ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Department of Mathematics
Semester Project

Game Theory and Traffic Control

Ralph Villiger

Supervision
Michel Bierlaire

9th of February 2000

Contents

Introduction

This work treats a promising application of game theory to the traffic control problem. Traffic control deals with the optimal signal settings in a urban transportation network. The behaviour of the network users, the car drivers, can be seen as a game between all drivers. Finally the interactions between the drivers and the traffic authority, the organ that determines the signal settings, can be interpreted as a game as well. This perspective enables an interesting approach to the traffic control problem and opens new ways for its solution.

First a short introduction to game theory is given. After having learned the basics about traffic theory the reader gets to know a game theoretic interpretation of different traffic problems. Finally a solution algorithm based on the game theoretic analysis is explained.

This work is considered as an introduction to the interaction between game theory and traffic science. A last chapter lines out a possible development of the presented theory.

Lausanne, February 2000

Ralph Villiger

Chapter 1

Game Theory

A game is a conflict situation where the involved persons get a result. The involved persons are called players. They have normally different goals. In most games they are aiming at maximizing their own profit. In general we know two types of games: random games and strategic games. In random games the players cannot influence the result. A typical example are dice games. Random games are discussed in probability theory. In strategic games on the other hand the players determine at least partly the outcome of the game by choosing their way of behaviour, i.e. their strategy. Chess is a well known strategic game. Game theory treats the strategic games.

1.1 Games in Strategic Form

There are different ways to describe a game. We will use the strategic form. A game is typically characterized by its *players*, their *strategies* and their *payoffs*.

The Players

The players can cooperate, build partial alliances or not cooperate at all. They are denoted $i \in \mathrm{I}$, with $|\mathrm{I}|=n$.

The Strategies

Each player i disposes of a certain set of strategies $\mathcal{S}_i$. The strategies s_i are the different options the player can choose. With respect to the conditions of the game, only certain combinations of strategies may be allowed, they are called *profiles*. The set of all profiles is $\mathrm{L} \subseteq \mathrm{S} = \mathcal{S}_1 \times \cdots \times \mathcal{S}_n$. Profiles are denoted $s \in \mathrm{L}$. To describe the strategies of player i's opponents we use the notation s_{-i}. In general the notation $-i$ concernes player i's opponents.

We make a difference between *pure strategies* and *mixed strategies*. A pure strategy is just one way to behave for a player, as described above. Pure strategies for player i are denoted as s_i. A mixed strategy is a set of strategies combined with a probability distribution over this set. Mixed strategies for player i are denoted as $\sigma_i = \sum_j P_j s_i^j$, where s_i^j are different pure strategies for player i and P_j are the probabilities, that the player chooses strategy s_i^j.

The Payoffs

The payoff is the result each player gets at the end of the game. The payoff function is first personal for each player and second dependent on the game profile. This means that a player's payoff is not only influenced by his own strategy but also by the other players' strategies. We write for player i's payoff function: $u_i(s)$ for $s \in \mathrm{L}$. Each player aims at maximizing his own payoff. Note that each player is supposed to decide rationally and that this is taken in account in the decision making process.

1.2 Equilibria

Game theorists are specially interested in the results of a game and less in the way the players come to this result. It may be interesting to know if there are some strategies the users prefer to others. An important aspect of game theory is the notion of *equilibrium*. We make a difference between pure and mixed equilibria. A *pure equilibrium* is a game profile that remains constant, i.e. the players do not want to change. We speak about a *mixed equilibrium* if there exists a probability distribution over pure strategies, which the players switch between. This means that the profile is composed of mixed strategies. We denote a mixed profile $\sigma \in \Sigma = \Omega \times \mathrm{L}$, where Ω is the probability space for the strategies. To simplify we assume from now on that Σ is the space of pure and mixed feasible profiles and we write $s \in \Sigma$ and $\sigma \in \Sigma$. In fact every pure strategy is a mixed strategy: $s_i^k = \sum_j \delta_{j,k} s_i^j$, where $\delta_{j,k}$, the Kronecker symbol, is the probability.

1.2.1 Strict Dominance

In some simple games the equilibrium is easy to find by iterated strict dominance. For the better understanding of the above given notions and to illustrate strict dominance we play through the so-called prisonners dilemma.

Two suspect persons are arrested by the police and put into different cells to avoid communication between the two. A prisonner gets a reward if he betrays the other and he is set free if he is not betrayed by the other. Identificating this dilemma with a game we get:

players	:	the two prisonners
strategies	:	to betray or not to betray
payoffs	:	$\begin{cases} +1 & \text{if he betrays and gets the reward} \\ +1 & \text{if he is set free} \\ -1 & \text{if he is betrayed} \end{cases}$

The payoff function can be represented in matrix form, where the lines are the strategies for player one and the colons the strategies for player two. The first number stands for player one's payoff, and the second for player two's payoff. So, if for example player 1 chooses to betray and player 2 does not, player 1's payoff is 2, he is set free and gets the reward, and player 2's payoff is -1, he is betrayed.

	to betray	not to betray
to betray	0,0	2,-1
not to betray	-1,2	1,1

We see that independent of the other player's choice betraying gives a higher payoff to the player. The strategy *to betray* strictly dominates the strategy *not to betray*. A player will never choose a dominated strategy, because whatever the other player decides to do, he coud win more with the dominating strategy. In our problem the situation is the same for the two players, both choose to betray the other. The profile *(to betray, to betray)* is a pure equilibrium and the payoff is for both zero. In the prisonners dilemma the equilibrium has been found by strict dominance.

Strict dominance can be expressed as follows:

A strategy $\hat{s}_i$ is strictly dominated for player i if

$$\exists \sigma_i \in \Sigma_i \text{ such that } u_i(\sigma_i, \sigma_{-i}) > u_i(\hat{s}_i, \sigma_{-i}) \quad \forall \sigma_{-i} \in \Sigma_{-i} \tag{1.1}$$

To illustrate the notion of mixed equilibrium we have a look at the next game. In this game the players, a man and a woman, can decide between two rooms, room 1 and room 2. Player 1, the man, is in love with player 2, the woman, and is therefore happy, if they choose the same room, and unhappy if they choose different rooms. The woman does not like the man and prefers not to be in the same room. The payoff matrix for this game is:

	room 1	room 2
room 1	1,-1	-1,1
room 2	-1,1	1,-1

There cannot be a pure equilibrium. If they choose the same room, the woman is going to change, if they choose different rooms, the man is going to change. No strategy dominates another, both strategies are of the same value for both players. If the woman's room choice is not known to the man, he has the same hope to be in the same room whatever he decides. He chooses each strategy with probability $\frac{1}{2}$. For the woman the situation is equivalent. This game has a mixed equilibrium with equilibrium profile $(\frac{1}{2}(\textit{room 1})+\frac{1}{2}(\textit{room 2}),\frac{1}{2}(\textit{room 1})+\frac{1}{2}(\textit{room 2}))$. Each player decides which room he takes by flipping a coin.

1.2.2 Nash Equilibrium

A game is in a *Nash equilibrium*, if no player can improve his payoff by choosing another strategy, given that the other players do not change their strategy. If a player sees a possibility to improve his actual payoff by changing his strategy, he logically does it and the system is therefore not in an equilibrium state. So in the Nash equilibrium each player's strategy is the best response to the other players' strategies. This gives:

The profile σ^* is a Nash equilibrium if

$$\sigma_i^* = \arg\max_{\sigma_i} u_i(\sigma_i, \sigma_{-i}^*) \quad \forall i \tag{1.2}$$

A game has not necessarily only one equilibrium, as we see in the following example. This is the same game like above, but this time the woman is in love with the man, too.

	room 1	room 2
room 1	1,1	-1,-1
room 2	-1,-1	1,1

Here the profiles (*room 1*,*room 1*) and (*room 2*,*room 2*) are both Nash equilibria. At which equilibrium we arrive depends on the initial state.

1.3 Different Game Types

Now we are going to examine different game types. They will be discussed with the following example.

There are two firms that produce the same item. Their strategies are the decision, how many of them they would like to produce. We denote the output level q_i. There are three different options, 3, 4 or 6. The price of an item is determined by $p = 12 - (q_1 + q_2)$. We assume costfree production. Their payoffs are $u_i = pq_i = (12 - (q_i + q_{-i}))q_i$. In profile $(3, 4)$ for example, the price is $p = 12 - (3 + 4) = 5$. Firm one has therefore a payoff $u_1 = 3 \cdot 5 = 15$. The second firm gets $u_2 = 4 \cdot 5 = 20$.

	3	4	6
3	18,18	15,20	9,18
4	20,15	16,16	8,12
6	18,9	12,8	0,0

1.3.1 One-Stage Cournot Game

In the Cournot game the players take their decision at the same moment. This means, that they do not know the other players' strategy. In fact their decision is based on the other players' behaviour of the preceeding level. The game is at an equilibrium, if the following is fullfilled:

$$\sigma_i^* = \arg\max_{\sigma_i} u_i(\sigma_i, \sigma_{-i}^*) \quad \forall i \tag{1.3}$$

1.3 is the same equation like 1.2. The games seen until now were all Cournot games. The equilibrium of our example game is the profile $(4, 4)$. This is easy to see: If the opponent chooses 4, the player's payoff can be 15, 16 or 12. So the best response is 4.

	4
3	15,20
4	**16**,16
6	12,8

This means that the profile $(4, 4)$ is an equilibrium. On the other hand it is the only one, because if the opponent chooses 3, the player's payoff can be 18, 20 or 18. So the player will switch again to 4 and the opponent will change his production up to 4, too, as seen before.

	3
3	18,18
4	**20**,15
6	18,9

Finally, if the opponent takes 6 as output level, the player's reaction will be production of 3 items, which will be answered with 4 by the opponent and so on.

	6
3	**9**,18
4	8,12
6	0,0

We have:
$(3,3) \Longrightarrow (4,4)$
$(3,4) \Longrightarrow (4,4)$
$(3,6) \Longrightarrow (3,4) \Longrightarrow (4,4)$
$(4,4) \Longrightarrow (4,4)$
$(4,6) \Longrightarrow (4,4)$
$(6,6) \Longrightarrow (3,3) \Longrightarrow (4,4)$

So whereever we start, the equilibrium is $(4,4)$.

