Bachelor Thesis

Time slot allocation by auctions

Author:
Nathalie van Zeijl

Supervisors:
Dr. A.F. Gabor and Dr. Y. Zhang

July 13, 2011

Abstract
In this research we investigated what effects are of an auction have on congestion on the road. We used the VCG mechanism for the idea of rules for the auction and we introduce a IP formulation which can be used to calculate an optimal solution for the auction. We compared the IP formulation to other combinatorial optimization problems in literature and we discuss some of the characteristics of the solution of the IP formulation. We end with a real data example of the A15, which shows it is possible to divide the trucks of companies more over the day.

CONTENTS

1. INTRODUCTION

1.1 General Problem Description

Traffic jam is a common problem, especially in the Netherlands. The Dutch government tries hard to reduce the congestion on the Dutch highways. One of the problem areas is the A15 by Rotterdam, which is a busy route for freight. This A15 leads directly to the port of Rotterdam and every day many companies send their trucks over this route, besides the normal traffic. This gives a lot of hold-up for these transports as well as the other traffic. The public reaction which follows, is demanding a decrease in the number of trucks at the busy hours. For this reason the government wants to reduce the congestion on the A15. The government knows that congestion is due to the fact that the trucks are traveling too much on the same time. So the government is interested in methods to divide the driving trucks more evenly over the day.

For this research the government considers a method which makes companies willing to use the A15 for freight on other times than they do in the current situation. In order to achieve this willingness from the companies, the government chose to compensate the companies for transporting goods on other hours. A possible way to achieve this compensation is by organizing an auction for time slots. Every company that is interested to transport goods on the route can submit a bid. This bid contains information about which starting time slot, on which trajectory and how many trucks they want to send. This information is represented by the value of time (VOT), which is the value the company associates with the information of the bid. It is assumed that a bid of a company is in terms of VOT and thereby the information about the starting time slot, trajectory and number of trucks is known. The government wants to make sure that the capacity of the road is never violated and that every company gets their total number of desired trucks.

The goal of the research is to make a start with analyzing the impact of an auction for time slots on the traffic congestion. We will describe a model to find an optimal solution for the auction, test in which situations it is possible to use it and apply the model on some real based data. We will also link our model to models in literature for future heuristic development.

1.2 Literature Research

The government wants to organize an auction in order to achieve compensations for driving on different hours. The mechanism design that will be used for this auction is the Vickey-Clarke-Groves (VCG) mechanism, which comes from the field of game theory studies. A nice introduction to the field of VCG mechanisms can be found in Nissan (2007)[13]. For the auction in this research the VCG mechanism states that all companies have each a set of valuations, in terms of VOT, given an outcome of the auction. This means that for every bid, the companies have their own value. In Nissan (2007)[13] also is proven that the VCG mechanism is incentive compatible. Incentive compatible means that with an optimal solution, a player in a VCG mechanism is always best off when he tells the truth. In this research the players are the companies and so the companies are best off when they bid their truth VOTs for the bids they make. The VOT is part of the utility function of the company, which the companies want to maximize in the game. This utility function consists of the VOT given the outcome of the auction and an additional payment. This payment is defined by the Groves mechanism (Groves (1973) [7]) and the Clarke pivot rule (Clarke (1971)[4]) which will be explained in more detail in the next section. The Groves mechanism also States that the outcome of the auction is given by a function which maximizes the total valuation.

Nisan and Ronen (2007)[14] also discuss the general theory of the VCG mechanism. However they also state that when the optimization problems, which are used to calculate a optimal solution for the aution, become larger, computers are incapable of solving the problem within polynomial time. This means that the problem is ‘NP-hard’. NP-hard means that the computation time for finding the optimal solution is too long and so it is preferred to find algorithms which can quickly find another, but still good solution to the problem. However in the article of Nisan and Ronen (2007)[14] is also shown that when such an algorithm is used, which means the problem is not solved optimally, the VCG mechanisms are not necessary incentive compatible anymore. This gives a problem, because then it is not known what the real valuation of the companies are and thus is there no certainty about reliability of the solution to the auction. Therefore in the article they introduce a second chance mechanism which is feasible truthful and is a modification of the VCG-based mechanism.

There is done a research to an auction for reducing traffic congestion by Wada and Akamatsu (2010)[2]. They used the tradable network permit (TNP) scheme (by their part proposed by Akamatsu, Sato and Nguyen (2006)[1]) in combination with the VCG mechanism and the Vickrey payment (Vickrey (1961) [17]) to trade routes on different time slots. In their case several players of the auction can win a permission to use a certain part of the road on a certain time. This means that not every player is assigned to a time slot, which is contradictory to our research because in this research every player (company) needs to be assigned to some time slot. They also formulate an optimization problem for this problem. Furthermore they show that for only one origin-destination combination this mechanism works effectively but that otherwise the problem will be computationally hard; i.e., the optimization problem is 'NP-hard'. To avoid this NP-hard problem that they propose an mechanism which is readily implementable. This mechanism is called a day-to-day auction mechanism.

In the next chapter there will be a further explanation of what the VCG mechanism means for this research. In chapter 3 we will describe the model we will use, which maximizes the total valuation of the auction. In chapter 4 some theoretical examples of data are given, in order to investigate the characteristics of the solution to the auction and to get an idea about when the computation time gets too long. In chapter 5 there will be an example of data inspired on real data about the A15. Finally we will finish with a conclusion and recommendations.

