Ruby jobs, as described below.

In this job, the costs are calculated based on the loads of 'representative' assignment.

```
replacecostassignment = OtTraffic.new

#recalculate the costs based on loads from this PMTURI:
replacecostassignment.initialLoad = [1,10,10,1,50,1]
#write this costs in PMTURI:
replacecostassignment.load = [1,10,10,1,25,1]

replacecostassignment.bprPerType = BprPerType
replacecostassignment.junctions = true
replacecostassignment.assignMethod = REPLACECOST

replacecostassignment.execute
```

Then, these turn costs transferred to impedances on the turns.

```
turndelay = []
turnnr = []

#get turndelays from assignment
turn5tabel = OtTable.new($Ot.mainVariantDirectory + "turn5data1.DB")
k=0
turn5tabel.open
turn5tabel.filter = "result = 25"
turn5tabel.filtered = true
recordCount = turn5tabel.recordCount
```

```
while !turn5tabel.eof?
  k=k+1
  turndelay[k] = turn5tabel.get[14]
  turnnr[k] = turn5tabel.get[0]
  turn5tabel.next
end
writeln 'Processed ',recordCount, ' records'
turn5tabel.close

#write turndelays as impedance
turn3tabel = OtTable.new($Ot.mainVariantDirectory + "turn3data1.DB")
turn3tabel.open
for i in 1..recordCount
  if !turn3tabel.locate(['turnnr','mode','time'],[turnnr[i],10,10])
  turn3tabel.append
  turn3tabel.set([turnnr[i],10,10,turndelay[i]])
  turn3tabel.post
  end
end
recordCount = turn3tabel.recordCount
writeln 'Processed ',recordCount, ' records'
turn3tabel.close
```

Now, an assignment can be performed. This assignment is based on the fixed values for the junction delays.

```
fixeddelayassignment = OtTraffic.new

fixeddelayassignment.assignMethod      = USEREQUILIBRIUM
fixeddelayassignment.load              = [1,10,10,1,24,1]

fixeddelayassignment.bprPerType        = BprPerType
fixeddelayassignment.iterations        = 100
fixeddelayassignment.epsilon           = 0.00001
fixeddelayassignment.junctions         = false
#is doesn't matter if junction modelling is on or off, since the
#impedance overrules the calculation of the junction delay

fixeddelayassignment.execute
```

Finally, the impedances are deleted, so that the situation is normal for next assignments.

```
myquery = OtQuery.new
myquery.sql = "delete from 'turn3data1.DB' where impedance > -1"
myquery.execute
```

This is a way to perform an assignment with fixed values for junction delays.