1.3.2 Leader-Follower Stackelberg Game

In the Stackelberg game the players are not equal anymore, but there is a leader and a follower. The leader determines his strategy and after that the follower adapts his behaviour to the situation given by the leader's choice. The leader is able to anticipate the followers reaction and determines his strategy with respect to this knowledge. The equilibrium profile can be described like that:

$$\begin{array}{rrcl} & \sigma^*_{leader} & = & \arg\max_{\sigma_{leader}} u_{leader}(\sigma_{leader}, \sigma^*_{follower}) \\ \text{subject to} & \sigma^*_{follower} & = & \arg\max_{\sigma_{follower}} u_{follower}(\sigma_{leader}, \sigma_{follower}) \end{array} \tag{1.4}$$

In our example, if player 1 is the leader and player 2 the follower, we get the following choices for the leader, given that the follower will optimize his own payoff with respect to the leader's strategy:

	follower's strategy	payoff
3	4	15,20
4	4	16,16
6	3	**18**,9

The leader chooses therefore output level 6, because this way he gets the highest payoff. The Stackelberg equilibrium is therefore the profile $(6,3)$. One could argue why in the Cournot game player 1 switches from the profile $(6,3)$ to $(4,3)$? He should know that this will end up in the equilibrium $(4,4)$, where he only receives 16 instead of 18 in the initial situation. There are several reasons. First, player 1 does not know that he is a leader (in the Cournot game he is not a leader in deed), so he does not behave as if he were one. Second, the Cournot game works without anticipation in contrast to the Stackelberg game. Third, player 2 could force player one after a while to change his strategy by producing four items. This would mean that player 2 would take the leadership.

The leader's payoff in the Stackelberg game is at least as good as his payoff in the Cournot game. This does not hold for the follower.

1.3.3 Monopoly Game

In the monopoly game there is only one player and he aims to maximize his payoff. Mathematically:

$$\sigma^* = \arg\max_{\sigma} u(\sigma) \tag{1.5}$$

In our example we can assume a cartel of the two firms, so they want to maximize the common payoff $u = u_1 + u_2$.

	3	4	6
3	**36**	35	27
4	35	32	20
6	27	20	0

The equilibrium profile is therefore $(3, 3)$

Let us have a quick look at the prices in each game.

Cournot	:	4
Stackelberg	:	3
Monopoly	:	6

We see that the buyers suffer under a monopoly, but surprisingly perfect concurrence like in the Cournot game is not to their best neither.

1.3.4 Bayesian Games with Uncertainty

Until now we assumed, that the players have perfect information about the other players' strategies and payoffs. In practice this is not always the case. It is likely that the players suffer from a lack of information that can influence their strategies and thus their payoffs. Each player has some private information about his own conditions that are not common knowledge. The private information proper to player i is called player i's *type* and is denoted θ_i. We can also say that θ_i is only observed by player i. We assume that the types θ_i belong to some space Θ_i. Furthermore each player assigns a probability $P(\theta_{-i}|\theta_i)$ to the other players' types given his own type. The payoff function depends now not only on the players' strategies but also on their different types ($u_i = u_i(s, \theta)$).

Games that deal with uncertainty are called *Bayesian Games*. A game with profile s^* and players' type θ^* is in its so-called *Bayesian Equilibrium* if the following holds:

$$s_i^* = \arg\max_{s_i} \sum_{\theta_{-i}} P(\theta_{-i}|\theta_i^*) u_i(s_i, s_{-i}^*, (\theta_i^*, \theta_{-i})) \quad \forall i \tag{1.6}$$

In fact, each player maximizes the expected gain, taking the mean over all possible types. The types can also be distributed continuously, in this case we have:

$$s_i^* = \arg\max_{s_i} \int_{\theta_{-i}} p(\theta_{-i}|\theta_i^*) u_i(s_i, s_{-i}^*, (\theta_i^*, \theta_{-i}))\, d\theta_{-i} \quad \forall i, \tag{1.7}$$

where $p(\theta_{-i}|\theta_i^*)$ is the conditional probability density of θ_{-i} given θ_i^*.

To illustrate such a Bayesian game we take up the same example of the two firms like before. This time the payoff function of the first firm $u_1 = q_1(12 - (q_1 + q2))$ is common knowledge. Firm 2 has $u_2 = q_2((12 + \theta) - (q_1 + q2))$ as payoff function. θ is firm 2's private information. Let us say that θ can take $\theta_L = 2$ or $\theta_H = -2$ as values. Firm 1 assigns probability $\frac{1}{2}$ to each possible type. The number $12 + \theta$ can be interpreted as the difference between the basic price and the unit production cost. The type θ_L can thus be seen as low production costs, firm 1 has medium production cost and type θ_H means high production costs. Firm 2's output level holds:

$$q_2^*(\theta) = \arg\max_{q_2} q_2((12 + \theta) - (q_1 + q_2)) \tag{1.8}$$

$$\Rightarrow q_2^*(\theta) = \frac{12 + \theta - q_1}{2} = 6 + \frac{\theta - q_1}{2} \tag{1.9}$$

For firm 1 we have:

$$q_1^*(\theta) = \arg\max_{q_1} \left\{ \frac{1}{2} q_1(12 - (q_1 + q_2(\theta_L))) + \frac{1}{2} q_1(12 - (q_1 + q_2(\theta_H))) \right\} \tag{1.10}$$

$$= \arg\max_{q_1} \left\{ 12 - q_1 - \frac{1}{2}(q_2(\theta_L) + q_2(\theta_H)) \right\} \tag{1.11}$$

$$= \arg\max_{q_1} q_1(6 - \frac{1}{2} q_1) \tag{1.12}$$

$$\Rightarrow q_1^* = 6 \tag{1.13}$$

This gives now for firm 2 the following output levels:

$$q_2^*(\theta_L) = 4 \tag{1.14}$$

$$q_2^*(\theta_H) = 2 \tag{1.15}$$

Now we have for the two types the following equilibria:

		θ_L	θ_H
firm 1	strategy	6	6
	payoff	12	24
firm 2	strategy	4	2
	payoff	16	4

For further information on game theory we refer to [2] and [5].

Chapter 2

The Traffic Problem

2.1 The Problem

Traffic signal control systems are methods to stear the traffic in urban networks. There are two groups involved in the problem. On the one hand there are the users, i.e. the drivers, and on the other hand the traffic authority who sets the signals.

The users behave the way they minimize their own disutility or costs. As disutility may mainly be considered the time spent in the network, but also the use of gas or the simplicity of the chosen path. Their options are the different choices of paths to go from their origin to their destination. Obviously their choice depends on how fluently they can drive, which means that their behaviour depends on the signal setting and on the other users' choices.

The traffic authority's goal is to minimize the total time spent in the network by all users. To achieve this objective the traffic authority disposes of several possibilities. They are outlined in an example of a simple crossroad. First the arrangement of the lanes has to be chosen. Obviously the result is not the same for different arrangements and different flows. The underlying optimization problem is the so-called *Traffic Design Problem*. The design of the urban network can only be seen as a long term tool for the traffic authority, since every change concernes the infrastructure. Another way to influence the traffic flow is the phasing at every signal. For every simple crossroad there are several possible phasings that lead to different results. The phasing change is a very useful mid-term method to stear traffic flows. Very easy to handle is the length of a phase. The optimal determination of the phase lengths is called *Traffic Control Problem*. It is the traffic authority's most powerful short-term tool and we will discuss it intensively. There are still some other ways to influence the traffic flow, that are nowadays also very handy, like for example variable speed limitations to change the costs of a certain path. They will not be treated in this work.

Clearly we have two optimization problems. The users' problem is called *Traffic Assignment Problem*, because the goal is to find out, how the drivers behave, i.e. to assign traffic flows to a given network with given signal settings. The traffic authority has to solve the *Traffic Control Problem*. In the beginning it makes sense to treat these two problems seperately, but it is evident, that they are correlated. Once the traffic authority has taken its decision, the drivers adapt themselves to the new circumstances, which leads to a new situation for the traffic authority and so on. So, further down we have to combine the two subproblems (*Combined Traffic Control-Assignment Problem*).

2.2 The Model

2.2.1 The Network

To deal mathematically with an urban transportation network we need a *model* of the infrastructure and a certain number of *parameters* and *variables*. The network is modelled by an oriented graph, where the arcs represent the streets and the nodes the crossroads. To maintain the particularities of the network design (for example lanes, where one is only allowed to turn left), we probably have to introduce some additional arcs and nodes at the crossroads. So actually the arcs represent the lanes and the nodes passages from one lane to another (across a crossroad or just a lane change).

2.2.2 The Parameters and Variables

In traffic theory there are *static* and *dynamic* models. In the dynamic version the parameters and variables become time dependent functions. Often the solution algorithms of the static problems cannot be applied to the dynamic version. In what follows we line out the dynamic parameters and variables. One gets the static parameters and variables by omitting the time dependency.

Traffic Flow Variables

The traffic flow variables describe the behaviour of the network users. Each user has an origin and a destination. This O-D pair (origin-destination) is connected by several paths and to each O-D pair a demand is assigned. The number of users on a certain link or path is taken as a continuous variable. Flows are defined as number of vehicles per time unit, so their dimension is, if we choose a minute as a time unit, $[\frac{1}{min}]$.

$x_{ap}^{rs}(t)$	=	number of vehicles on link a from r to s along path p at time t.
$u_{ap}^{rs}(t)$	=	entry flow on link a from r to s along path p at time t.
$v_{ap}^{rs}(t)$	=	exit flow on link a from r to s along path p at time t.
$x_a(t)$	=	number of vehicles on link a at time t. $x_a(t) = \sum_{rs}\sum_p x_{ap}^{rs}(t)$
$u_a(t)$	=	entry flow on link a at time t. $u_a(t) = \sum_{rs}\sum_p u_{ap}^{rs}(t)$
$v_a(t)$	=	exit flow on link a at time t. $v_a(t) = \sum_{rs}\sum_p v_{ap}^{rs}(t)$
$U_{ap}^{rs}(t)$	=	cumulative number of vehicles entering link a from r to s along path p at time t.
$V_{ap}^{rs}(t)$	=	cumulative number of vehicles leaving link a from r to s along path p at time t.
$h_p^{rs}(t)$	=	flow departed from r to s along path p at time t.
$D^{rs}(t)$	=	demand from origin r to destination s at time t.

Signal Control Parameters

In what follows an intersection is a group of nodes that concern the same signal/crossroad. Remember that we added some nodes in order to respect the traffic rules on the lanes. To every intersection a certain phasing is assigned and all phases form a cycle. Once a cycle is completed it starts again.

Cycle	:	A complete sequence of signal indications.
Phase	:	The portion of a signal cycle allocated to any combination of traffic movements receiving simultaneous right-of-way.
Cycle Length	:	The total time for a signal to complete one cycle, denoted as C or also $C_i(t)$ (cycle length for intersection i at time t).
Green Time	:	The time within a given phase during which the green indication is shown, denoted as G or $G_i^m(t)$ (green time for phase m of intersection i at time t, this is actually a decision variable of the traffic authority).
Minimum Green Time	:	Minimum guaranteed green time, denoted as $G_i^{m,min}(t)$ (minimum green time for phase m of intersection i at time t).
Lost Time	:	The time during which the intersection is not effectively used by any traffic movement. This occurs during the "yellow", "all red" and the beginning of each phase, denoted as L.
$\theta_{ia}^m(t)$	=	$\begin{cases} 1 \text{ if link } a \text{ is in phase } m \text{ at time } t \\ 0 \text{ otherwise} \end{cases}$

Signal Control Variables

The signal control variables are the values, the traffic authority can influence directly. It is the green time. The green split is a variable that is easier to handle, but its meaning is the same.

$G_i^m(t)$ = Green time for phase m of intersection i at time t.
$g_i^m(t)$ = Green split for phase m of intersection i at time t.
$g_a(t)$ = Green split for lane a at time t.

The green time split is defined as follows:

$$g_i^m(t) = \frac{G_i^m(t) - G_i^{m,min}(t)}{C_i(t) - \sum_m (L_i^m(t) + G_i^{m,min}(t))} = \frac{G_i^m(t) - G_i^{m,min}(t)}{\sum_m (G_i^m(t) - G_i^{m,min}(t))} \quad (2.1)$$

$$g_a(t) = \sum_m \theta_{ia}^m(t) g_i^m(t) \quad (2.2)$$

where $L_i^m(t)$ is the lost time at phase m of intersection i at time t. Clearly we have the relationship:

$$\sum_i g_i^m(t) = 1 \quad (2.3)$$

Note that in our model the cycle length has no importance, because the green splits are taken as continous variables. In practice it could be important. The shorter the cycle, the more updated to the actual traffic amount are the signal settings, but the smaller is its efficiency due to the constant lost time.

Cost Functions

The costs or disutilities are mainly represented as the time spent on the link, on the path or in the network. This work does not deal with cost functions depending on other parameters than the time spent in the network like for example the use of gas.

$\tau_a(t)$ = travel cost/time for flows entering link a at time t.
$c_p^{rs}(t)$ = travel cost/time for path p between O-D pair (r, s) for flows departing from origin r at time t.

If we define the costs as the time spent in the network, $c_p^{rs}(t)$ can be expressed recursively:

$$c_p^{rs}(t) = \tau_{a_1}(t) + \tau_{a_2}(t + \tau_{a_1}(t)) + \cdots \quad (2.4)$$

where p is composed of the links a_1,a_2,$\cdots$. It seems natural, that the link time function $\tau_a(t)$ depends on the green splits. The bigger the green split, the longer the green phase, the less the congestion, the lower the link travel cost. So it is decreasing with increasing green split. Let us have a look at the dependency on the number of vehicles on the link. Even if the link is empty it takes time to get from one end to the other. This value is called free flow travel time. It is easy to see that the link travel time is increasing with the number of vehicles. So, instead of $\tau_a(t)$ it seems more natural to write $\tau_a(x_a(t), g_a(t))$ or even $\tau_a(u_a(t), x_a(t), g_a(t))$.

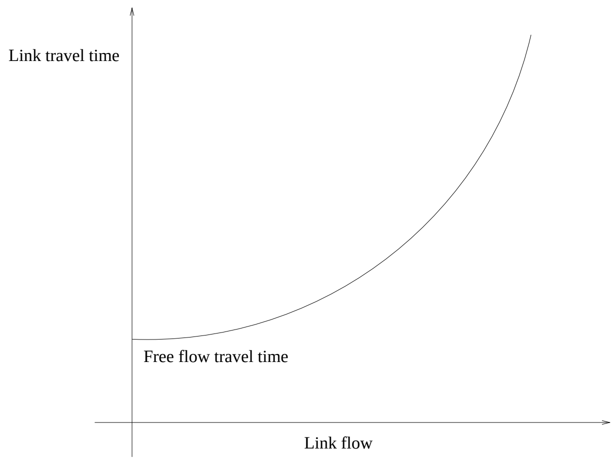

Figure 2.1: The Link Travel Time Function

The figure 2.1 shows the general behaviour of the link travel time in dependency to the link flow. We give now two examples of an explicit function form for $\tau_a(t)$.

The Linear Queuing Delay Function

$$\tau_a(t) = \tau_a(0) + \frac{x_a(t)}{s_a g_a(t)} \tag{2.5}$$

where $\tau_a(0)$ is the free flow travel time and s_a is the saturation rate of link a. The saturation rate is the maximum flow rate, i.e. the maximum number of vehicles that could pass per time unit if the green signal were available 100% of the time.

The Highway Capacity Manual Link Performance Function

$$\tau_a(t) = 0.38\, C_i(t) \frac{(1 - g_a(t))^2}{1 - \frac{u_a(t)}{s_a}} + 173\,(\rho_a(t))_2 \left[(\rho_a(t) - 1) + \sqrt{(\rho_a(t) - 1)^2 + 16\, \frac{\rho_a(t)}{s_a}} \right] \tag{2.6}$$

where C_i is the cycle length of intersection i to which a belongs and $\rho_a(t) = \frac{u_a(t)}{s_a g_a(t)}$

is the degree of current saturation.

2.3 Traffic Assignment

The traffic assignment problem treats the question, which way the traffic users choose, once a signal setting is given. It is evident that it takes time until a general behaviour is established, but for a given signal setting there will be found an equilibrium once. This will be the case, if no user sees a possibility to lower his travel costs. This means that all other paths leading from the user's origin to his destination have the same or higher costs. This means again that either the cost of a path is minimal or the path is not used. This equilibrium is called *Wardrop Equilibrium*. It is the traffic theoretical counterpart to the Nash equilibrium of game theory. Actually, a Wardrop equilibrium is a Nash equilibrium, as we will see in the next chapter.

The Static Case

We give now first the formulation of the traffic assignment problem for the static case. Static means, that there is no time dependency, in particular the O-D demands are constant. We will see later that the problem below has exactly the Wardrop equilibrium as solution, and that the function $z(\mathbf{u})$ is therefore the objective function of the traffic assignment problem.

$$\begin{array}{rrcl} & \min z(\mathbf{u}) & = & \sum_a \int_0^{u_a} \tau_a(\omega) d\omega \\ \text{subject to} & \sum_p h_p^{rs} & = & D^{rs} \\ & h_p^{rs} & \geq & 0 \\ & u_a & = & \sum_{rs} \sum_p h_p^{rs} \delta_{a,p}^{rs} \end{array} \tag{2.7}$$

where $\delta_{a,p}^{rs} = \begin{cases} 1 & \text{if link } a \text{ belongs to path } p \\ 0 & \text{otherwise} \end{cases}$

The objective function has to be optimized with respect to the flows h_p^{rs}, the decision variables of the user community. Note that a flow cannot be the decision variable of a network user. He can only decide about his route choices. That is why we introduce the notion of *user community*. To solve this problem we use the method of the Lagrangian multipliers. We define the auxiliary function $L_{ass}(\mathbf{h}, \lambda)$:

$$L_{ass}(\mathbf{h}, \lambda) = z(\mathbf{u}(\mathbf{h})) + \sum_{rs} \lambda_{rs}(D_{rs} - \sum_{rs} h_p^{rs}) \tag{2.8}$$

We have now an equivalent problem to 2.7:

$$\begin{array}{rrcl} & \min L_{ass}(\mathbf{h}, \lambda) & = & z(\mathbf{u}(\mathbf{h})) + \sum_{rs} \lambda_{rs}(D_{rs} - \sum_{rs} h_p^{rs}) \\ \text{subject to} & h_p^{rs} & \geq & 0 \end{array} \tag{2.9}$$

This leads us to the following optimality conditions:

$$h_p^{rs}\frac{\partial L_{ass}(\mathbf{h},\lambda)}{\partial h_p^{rs}} = 0 \tag{2.10}$$

$$\frac{\partial L_{ass}(\mathbf{h},\lambda)}{\partial h_p^{rs}} \geq 0 \tag{2.11}$$

$$\frac{\partial L_{ass}(\mathbf{h},\lambda)}{\partial \lambda_{rs}} = 0 \tag{2.12}$$

Calculating the different terms we get:

$$\frac{\partial L_{ass}(\mathbf{h},\lambda)}{\partial h_p^{rs}} = \frac{\partial z(\mathbf{u}(\mathbf{h}))}{\partial h_p^{rs}} + \frac{\partial}{\partial h_p^{rs}}\sum_{rs}\lambda_{rs}(D_{rs} - \sum_p h_p^{rs}) \tag{2.13}$$

$$\frac{\partial z(\mathbf{u}(\mathbf{h}))}{\partial h_p^{rs}} = \sum_b \frac{\partial z(\mathbf{u})}{\partial u_b}\frac{\partial u_b}{\partial h_p^{rs}} \tag{2.14}$$

$$\frac{\partial z(\mathbf{u})}{\partial u_b} = \frac{\partial}{\partial u_b}\sum_a \int_0^{u_a} \tau_a(\omega) \quad d\omega = \tau_b \tag{2.15}$$

$$\frac{\partial u_b}{\partial h_p^{rs}} = \delta_{b,p}^{rs} \tag{2.16}$$

$$\frac{\partial z(\mathbf{u}(\mathbf{h}))}{\partial h_p^{rs}} = \sum_b \tau_b \delta_{b,p}^{rs} = c_p^{rs} \tag{2.17}$$

$$\frac{\partial}{\partial h_p^{rs}}\sum_{rs}\lambda_{rs}(D_{rs} - \sum_p h_p^{rs}) = -\lambda_{rs} \tag{2.18}$$

Finally we get:

$$\frac{\partial}{\partial h_p^{rs}} L_{ass}(\mathbf{h},\lambda) = c_p^{rs} - \lambda_{rs} \tag{2.19}$$

The optimality conditions are now:

$$h_p^{rs}(c_p^{rs} - \lambda_{rs}) = 0 \tag{2.20}$$

$$c_p^{rs} - \lambda_{rs} \geq 0 \tag{2.21}$$

$$h_p^{rs} \geq 0 \tag{2.22}$$

Now we have arrived at the Wardrop equilibrium. Equation 2.21 imposes the interpretation of λ_{rs} as minimal path travel time between the O-D pair (r, s). It says that all path travel times from r to s are at least as great as λ_{rs}. Equation 2.20 says, that only the paths with minimal path travel time are frequented. If the path travel time is minimal, $c_p^{rs} - \lambda_{rs}$ is zero, and h_p^{rs} can take positive values. If the path travel time is not minimal, necessarily h_p^{rs} has to be zero.

The whole development shows, that the objective function of the traffic assignment problem is $z(\mathbf{u}) = \sum_a \int_0^{u_a} \tau_a(\omega) \quad d\omega$. Unfortunately there is not any economical interpretation of this term.

The Dynamic Case

The hypothesis that the O-D demands do not change with time is not realistic. Nevertheless it gives us a good idea about users' behaviour in a network. To treat the dynamic case there are two main options. Either we continue the flow based analytical model or we proceed by simulation. In continuation we will work with the flow based model. But the differences between the two options will be lined out further down.

To be sure that the model is still consistent in the dynamic case, we have to add some conditions that take care about the consequences of the actual flow pattern to the future. We have to be sure that the actual amount of vehicles at a crossroad is at the next crossroad, once the time it takes to go from one to the other elapsed. There are several classes of conditions:

Link dynamic equations
The change rate of the number of vehicles on link a on path p from r to s at time t is determined by the difference between the entry and exit flow of link a at time t with respect to path p connecting the O-D pair (r, s)

$$\frac{dx_{ap}^{rs}(t)}{dt} = u_{ap}^{rs}(t) - v_{ap}^{rs}(t) \qquad (2.23)$$

Flow conservation equations
The entry flow of a link a on path p from r to s at time t is either equal to the flow departing from r to s on path p at time t if a is the first link of path p, or equal to the exit flow of the preceeding link at time t with respect to the path p connecting r and s.

$$u_{ap}^{rs}(t) = \begin{cases} h_p^{rs}(t), & a \text{ is the first link on path } p \\ v_{bp}^{rs}(t), & a \text{ is after } b \text{ on path } p \end{cases} \qquad (2.24)$$

Flow propagation equations
A travel unit, that enters link a at time t, leaves link a only at time $t + \tau_a(t)$. The Flow prpagation equations take this fact in account.

$$\begin{aligned} V_{ap}^{rs}(t) &= \int_{\omega+\tau_a(\omega)\leq t} u_{ap}^{rs}(\omega) \quad d\omega && (2.25) \\ U_{ap}^{rs}(t) &= U_{ap}^{rs}(0) + \int_0^t u_{ap}^{rs}(\omega) \quad d\omega && (2.26) \\ V_{ap}^{rs}(t) &= V_{ap}^{rs}(0) + \int_0^t v_{ap}^{rs}(\omega) \quad d\omega && (2.27) \end{aligned}$$

Definitional constraints
The link flows are defined as the sum over all path flows passing by that link.

$$\begin{aligned} u_a(t) &= \sum_{r,s,p} u_{ap}^{rs}(t) && (2.28) \\ v_a(t) &= \sum_{r,s,p} v_{ap}^{rs}(t) && (2.29) \\ x_a(t) &= \sum_{r,s,p} x_{ap}^{rs}(t) && (2.30) \end{aligned}$$

Boundary conditions
At the beginning the network is empty.

$$\begin{aligned} U_{ap}^{rs}(0) &= 0 && (2.31) \\ V_{ap}^{rs}(0) &= 0 && (2.32) \\ x_{ap}^{rs}(0) &= 0 && (2.33) \end{aligned}$$

Non-negativity conditions
The flows must be non-negative.

$$\begin{aligned} u_{ap}^{rs}(t) &\geq 0 && (2.34) \\ v_{ap}^{rs}(t) &\geq 0 && (2.35) \\ x_{ap}^{rs}(t) &\geq 0 && (2.36) \end{aligned}$$

The optimization problem is:

$$\begin{array}{l} \min \int_0^T \sum_a \int_0^{u_a(t)} \tau_a(\omega(t)) \quad d\omega(t)\, dt \\ \text{subject to the constraints listed above} \end{array} \tag{2.37}$$

To ensure that the optimum is taken over the whole time horizon an integral over this time horizon is added. For better justification of the dynamic traffic assignment problem we refer to the appendix A.

The solution of the dynamic traffic assignment problem is more problematic, because the decision variables have become functions. Some special algorithms are necessary.

Note that to calculate the path costs it is necessary to know the future link travel times. This is of course a rather unrealistic hypothesis. There are some ways to introduce uncertainty to the users' perception that we will see further down in the *Stochastic User Equilibrium*. But the analytical flow based model has no use while investigating the effect that traffic information could have to the users' behaviour, since the users are already perfectly informed. For this purpose we should use simulation. The difference to the analytical model is that the simulated network users may change their path choices while they are underway. In the analytical model they decide at the beginning, which way they are going to take. In the simulation model the path choices get updated depending on how the network situation changed. Obviously they have to decide at every crossroad again, which path they prefer and the path choices may be different to the ones taken before. This might be due to an unexpected traffic development or to better information.

2.4 Traffic Control

The traffic authority's goal is to minimize the general time spent in the network. We investigate first the best possible state of the system, the so-called *System Optimum*. This is not necessarily the state the traffic authority can achieve, as we will see further down.

The formulation of the optimization problem is:

$$\begin{array}{lrcl} & \min Z(\mathbf{u}) & = & \sum_a u_a \tau_a(u_a) \\ \text{subject to} & \sum_p h_p^{rs} & = & D^{rs} \\ & h_p^{rs} & \geq & 0 \\ & u_a & = & \sum_{rs} \sum_p h_p^{rs} \delta_{a,p}^{rs} \end{array} \tag{2.38}$$

The objective function has to be minimized with respect to the flows h_p^{rs}. To solve this problem we use again the method of the Lagrangian multipliers. We define the auxiliary function $L_{con}(\mathbf{h}, \pi)$:

$$L_{con}(\mathbf{h}, \pi) = Z(\mathbf{u}(\mathbf{h})) + \sum_{rs} \pi_{rs}(D_{rs} - \sum_{rs} h_p^{rs}) \tag{2.39}$$

We have now an equivalent problem to 2.38:

$$\begin{array}{lrcl} & \min L_{con}(\mathbf{h}, \pi) & = & Z(\mathbf{u}(\mathbf{h})) + \sum_{rs} \pi_{rs}(D_{rs} - \sum_{rs} h_p^{rs}) \\ \text{subject to} & h_p^{rs} & \geq & 0 \end{array} \tag{2.40}$$

This leads us to the following optimality conditions:

$$h_p^{rs} \frac{\partial L_{con}(\mathbf{h}, \pi)}{\partial h_p^{rs}} = 0 \tag{2.41}$$

$$\frac{\partial L_{con}(\mathbf{h}, \pi)}{\partial h_p^{rs}} \geq 0 \tag{2.42}$$

$$\frac{\partial L_{con}(\mathbf{h}, \pi)}{\partial \pi_{rs}} = 0 \tag{2.43}$$

Calculating the different terms we get:

$$\frac{\partial L_{con}(\mathbf{h}, \pi)}{\partial h_p^{rs}} = \frac{\partial Z(\mathbf{u}(\mathbf{h}))}{\partial h_p^{rs}} + \frac{\partial}{\partial h_p^{rs}} \sum_{(r,s)} \pi_{rs}(D_{rs} - \sum_{p} h_p^{rs}) \tag{2.44}$$

$$\frac{\partial}{h_p^{rs}} \sum_{(r,s)} \pi_{rs}(D_{rs} - \sum_{p} h_p^{rs}) = -\pi_{rs} \tag{2.45}$$

$$\begin{aligned} \frac{\partial}{\partial h_p^{rs}} Z(\mathbf{u}(\mathbf{h})) &= \sum_b \frac{\partial Z(\mathbf{u})}{\partial u_b} \frac{\partial u_b}{\partial h_p^{rs}} \\ &= \sum_b \frac{\partial Z(\mathbf{u})}{\partial u_b} \delta_{b,p}^{rs} \\ &= \sum_b \delta_{b,p}^{rs} \frac{\partial}{\partial u_b} \sum_a u_a \tau_a(u_a) \\ &= \sum_b \delta_{b,p}^{rs} [\tau_b(u_b) + u_b \frac{d\tau_b(u_b)}{du_b}] = \sum_b \delta_{b,p}^{rs} \tilde{\tau}_b \end{aligned} \tag{2.46}$$

$\tilde{\tau}_b$ can be interpreted as the marginal contribution of an additional flow unit to the total travel time on this link. τ_b is the travel time perceived by this unit and $\frac{d\tau_b}{du_b}$ is the supplementary travel time for all the users that are already on link b due to the increased congestion. We finally have:

$$\frac{\partial Z(\mathbf{u}(\mathbf{h}))}{\partial h_p^{rs}} = \sum_b \delta_{b,p}^{rs} \tilde{\tau}_b = \tilde{c}_p^{rs} \tag{2.47}$$

where $\tilde{c}_p^{rs}$ is the marginal cost on path p.

The optimality conditions are now:

$$\begin{aligned} h_p^{rs}(\tilde{c}_p^{rs} - \pi_{rs}) &= 0 & (2.48) \\ \tilde{c}_p^{rs} - \pi_{rs} &\geq 0 & (2.49) \end{aligned}$$

Analogous to the traffic assignment problem we may interprete that the system is in its optimum when either the marginal cost of a path is minimal or the path is not used. Note that the flows are not the decision variables of the traffic authority. The traffic authority's tools are the green splits. The investigated optimum is therefore an absolute system optimum and represents a benchmark. The minimum the traffic authority can achieve is probably somewhat higher and the system optimum might only be achieved with the cooperation of the network users. The optimal situation the traffic authority can provoque can be calculated in a similar way. We treat directly the dynamic case. The optimization problem is:

$$\begin{array}{rrcl} & \min Z_g & = & \int_0^T \sum_a u_a(t)\tau_a(u_a(t), x_a(t), g_a(t)) \quad dt \\ \text{subject to} & \sum_m g_i^m(t) & = & 1 \\ & g_a(t) & = & \sum_m g_i^m(t)\theta_{ia}^m(t) \\ & g_i^m(t) & \geq & 0 \end{array} \tag{2.50}$$

Of course all the conditions for the dynamic system have to be respected, too. This time the objective function has to be minimized with respect to $g_i^m(t)$, the traffic authority's decision variables. The Lagrange function becomes:

$$L_g(g_i^m(t), \pi(t)) = \sum_a u_a(t)\tau_a(t) + \sum_i \pi_i(t)(1 - \sum_m g_i^m(t)) \tag{2.51}$$

The optimality constraints are:

$$g_i^m(t)\frac{\partial L_g}{\partial g_i^m(t)} = 0 \quad (2.52)$$
$$\frac{\partial L_g}{\partial g_i^m(t)} \geq 0 \quad (2.53)$$
$$g_i^m(t) \geq 0 \quad (2.54)$$

Calculating the different terms we get:

$$\frac{\partial L_g}{\partial g_i^m(t)} = \frac{\partial}{\partial g_i^m(t)}\sum_a u_a(t)\tau_a(t) + \frac{\partial}{\partial g_i^m(t)}\sum_i \pi_i(t)(1-\sum_m g_i^m(t)) \quad (2.55)$$
$$\frac{\partial}{\partial g_i^m(t)}\sum_i \pi_i(t)(1-\sum_m g_i^m(t)) = -\pi_i \quad (2.56)$$
$$\begin{aligned}\frac{\partial}{\partial g_i^m(t)}\sum_a u_a(t)\tau_a(t) &= \sum_a u_a(t)\frac{\partial \tau_a(t)}{\partial g_a(t)}\frac{\partial g_a(t)}{\partial g_i^m(t)} \\ &= \sum_a u_a(t)\frac{\partial \tau_a(t)}{\partial g_a(t)}\theta_{ia}^m(t) \\ &= \sum_a \tilde{\tau}_a(t)\theta_{ia}^m(t) \\ &= \tilde{c}_i^m(t) \quad (2.57)\end{aligned}$$

where $\tilde{\tau}_a(t)$ can be interpreted as the marginal link delay and $\tilde{c}_i^m(t)$ as the marginal phase delay. The optimality conditions become:

$$g_i^m(t)(\tilde{c}_i^m(t) - \pi_i(t)) = 0 \quad (2.58)$$
$$\tilde{c}_i^m(t) - \pi_i(t) \geq 0 \quad (2.59)$$
$$g_i^m(t) \geq 0 \quad (2.60)$$

Again the multipliers can be interpreted as the minimal marginal phase delay. The marginal phase delay should be minimal or zero.

Note that the above calculus delivers us the optimal signal setting to a given flow pattern, and not a generally optimal signal setting. It is likely that the given flow pattern is not a user equilibrium to the calculated signal settings.

2.5 Stochastic User Optimum

As already discussed, it is rather unprobable that all the users have perfect information about the path costs. The introduction of random perception errors would make the model more realistic. There are different models for these perception errors. Well known are the *C-Logit Model*, the *Multinomial Logit Model* and the *Probit Model*. The main difference is the random distribution of the errors and the hypothesis about its independency.

To every path p leading from origin r to destination s a perception error $\epsilon_p^{rs}(t)$ is assigned. Hence the probability that the user chooses path p is the probability, that he thinks p is the quickest way to get from r to s.

$$P_p^{rs}(t) = Prob[c_p^{rs}(t) + \epsilon_p^{rs}(t) \leq c_q^{rs}(t) + \epsilon_q^{rs}(t) \quad \forall q] \tag{2.61}$$

At the stochastic user optimum the flows are therefore:

$$h_p^{rs}(t) = D^{rs}(t) P_p^{rs}(t) \tag{2.62}$$

The network users can be subdivided in different classes. These can be described by the variance of their perception error. So perfectly guided users have for example a zero perception error. With increasing variance of the perception error the users get less and less guided or informed. The size of each user class is determined by its demand, because to every class n of users, a O-D demand $D_{np}^{rs}(t)$ is assigned. Obviously the flows also have to be treated seperately with respect to every class. The stochastic user optimum is therefore characterized as follows:

$$P_{np}^{rs}(t) = Prob[c_p^{rs}(t) + \epsilon_{np}^{rs}(t) \leq c_q^{rs}(t) + \epsilon_{nq}^{rs}(t) \quad \forall q] \tag{2.63}$$

$$h_{np}^{rs}(t) = D_n^{rs}(t) P_{np}^{rs}(t) \tag{2.64}$$

Sometimes a last class of habitual users is added. This are users that take a predetermined way independent to the network situation. The best example are public transport services. But this class does not represent an optimization problem, but leads only to an already known increase of link travel costs.

Note that also a model with perfect informed users can be described this way. The perception error ϵ is just zero. From now on we assume that the user optimum is a stochastic user optimum, but to make the model not too heavy we will not make the difference between the different classes of user types.

2.6 Combined Control-Assignment Problem

As already seen, the link time travel function τ_a depends on the green splits. Then also the path travel times depend on a change of the signal settings and therefore an optimization of the traffic control problem has a different user behaviour as consequence. This again means that the previously calculated signal setting is likely to be not optimal anymore for the new flow pattern. It is therefore necessary to take the interactions between the traffic control and the traffic assignment problem into account. The goal is to find an equilibrium, i.e. a signal setting that is optimal to the flow pattern, and a flow pattern that is optimal to the signal setting.

$$\begin{array}{rcl} \mathbf{g}^* & = & \arg\max_{\mathbf{g}} Z_{control}(\mathbf{g}, \mathbf{h}^*) \\ \mathbf{h}^* & = & \arg\max_{\mathbf{h}} Z_{assignment}(\mathbf{g}^*, \mathbf{h}) \end{array} \tag{2.65}$$

where $Z_{control}$ is the total delay function, the objective function of the traffic control problem, and $Z_{assignment}$ is the objective function of the traffic assignment problem.

A special look should be taken at the way, a stable solution could be reached. One possibility is the above described iteration, this means that we solve alternatively the traffic assignment problem and then the traffic control problem. If this method converges, it converges to an equilibrium. Another option is to take already a change of the drivers' route choices in account while looking for the best green splits. For this purpose we are going to use a game theoretic interpretation of the traffic problem.

For a more detailed analysis of the static traffic problem consult [4]. For the dynamic problem see also [1] and [3].

Chapter 3

Game Theoretic Interpretation of the Traffic Problem

In this chapter the different traffic problems will be interpreted as games. This enables the formulation of solution algorithms, specially for the combined traffic control-assignment problem.

3.1 The Traffic Assignment Game

The traffic assignment problem handles as already seen with the distribution of traffic flows over a given transportation network. Its solution is a Wardrop equilibrium, where every driver chooses the path with the minimum path travel time. In the Wardrop equilibrium a driver cannot lower his travel time given the other drivers' behaviour. This is equivalent to the definition of the Nash equilibrium. Hence, it is likely to formulate the assignment problem as a game:

players	the drivers
strategies	the different paths they can choose to get from their origin to their destination
payoffs	the negative travel time

As payoffs we take the negative travel time, because in game theory the payoff is always to maximize.

Since the flows correspond to the drivers' route choices, we take the flows $\mathbf{h}$ as strategies and H as strategy space. Of course a driver cannot have a flow as a strategy. To be able to handle with flows as strategies we have to transform the game a little. We define it as a monopoly game, where the whole user community is one player:

player	the user community
strategies	the different flows
payoffs	the negative objective function of the traffic assignment problem

We have seen that the optimum of the traffic assignment objective function is the Wardrop equilibrium. So the result of the new monopoly game will be the same and the drivers will not cooperate, although they have been put together.

3.2 The Traffic Control Game

In the traffic control problem the traffic authority aims at minimizing the total travel time. As a game this gives:

player	the traffic authority
strategies	the different signal settings
payoff	the negative total travel time

The game above is a formulation of the problem the traffic authority encounters. We take the green splits $\mathbf{g}$ as the strategies and G as the strategy space of the traffic authority. The problem can be widen by adding the flows to the traffic authority's strategies. This way the solution can be seen as a benchmark to the combined traffic control-assignment problem. The absolute optimum is the equilibrium of the traffic authority's game if we add also the different phasings and the network design possibilities to its strategies. Obviously all these traffic control games are monopoly games.

3.3 The Combined Traffic Control-Assignment Game

The fact that the traffic authority and the network users do not have the same objectives, but their payoffs depend on the other party's decision, encourages us to define a game between the traffic authority and the network users.

3.3.1 The Cournot Game

Comparing the equations 1.3 and 2.65 we see that the equilibrium of the iteration method corresponds to the equilibrium of a Cournot game. In fact the interaction traffic authority-network users can be interpretated as a Cournot game:

players	the traffic authority and the network user community
strategies	the green splits for the traffic authority and the flows for the network user community
payoffs	negative objective functions of the traffic control and assignment problem

This game corresponds exactly to the iteration method. Although in a Cournot game the players take their decision at the same time, in our specific game only one player changes his strategy at once. This can be explained like that: In the beginning we start with any signal setting and its optimal flow pattern. On the next level of the game, the drivers have no need to change their strategy, because it is already optimal to the given signal setting. But the traffic authority has to adapt his strategy to the given flow pattern. On the next level the traffic authority can keep its signal setting, because it is optimal to the given flow pattern thanks to the change undertaken last time. Now the traffic users have to reevaluate their strategies and so on.

3.3.2 The Stackelberg Game

A more exact analysis shows that the traffic authority can impose a certain type of behaviour to the network users, because they are always aiming at the Wardrop equilibrium. The network users would never behave different to the Wardrop strategy in the hope that they could influence this way the signal settings. We have therefore a natural role distribution of leader and follower. The traffic authority is the leader, the network users the followers. We can define a Stackelberg game:

leader	the traffic authority
follower	the network user community
strategies	the green splits for the traffic authority and the flows for the network user community
payoffs	negative objective functions of the traffic control and assignment problem

The equilibrium can therefore be expressed as follows:

$$\begin{array}{rcl} \mathbf{g}^* & = & \arg\max_{\mathbf{g}} Z_{control}(\mathbf{g}, \mathbf{h}^*) \\ \text{subject to} \quad \mathbf{h}^* & = & \arg\max_{\mathbf{h}} Z_{assignment}(\mathbf{g}, \mathbf{h}) \end{array} \tag{3.1}$$

In the Stackelberg game the traffic authority uses his knowledge about the users' behaviour, that is that they aim at the Wardrop equilibrium. It simply chooses the signal setting with the most advantageous Wardrop equilibrium that it can anticipate.

The traffic authority's payoff is therefore in the Stackelberg game at least as high as in the Cournot game. We see that the Stackelberg equilibrium fullfills the conditions of a Cournot equilibrium. but the Cournot game can end in another, for the traffic authority less advantageous equilibrium. This can depend on the initial situation, as seen on page 4.

This chapter followed the ideas of [1].

Chapter 4

The Solution Algorithms

As already mentionned above the decision variables of the different traffic problems become time dependent functions in the dynamic case. It is therefore impossible to solve these optimization problems by linear programming and we have to help ourselves with other algorithms. Further down we have also to proceed to a discretisation of these time dependent functions. Before attacking the solution algorithms concretely, we will have a short look at the applied method.

4.1 The Convex Combination Method

The Problem is the following:

$$\begin{array}{rlcl} & \min z(\mathbf{x}) & & \\ \text{subject to} & \sum_i a_{ij} x_i & \geq & b_j \quad \forall j \in \mathcal{J} \end{array} \tag{4.1}$$

The function z is to minimize with respect to $\mathbf{x}$. We proceed by iterations and at the n^{th} iteration we have $\mathbf{x}^n$ as current solution. From $\mathbf{x}^n$ on we would like to improve in the most promising direction. The most promising direction can be the direction with the steepest negative gradient. For any feasible $\mathbf{y}$ the slope in the direction $\mathbf{y} - \mathbf{x}^n$ is:

$$- \bigtriangledown z(\mathbf{x}^n) \frac{\mathbf{y} - \mathbf{x}^n}{\parallel \mathbf{y} - \mathbf{x}^n \parallel} \tag{4.2}$$

The slope of z in the direction of $\mathbf{y} - \mathbf{x}^n$ in the vicinity of $\mathbf{x}^n$ is nothing more than the projection of the negative gradient of z in $\mathbf{x}^n$ in the direction of $\mathbf{y} - \mathbf{x}^n$. But even if the gradient is steep the improvement may be modest because we cannot move very far. So it seems useful to take this aspect into account, too. The further we may go in a certain direction, the more likely we approach to the minimum.

We can define the *Drop* as the product of the directional slope in the vicinity of $\mathbf{x}^n$ with the distance $\| \mathbf{y} - \mathbf{x}^n \|$. If $z(\mathbf{x})$ is convex the drop is a benchmark to the improvement that can be achieved by moving in this direction. It seems natural to move in the direction of the maximal drop. So we have the following subproblem at the n^{th} iteration:

$$\begin{array}{rl} & \max_{\mathbf{y}^n}\{- \bigtriangledown z(\mathbf{x}^n)(\mathbf{y}^n - \mathbf{x}^n)\} \\ \text{subject to} & \mathbf{y}^n \text{ feasible} \end{array} \tag{4.3}$$

This is of course equivalent to

$$\begin{array}{rl} & \max_{\mathbf{y}^n}\{- \bigtriangledown z(\mathbf{x}^n)\mathbf{y}^n\} \\ \text{subject to} & \mathbf{y}^n \text{ feasible,} \end{array} \tag{4.4}$$

because the term $\bigtriangledown z(\mathbf{x}^n)\mathbf{x}^n$ is constant. $\mathbf{y}^n$ being solution of a linear program lies at the boundary, so $\mathbf{x}^{n+1}$ lies somewhere between $\mathbf{x}^n$ and $\mathbf{y}^n$.

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \alpha_n(\mathbf{y}^n - \mathbf{x}^n) \tag{4.5}$$

Once we have found $\mathbf{y}^n$ we have to solve a second subproblem:

$$\alpha_n = \arg\min_{\alpha} z(\mathbf{x}^{n+1}) = \arg\min_{\alpha} z[\mathbf{x}^n + \alpha(\mathbf{y}^n - \mathbf{x}^n)] = \arg\min_{\alpha} z[(1-\alpha)\mathbf{x}^n + \alpha\mathbf{y}^n] \tag{4.6}$$

The last term explains the name of the convex combination method. This second subproblem may be very costy and often a prefixed value is taken for α_n like for example:

$$\alpha_n = \frac{1}{n+1} \tag{4.7}$$

The iteration stops either if a sufficient convergence or a maximum number of iterations is achieved. The convex combination method can finally be described in the following way:

step 0	*initialize*	$\mathbf{x}^1$ and $n = 1$
step 1	*find direction*	find $\mathbf{y}^n$ that solves 4.3
step 2	*determine step size*	α_n
step 3	*move*	$\mathbf{x}^{n+1} = (1-\alpha_n)\mathbf{x}^n + \alpha_n\mathbf{y}^n$
step 4	*check convergence*	if $\lvert\mathbf{x}^{n+1} - \mathbf{x}^n\rvert < \varepsilon$ or if $n = N$ then STOP
		else $n = n + 1$ and go to step 1

Note that the starting point $\mathbf{x}^1$ must be an interior point.

4.2 The Convex Combination Method Applied to the Traffic Assignment Problem

In the traffic assignment problem the function to minimize is according to 2.37:

$$Z_{assignment}(\mathbf{h}) = \int_0^T \sum_a \int_0^{u_a(t)} \tau_a(\omega(t))d\omega(t)\,dt \tag{4.8}$$

At the n^{th} iteration the subproblem 4.4 in step 1 gives:

$$\begin{array}{rcl} \bar{\mathbf{h}}^n(t) & = & \arg\max_{\bar{\mathbf{h}}(t)}\{-\bigtriangledown Z_{assignment}(\mathbf{h}^n(t))\bar{\mathbf{h}}(t)\} \\ & = & \arg\min_{\bar{\mathbf{h}}(t)}\{\int_0^T \sum_a \tau_a(u_a^n)\bar{u}_a\} \\ \text{subject to} & & \bar{\mathbf{h}}(t) \text{ forms an admissible flow pattern with link entry flows } \bar{u}_a \end{array} \tag{4.9}$$

It is easy to see that this subproblem is nothing more than the traffic control problem with constant link travel time functions with respect to the flow. This problem is solved by assigning all drivers to the path with the minimum travel time. This is a *All-or-Nothing Assignment* and we will refer to it also under the name of *Free Flow Assignment*, because the path travel cost remains constant. $\bar{\mathbf{h}}^n(t)$ is therefore the free flow assignment with respect to the path travel times corresponding to the flow pattern $\mathbf{h}^n(t)$. The algorithm to solve the traffic assignment problem is therefore structured like this:

step 0	*initialize*	
step 1	*find free flow assignment* $\bar{\mathbf{h}}^n(t)$	according to the path travel times given by $\mathbf{h}^n(t)$
step 2	*determine step size*	α_n
step 3	*move*	$\mathbf{h}^{n+1}(t) = (1-\alpha_n)\mathbf{h}^n(t) + \alpha_n\bar{\mathbf{h}}^n(t)$
step 4	*check convergence*	if $\lvert\mathbf{h}^{n+1}(t) - \mathbf{h}^n(t)\rvert < \varepsilon\forall t$ or if $n = N$ then STOP else $n = n+1$ and go to step 1

Here of course the updating of the travel times at every iteration is missing. It will be included in step 1 and is called *Network Loading Algorithm*, that will be explained further down. The step size α_n will be taken as in 4.7.

As initialization in step 0 the free flow assignment can be used.

For the case that a stochastic user equilibrium is to find step 1 is to modify in the following sense:

step 1.1	*compute route choice probability according to* $\mathbf{h}^n(t)$
step 1.2	*compute* $\bar{\mathbf{h}}^n(t)$ *according to* 2.62

A special algorithm is needed to compute the route choice probabilities.

4.3 The Convex Combination Method Applied to the Traffic Control Problem

In the traffic control problem the function to minimize is according to 2.50:

$$Z_{control}(\mathbf{g}) = \int_0^T \sum_a u_a(t)\tau_a(g_a(t))dt \tag{4.10}$$

at the n^{th} iteration the subproblem 4.4 in step 1 gives:

$$\begin{array}{rcl} \bar{\mathbf{g}}^n(t) & = & \arg\max_{\bar{\mathbf{g}}(t)}\{-\bigtriangledown Z_{control}(\mathbf{g}^n(t))\bar{\mathbf{g}}(t)\} \\ & = & \arg\min_{\bar{\mathbf{g}}(t)}\{\sum_{\bar{g}_i^m(t)} \frac{\partial}{\partial \bar{g}_i^m(t)} \left(Z_{control}(\mathbf{g}^n(t))\right) \bar{g}_i^m(t)\} \\ \text{subject to } \sum_m \bar{g}_i^m & = & 1 \\ \bar{g}_a(t) & = & \sum_m \bar{g}_i^m(t)\theta_{ia}^m(t) \\ \bar{g}_i^m(t) & \geq & 0 \end{array} \tag{4.11}$$

where $\mathbf{g}$(t) denotes a set of green splits. The secret now is to calculate the derivation

$$\frac{\partial Z_{control}(\mathbf{g}^n(t))}{\partial \bar{g}_i^m(t)}. \tag{4.12}$$

This derivation depends on the game type we choose. It will be treated later.

Once $\bar{\mathbf{g}}$ is found the algorithm to solve the traffic control problem can be described like this:

step 0	*initialize*	
step 1	*find* $\bar{\mathbf{g}}^n(t)$	starting from $\mathbf{g}^n(t)$
step 2	*determine step size*	α_n
step 3	*move*	$\mathbf{g}^{n+1}(t) = (1-\alpha_n)\mathbf{g}^n(t) + \alpha_n\bar{\mathbf{g}}^n(t)$
step 4	*check convergence*	if $\lvert\mathbf{g}^{n+1}(t) - \mathbf{g}^n(t)\rvert < \varepsilon \quad \forall t$ or if $n = N$ then STOP else $n = n+1$ and go to step 1

Here of course step 1 also includes the updating of the link and path travel times.

As initialization in step 0 the uniform green split distribution can be used.

4.4 Network Loading

The goal of network loading is to find time dependent link and path travel times over the whole time horizon, given a flow pattern and a signal setting. Once the flow pattern is given, the link travel times cannot just be calculated by the explicit form of the link travel time functions in the dynamic case, because it is not obvious on which link of his path a driver contributes to a congestion at which instant. The knowledge of the link and path travel times are essential for the traffic assignment and control problems. To handle this problem we have to discretize the flow constraints on page 17. We assume a finite time horizon T. T is subdivided in n equal parts Δt. To simplify we assume, that Δt is precisely the time unit.

Link dynamic equations
The derivation gets now the change rate during Δt. Since Δt is the time unit, it can be neglected.

$$x_{ap}^{rs}(k+1) = x_{ap}^{rs}(k) + u_{ap}^{rs}(k) - v_{ap}^{rs}(k) \tag{4.13}$$

Flow conservation equations

$$u_{ap}^{rs}(k) = \begin{cases} h_p^{rs}(k), & a \text{ is the first link on path } p \\ v_{bp}^{rs}(k), & a \text{ is after } b \text{ on path } p \end{cases} \tag{4.14}$$

Flow propagation equations
The integrals become now sums.

$$V_{ap}^{rs}(k) = \sum_{j+\tau_a(j)\leq k} u_{ap}^{rs}(j) \tag{4.15}$$

$$U_{ap}^{rs}(k) = \sum_{j=0}^{k} u_{ap}^{rs}(j) \tag{4.16}$$

$$v_{ap}^{rs}(k) = V_{ap}^{rs}(k) - V_{ap}^{rs}(k-1) \tag{4.17}$$

Definitional constraints

$$u_a(k) = \sum_{r,s,p} u_{ap}^{rs}(k) \tag{4.18}$$

$$v_a(k) = \sum_{r,s,p} v_{ap}^{rs}(k) \tag{4.19}$$

$$x_a(k) = \sum_{r,s,p} x_{ap}^{rs}(k) \tag{4.20}$$

Boundary conditions

$$
\begin{aligned}
U_{ap}^{rs}(0) &= 0 && (4.21)\\
V_{ap}^{rs}(0) &= 0 && (4.22)\\
x_{ap}^{rs}(0) &= 0 && (4.23)
\end{aligned}
$$

Non-negativity conditions

$$
\begin{aligned}
u_{ap}^{rs}(k) &\geq 0 && (4.24)\\
v_{ap}^{rs}(k) &\geq 0 && (4.25)\\
x_{ap}^{rs}(k) &\geq 0 && (4.26)
\end{aligned}
$$

With the five equations 4.13 - 4.17 we can, once the link travel times $\tau_a(k)$ and the flows $h_p^{rs}(k)$ given, calculate the five unknowns $x_{ap}^{rs}(k)$,$u_{ap}^{rs}(k)$,$v_{ap}^{rs}(k)$,$U_{ap}^{rs}(k)$ and $V_{ap}^{rs}(k)$. The key equation is 4.15, the only one where $\tau_a(k)$ interferes. The network loading algorithm follows the below described idea:

At iteration n we compute the values of $x_{ap}^{rs}(k)$,$u_{ap}^{rs}(k)$,$v_{ap}^{rs}(k)$,$U_{ap}^{rs}(k)$ and $V_{ap}^{rs}(k)$ thanks to the link travel times $\tau_a^n(k)$. After that we can compute $u_a(k)$,$v_a(k)$ and $x_a(k)$ using the definitional constraints. These values enable us to calculate $\tau_a^{new}(k)$, because the link travel time function depends precisely on these values and the green phases, but these do not change. Then we define $\tau_a^{n+1}(k) = (1-\alpha_n)\tau_a^n(k)+\alpha_n\tau_a^{new}(k)$, where $\alpha_n = \frac{1}{n+1}$. We smooth $\tau_a(k)$ this way in order to avoid an alternation. To finish we check convergency. This gives:

step 0	*initialize*
step 1	*compute* $x_{ap}^{rs}(k)$,$u_{ap}^{rs}(k)$,$v_{ap}^{rs}(k)$,$U_{ap}^{rs}(k)$ and $V_{ap}^{rs}(k)$
step 2	*compute* $u_a(k)$,$v_a(k)$ and $x_a(k)$
step 3	*calculate* $\tau_a^{new}(k)$
step 4	*update* $\tau_a^{n+1}(k)$
step 5	*check convergence* else $n = n + 1$ and go to step 1

As initialization in step 0 the free flow travel times can be used.

4.5 The Combined Traffic Control-Assignment Problem

The combined traffic control-assignment problem can now be solved by the following algorithm:

step 0	*initialize*	$\mathbf{h}^0, \mathbf{g}^0, n = 0$
step 1	*solve traffic assignment problem*	$\mathbf{h}^{n+1}$
step 2	*solve traffic control problem*	$\mathbf{g}^{n+1}$
step 3	*check convergence*	if $\mathbf{h}^{n+1} \approx \mathbf{h}^n$ and $\mathbf{g}^{n+1} \approx \mathbf{g}^n$ then STOP else $n = n + 1$ and go to step 1

We see that the combined traffic control problem is mainly a traffic control problem. The traffic assignment algorithm is just used to proceed to the next iteration of the traffic control problem with an updated network. The algorithm can be modified by limiting severely the maximum number of iterations in the subproblems in step 1 and step 2 or by omitting step 1 and updating the flow pattern at every iteration of the traffic control problem.
By the way, only in the traffic control algorithm appear the differences between the different games, more precisely in the term 4.12.

4.5.1 The Cournot Game

As mentionned we have to calculate the term 4.12 with respect to the hypothesis underlying the game. In the Cournot game the traffic authority does not anticipate the network users' reaction and assumes that they will not change their route choices. The authority optimizes its strategy with respect to the actual drivers' behaviour, which will not be influenced by its own strategy in its opinion. More concretely this means that the flows $\mathbf{h}$ do not depend on the green splits $\mathbf{g}$. In order to proceed to the derivation it is handy to redefine the objective function of the traffic control problem. We also discretize the function.

$$Z_{control} = \sum_k \sum_{rs} \sum_p h_p^{rs}(k) c_p^{rs}(k) \tag{4.27}$$

where $c_p^{rs}(k)$ is defined as in 2.4. This is just the path based formulation of the objective function in problem 2.50. This function is now to derivate as seen in 4.11 with respect to the green splits in order to get the gradient of the objective function.

$$\frac{\partial Z_{control}}{\partial g_i^m(k)} = \sum_a \frac{\partial Z_{control}}{\partial g_a(k)} \frac{\partial g_a(k)}{\partial g_i^m(k)} = \sum_a \frac{\partial Z_{control}}{\partial g_a(k)} \theta_{ia}^m(k) \tag{4.28}$$

Of course the sum has only to be taken over the incoming lanes a to i. Now still the objective function is to derivate with respect to $g_a(t)$.

$$\frac{\partial Z_{control}}{\partial g_a(k)} = \sum_l \sum_{r,s} \sum_p h_p^{rs}(l) \frac{\partial c_p^{rs}(l)}{\partial g_a(k)} \tag{4.29}$$

Here we used the hypothesis that the flows are given in the Cournot game.

$$\frac{\partial c_p^{rs}(l)}{\partial g_a(k)} = \begin{cases} \frac{\partial c_p^{rs}(l)}{\partial \tau_a(k)} \frac{\partial \tau_a(k)}{\partial g_a(k)}, & \text{if } l + c_p^{ra}(l) = k \\ 0, & \text{otherwise} \end{cases} \tag{4.30}$$

$$\frac{\partial c_p^{rs}(l)}{\partial \tau_a(k)} = \begin{cases} 1, & \text{if } l + c_p^{ra}(l) = k \\ 0, & \text{otherwise} \end{cases} \tag{4.31}$$

where $c_p^{ra}(l)$ denotes the path travel time from r to a on path p that goes on to s for a network user departing from r at time l. Substituting 4.29 - 4.31 in 4.28 we get:

$$\frac{\partial Z_{control}}{\partial g_i^m(k)} = \sum_a \sum_{r,s} \sum_p \sum_{l+c_p^{ra}(l)=k} \theta_{ia}^m(k) h_p^{rs} \frac{\partial \tau_a(k)}{\partial g_a(k)} \tag{4.32}$$

$\frac{\partial \tau_a(k)}{\partial g_a(k)}$ can be obtained from the explicit function form of τ_a.

Now we have all the necessary information for the dynamic combined traffic control-assignment problem based on a Cournot game model.

4.5.2 The Stackelberg Game

The main difference between the Stackelberg game and the Cournot game is, that the traffic authority anticipates a change of the network users' behaviour. The traffic authority is conscient of the influence of the signal settings on the drivers' route choices. **h** depends on **g**. We have to rewrite therefore equation 4.29:

$$\begin{aligned} \frac{\partial Z_{control}}{\partial g_a(k)} &= \sum_l \sum_{r,s} \sum_p \frac{\partial}{\partial g_a(k)} (h_p^{rs}(l) c_p^{rs}(l)) \\ &= \sum_l \sum_{r,s} \sum_p \left[\frac{\partial h_p^{rs}(l)}{\partial g_a(k)} c_p^{rs}(l) + \frac{\partial c_p^{rs}(l)}{\partial g_a(k)} h_p^{rs}(l) \right] \end{aligned} \tag{4.33}$$

The term $\frac{\partial c_p^{rs}(l)}{\partial g_a(k)}$ will be calculated in the same way as in the Cournot model. Let us have a look at the other term:

$$\frac{\partial h_p^{rs}(l)}{\partial g_a(k)} = \sum_q \frac{\partial h_p^{rs}(l)}{\partial c_q^{rs}(l)} \frac{\partial c_q^{rs}(l)}{\partial g_a(k)} \tag{4.34}$$

$$\frac{\partial c_q^{rs}(l)}{\partial g_a(k)} = \begin{cases} \frac{\partial c_q^{rs}(l)}{\partial \tau_a(k)} \frac{\partial \tau_a(k)}{\partial g_a(k)}, & \text{if } l + c_q^{ra}(l) = k \\ 0, & \text{otherwise} \end{cases} \tag{4.35}$$

$$\frac{\partial c_q^{rs}(l)}{\partial \tau_a(k)} = \begin{cases} 1, & \text{if } l + c_q^{ra}(l) = k \\ 0, & \text{otherwise} \end{cases} \tag{4.36}$$

where q denotes the different paths leading from r to s. To calculate $\frac{\partial h_p^{rs}(l)}{\partial c_q^{rs}(l)}$ we use 2.62. Hence:

$$\frac{\partial h_p^{rs}(l)}{\partial c_q^{rs}(l)} = D^{rs}(l) \frac{\partial P_p^{rs}(l)}{\partial c_q^{rs}(l)} \tag{4.37}$$

The term $\frac{\partial P_p^{rs}(l)}{\partial c_q^{rs}(l)}$ is given by the used route choice probability model. Finally we can substitute 4.34 - 4.37 and 4.30 - 4.31 into 4.33:

$$\begin{aligned} \frac{\partial Z_{control}}{\partial g_i^m(k)} &= \sum_a \theta_{ia}^m(k) \sum_{r,s} \sum_p \left[\frac{\partial h_p^{rs}(l)}{\partial g_a(k)} c_p^{rs}(l) + \frac{\partial c_p^{rs}(l)}{\partial g_a(k)} h_p^{rs}(l) \right] \\ &= \sum_a \theta_{ia}^m(k) \sum_{r,s} \sum_p \sum_{l + c_p^{ra}(l) = k} \left[D^{rs}(l) \left(\sum_q \frac{\partial P_p^{rs}(l)}{\partial c_q^{rs}(l)} \right) c_p^{rs}(l) \frac{\partial \tau_a(k)}{\partial g_a(k)} + \frac{\partial \tau_a(k)}{\partial g_a(k)} h_p^{rs}(l) \right] \\ &= \sum_a \theta_{ia}^m(k) \sum_{r,s} \sum_p \sum_{l + c_p^{ra}(l) = k} \frac{\partial \tau_a(k)}{\partial g_a(k)} \left[c_p^{rs}(l) D^{rs}(l) \sum_q \frac{\partial P_p^{rs}(l)}{\partial c_q^{rs}(l)} + h_p^{rs}(l) \right] \end{aligned} \tag{4.38}$$

Now we have all the necessary information for the dynamic combined traffic control-assignment problem based on a Stackelberg game model.

4.5.3 The Monopoly Game

In the monopoly game the drivers are not to decide but the traffic authority disposes also of the flows as decision variables. This has two main consequences. First the assignment of traffic flows has to be made with another objective function for the traffic assignment algorithm, i.e. the traffic authority's objective function. This function has to be minimized in the traffic assignment algorithm with respect to the flows **h**. Second the traffic authority does not have to assume that the drivers will change their behaviour, because they have the same goal, that is minimizing the total delay. The combined traffic control-assignment algorithm of page 35 can therefore be used for the monopoly game, too, by applying the Cournot traffic control algorithm and a modified traffic assignment algorithm that will be lined out here:

The objective function is the same like 4.27:

$$Z_{assignment} = \sum_k \sum_{rs} \sum_p h_p^{rs}(k) c_p^{rs}(k) \tag{4.39}$$

This time the gradient has to be taken with respect to the flows $\mathbf{h}$.

$$\begin{aligned}\frac{\partial Z_{assignment}}{\partial h_q^{vw}(l)} &= \frac{\partial}{\partial h_q^{vw}(l)} \left(\sum_k \sum_{rs} \sum_p h_p^{rs}(k) c_p^{rs}(k) \right) \\ &= \sum_k \sum_{rs} \sum_p \left(\frac{\partial h_p^{rs}(k)}{\partial h_q^{vw}(l)} c_p^{rs}(k) + \frac{\partial c_p^{rs}(k)}{\partial h_q^{vw}(l)} h_p^{rs}(k) \right) \\ &= \tilde{c}_q^{vw}(l) \end{aligned} \tag{4.40}$$

where $\tilde{c}_q^{vw}(l)$ denotes the marginal path costs. For better understanding compare with 2.47. With this result the combined traffic control-assignment algorithm is known.

4.5.4 The Results

The discussed algorithms have been applied to the urban network from the Boston Back Bay by Owen Jianwen Chen (cf. [1]). The Cournot and Stackelberg models showed an improvement of about 11% and 12% with respect to the total time delay to the existing traffic system. The models have been tested for the two link travel time functions 2.5 and 2.6 and the results are about the same. The optimality gap between the Stackelberg model and the monopoly model is 2%-3%. This represents the difference between the user optimum and the system optimum.

A more detailed description of the convex combination method can be found in [4]. Its application to the traffic problem is taken from [1].

Chapter 5

Traffic as a Bayes Game

This chapter discusses a possible development of the game theoretic interpretation of the traffic problem discussed in the previous chapters. Further research possibilities are lined out in the conclusion.

5.1 The Current Game

As we have seen it is rather unrealistic that all traffic users have perfect information. We helped ourselves by introducing perception errors and by defining a stochastic equilibrium. The whole development served to introduce some uncertainty. Let us have a look at an interpretation of the traffic game as a Bayesian game. Bayesian games deal precisely with uncertainty.

In what follows we try to take more uncertainty aspects in account than in the model seen.

In our model the drivers were subdivided in different user classes with respect to their perception error. We assumed that to every class n we can assign a certain percentage λ_n $(0 \leq \lambda_n \leq 1)$ and a certain perception error ϵ_n (This way we defined the flow of class n from r to s $h_n^{rs} = \lambda_n D^{rs}$). It seems natural to interpret the degree of the perception errors as n types for the user community and assigning them a probability λ_n. But this interpretation is wrong, as we are going to see. The Bayesian equilibrium is caracterized as in 1.6:

$$s_i^* = \arg\max_{s_i} \sum_{\theta_{-i}} P(\theta_{-i}|\theta_i^*) u_i(s_i, s_{-i}^*, (\theta_i^*, \theta_{-i})) \quad \forall i \tag{5.1}$$

The stochastic equilibrium $(\mathbf{g}^*, \mathbf{h}^*)$ is defined by:

$$\begin{array}{rcl} \mathbf{g}^* & = & \arg\max_{\mathbf{g}} Z_{control}(\mathbf{g}, \sum_n h_n^*) \\ \mathbf{h}^* & = & \arg\max_{\mathbf{h}} Z_{assignment}(\sum_n h_n, \mathbf{g}^*) \end{array} \quad (5.2)$$

But the objective functions $Z_{control}$ and $Z_{assignment}$ are not linear. The above given interpretation is therefore wrong in deed. It is necessary to define the types in a precise way.

5.2 The Types

5.2.1 The Drivers' Types

A type is defined as private information that is not common knowledge. The information degree of the drivers' community is described by its user class distribution. Let's say that there are N user classes n. A distribution is a vector $(\lambda_1, \cdots, \lambda_N)$ with $\sum_n \lambda_n = 1$. This distribution is given, but it is probably not exactly known by the traffic authority. Hence we can take the class distribution as the drivers' type. The traffic authority assigns to each distribution a probability.

5.2.2 The Traffic Authority's Type

On the other hand we have the traffic authority's degree of information. With the aid of a surveillance system it can stay updated about the present traffic conditions. We can also assume that the travel costs are calculated more or less correctly, because the traffic authority disposes of collected data for every link. But there can be errors of prediction of the future O-D demands. These influence the travel times calculated by the traffic authority and therefore also its strategy to choose, i.e. the green splits. We can assign a probability distribution to the degree of the demand prediction error.

5.3 The Bayes Game

To summarize we have the following types:

drivers	distribution of the different user classes θ_D
traffic authority	demand prediction error θ_P.

5.3.1 The Bayes-Cournot Game

The Bayes equilibrium can thus be expressed as follows:

The game profile $(\mathbf{g}^*, \mathbf{h}^*)$ and the type (θ_P^*, θ_D^*) is a Bayes equilibrium, if the following two equations hold:

$$\begin{array}{rcl} \mathbf{g}^* & = & \arg\max_{\mathbf{g}} \sum_{\theta_D} P(\theta_D) Z_{control}(\mathbf{g}, \mathbf{h}^*, \theta_P^*, \theta_D) \\ \mathbf{h}^* & = & \arg\max_{\mathbf{h}} \sum_{\theta_P} P(\theta_P) Z_{assignment}(\mathbf{h}, \mathbf{g}^*, \theta_D^*, \theta_P) \end{array} \tag{5.3}$$

Note that the traffic authority's type, the demand prediction error, has no influence on the drivers' payoff. We can therefore rewrite statement 5.3:

$$\begin{array}{rcl} \mathbf{g}^* & = & \arg\max_{\mathbf{g}} \sum_{\theta_D} P(\theta_D) Z_{control}(\mathbf{g}, \mathbf{h}^*, \theta_P^*, \theta_D) \\ \mathbf{h}^* & = & \arg\max_{\mathbf{h}} Z_{assignment}(\mathbf{h}, \mathbf{g}^*, \theta_D^*) \end{array} \tag{5.4}$$

This formulation corresponds more or less to the statement 2.65 about the equilibrium of the combined traffic control-assignment problem. Remember that this was at the same time the formulation for the Cournot equilibrium. Hence expression 5.4 is equivalent to the Cournot equilibrium, but this time with uncertainty.

5.3.2 The Bayes-Stackelberg Game

The Stackelberg expression is:

$$\begin{array}{rrcl} & \mathbf{g}^* & = & \arg\max_{\mathbf{g}} \sum_{\theta_D} P(\theta_D) Z_{control}(\mathbf{g}, \mathbf{h}^*, \theta_P^*, \theta_D) \\ \text{subject to} & \mathbf{h}^* & = & \arg\max_{\mathbf{h}} Z_{assignment}(\mathbf{h}, \mathbf{g}, \theta_D^*) \end{array} \tag{5.5}$$

As in the common Stackelberg game the traffic authority chooses the Cournot equilibrium with the highest payoff, but this time the payoff is calculated as the expected payoff over all possible user class distributions.

5.4 Discussion

With the perception and the demand prediction errors the most annoying aspects of the analytical model, the a priori known travel times, are removed. It is still to discuss, how to implement the demand prediction error. One possibility is to add a travel time error to the calculated travel times. A more natural, but more complex method is to modify the demand matrix by a stochastic perturbation. Note that in practical application the introduction of a random demand prediction error makes no sense. First the traffic authority assumes that its prediction is correct and second it is very doubtful that a superposition of a random error makes the model more exact. But the introduction of random errors gives us an idea of the consequences that perception and prediction errors can have.

The introduction of uncertainty might make the model more realistic, but it makes it also heavier.

The ideas of this chapter are developped by the author.

Conclusion

The results obtained by Chen (cf. [1]) show, that a game theoretic approach to the combined traffic control-assignment problem offers an astonishingly large improvement to current signal guidance systems. An improvement of more than 10% is achieved only by optimizing the green splits. It seems likely that an adaption of the phasing leads to further improvements.

The developped method is very promising, but the conversion to practice might meet some problems. The green splits are taken as continuous variables. This leads to a smoothed traffic flow, while in practice the alternation of green and red phase leads to a natural discretization. We have to deal with accumulations of traffic. This demands a coordination between the different traffic signals.

Furthermore an investigation about the users' behaviour will deliver more realistic datas. The models for the different user classes are still more or less empirical. The network users' lack of information and other aspects of uncertainty open even a bigger avenue of research, namely the identification of the traffic game with a game dealing with uncertainty as seen in chapter 5.

The running time of the proposed algorithm is, implemented in *MATLAB* (cf. [1]), slower than real time, using a reasonable time discretization (not too coarse). But finally the goal has to be real time application, modifying the demand models with respect to the current situation. So, another algorithm than the gradient based convex combination method is to study. This all the more because the algorithm may give us just a local instead of a global optimum if the feasible region is non-convex.

Bibliography

[1] Owen Jianwen CHEN, *Integration of Dynamic Control and Assignment*, Massachusetts, 1998

[2] Drew FUDENBERG and Jean TIROLE, *Game Theory*, Massachusetts, 1991.

[3] Anna NAGURNEY and Ding ZHANG, *Projected Dynamical Systems and Variational Inequalities with Applications*, Massachusetts, 1996.

[4] Yosef SHEFFI, *Urban Transportation Networks*, New Jersey, 1985.

[5] J. SZÉP and F. FORGÓ, *Introduction to the Theory of Games*, Budapest, 1985.

Appendix A

Dynamic Traffic Assignment Problem

We are going to show, that the dynamic traffic assignment problem is given as follows:

$$\begin{array}{rrcl} & \min Z & = & \int_0^T \sum_a \int_0^{u_a(t)} \tau_a(\omega) \quad d\omega dt \\ \text{s.t.} & \sum_p h_p^{rs}(t) & = & D^{rs}(t) \quad \forall t \\ & h_p^{rs}(t) & \geq & 0 \end{array} \tag{A.1}$$

We now apply the Lagrange multipliers method and develop the optimality constraints to see, that at the solution of the problem, the system is in a Wardrop equilibrium. The corresponding Lagrange function to A.1 is:

$$L(\mathbf{h}, \lambda) = \int_0^T \sum_a \int_0^{u_a(t)} \tau_a(\omega)\, d\omega\, dt + \sum_{rs} \lambda_{rs}(t)(D^{rs}(t) - \sum_p h_p^{rs}(t)) \tag{A.2}$$

The optimality constraints are:

$$h_p^{rs}(t) \frac{\partial L(\mathbf{h}, \lambda)}{\partial h_p^{rs}(t)} = 0 \tag{A.3}$$

$$\frac{\partial L(\mathbf{h}, \lambda)}{\partial h_p^{rs}(t)} \geq 0 \tag{A.4}$$

$$\frac{\partial L(\mathbf{h}, \lambda)}{\partial \lambda_{rs}(t)} = 0 \tag{A.5}$$

Calculating the different terms we get:

$$\frac{\partial L(\mathbf{h},\lambda)}{\partial h_p^{rs}(k)} = \sum_b \frac{\partial L(\mathbf{h},\lambda)}{\partial u_b(k+c_p^{rb}(k))} \frac{\partial u_b(k+c_p^{rb}(k))}{\partial h_p^{rs}(k)} \tag{A.6}$$

$$= \sum_b \frac{\partial\left(\int_0^T \sum_a \int_0^{u_a(t)} \tau_a(\omega)d\omega\, dt\right)}{\partial u_b(k+c_p^{rb}(k))} \frac{\partial u_b(k+c_p^{rb}(k))}{\partial h_p^{rs}(k)} - \lambda_{rs}(k) \tag{A.7}$$

$$= \int_0^T \sum_b \tau_b(k+c_p^{rb}(k))\delta_{b,p}^{rs}\, dt - \lambda_{rs}(k) \tag{A.8}$$

$$= \int_0^T c_p^{rs}(k)dt - \lambda_{rs}(k) \tag{A.9}$$

$$= T \cdot c_p^{rs}(k) - \lambda_{rs}(k) \tag{A.10}$$

So, the optimality constraints are:

$$h_p^{rs}(t)(T \cdot c_p^{rs}(t) - \lambda_{rs}(t)) = 0 \tag{A.11}$$

$$T \cdot c_p^{rs}(t) - \lambda_{rs}(t) \geq 0 \tag{A.12}$$

$$\sum_p h_p^{rs}(t) = D^{rs}(t) \tag{A.13}$$

But this is the definition of the Wardrop equilibrium, if we interprete $\lambda_{rs}(t)$ as the minimal path travel time for path p connecting r and s times T.

Note that in A.6 we used the chain rule and we derivate first with respect to all entry flows concerned by $h_p^{rs}(k)$. For the inner derivation in A.8 we used the relationship

$$u_b(t) = \sum_{rs,p} \sum_{\hat{t}+c_p^{rb}(\hat{t})=t} h_p^{rs}(\hat{t})\delta_{b,p}^{rs} \tag{A.14}$$

Finally, in A.9 we used 2.4.

The justification of the objective function of the traffic assignment problem for the dynamic case has been written down by the author.