2. THE VICKREY-CLARKE-GROVES AUCTION

The auction the government wants to organize to allocate the time slots, will be organized as a Vickrey-Clarke-Groves (VCG) auction. This VCG mechanism (Parkes (2001) [16] and Nissan (2007) [13]) comes from game theory studies and is based on the theories of the Vickrey auction (Vickrey (1961) [17]), the Clarke pivot rule (Clarke (1971) [4])and the Groves mechanism (Groves (1973) [7]). The Groves mechanism states that participants of the game (from now on called *players*) have a quasi-linear utility function for some set of possible alternatives, K. According to the Groves mechanism this utility function is given by (Groves (1973) [7]):

$$U_i(k, p_i, \theta_i) = v_i(k, \theta_i) - p_i \tag{2.1}$$

Here $U_i(k, p_i, \theta_i)$ stands for the utility of player i, given an alternative k, the payment p_i and the valuations of player i, θ_i. The utility of a player depends on the player's value for alternative k and on a payment from the player to the mechanism. This is denoted by $v_i(k, \theta_i)$ and p_i respectively. The payment is defined by the Groves mechanism as (Groves (1973) [7]):

$$p_i(\hat{\theta}) = h_i(\hat{\theta}_{-i}) - \sum_{j \neq i} v_j(k^*, \hat{\theta}_j) \tag{2.2}$$

In equation (2.2) $\hat{\theta}$ denotes the set of reported preferences by the players, which are not necessarily their truth preferences. Also k^* represents the alternative which maximizes the total reported value over all players. So the payment depends, given the set of reported preferences, on a function h_i and on the total valuation, given alternative k^*, minus the value of player i itself. The $\hat{\theta}_{-i}$ in the function h_i represents a function over all reported preferences except the one of player i. The function h_i is specified by the Clarke pivot rule. The Clarke pivot rule states the function as (Clarke (1971) [4]):

$$h_i(\hat{\theta}_{-i}) = \sum_{j \neq i} v_j(k^*_{-i}, \hat{\theta}_j) \tag{2.3}$$

This function means that the payment also depends on the best solution without player i. In equation (2.3) k^*_{-i} represents the alternative which maximizes the total reported value over all players except player i. With this pivot rule of Clarke the payment for the auction becomes:

$$p_i(\hat{\theta}) = \sum_{j \neq i} v_j(k^*_{-i}, \hat{\theta}_j) - \sum_{j \neq i} v_j(k^*, \hat{\theta}_j) \tag{2.4}$$

And now the utility function becomes (Parkes (2001) [16]):

$$U_i(k, p_i, \theta_i) = v_i(k, \theta_i) - \sum_{j \neq i} v_j(k^*_{-i}, \hat{\theta}_j) + \sum_{j \neq i} v_j(k^*, \hat{\theta}_j) \tag{2.5}$$

The Vickrey auction (Vickrey (1961) [17])is a sealed-bid and second-price auction. In this Vickrey auction the sealed-bids mean that the participants of the auction only have information about their own bids and no information about the others. Second-price auctions are auctions where the player with the highest bid wins, but pays the second-highest bid.

The VCG mechanism has some nice properties. It is allocatively-efficient which means that the total value over all players is maximized. For this mechanism individual-rationality holds, which means that the utility for player i for participation in a game is at least the value of the utility when not participate. The most important property for this research however is that the mechanism is incentive-compatible, which means that the players will truthfully report information about their preferences for a optimal solution. This incentive-compatible property does not depend on the function of h_i (Parkes (2001) [16]).

2.1 VCG for This Research

In this research the government wants to organize an auction for time slots. This auction should ensure that the times on which the trucks are driving, are spread more evenly over a the day. The companies in this research are the players of the auction. The government changes some parts of the VCG auction. First the government decided to compensate the companies for having to drive on other times than they like. To ensure this we will for simplicity set the function of h_i to zero, which will change the utility function of the companies change into:

$$U_i(k, p_i, \theta_i) = v_i(k, \theta_i) + \sum_{j \neq i} v_j(k, \hat{\theta}_j) \tag{2.6}$$

As can be seen now there will be a positive transaction from the mechanism (the government) to the players. We can make this change because the form of function h_i does not have any effect on the property that the mechanism is incentive-compatible (Parkes (2001) [16]).

Secondly the government wants that the auction gives a solution in which exactly the amount of trucks is given to every companies, which they desires to send onto the road. This has some consequences for the way to find the solution for the auction. This will be discussed in the next chapter.

The companies need to offer bids in order to participate to the auction. These bids will be in terms of Value of Time (VOT). This VOT is related to a number of things:

- The number of trucks
- The time slot on which the trucks are supposed to be starting.
- The origin and destination of the trucks. This will be referred to as the used *trajectory*.

These details with the corresponding VOT form a bid for a company. From now on there will be referred to a *group* of trucks as a bid with the above properties. A group of trucks also contains information about which company the bid is from.

A company should bid for more than only the time slots a company wants, otherwise the auction would make no sense. Further do the companies need to give information about the number trucks they want to send on the road in total, which from now will be referred to as the *possessed* number of trucks. The groups of trucks with corresponding VOT will be offered to the auction.

When the bids are placed, the government needs to find a way to allocate time slots to the companies by maximizing the total VOT. The model which is used for this is explained in detail in the next chapter.

Finally when the government found a solution, the companies will receive the information about which group has 'won' the auction. They also receive information about the payment the company gets for that group. In this research we will focus only on how to find the 'winning' groups, so not on the payment.

In the next chapter we will introduce the IP formulation, which will be used throughout this research to try to find an optimal solution for the auction.

3. IP FORMULATION

The VCG mechanism requires an optimal solution for the allocation problem as stated in the previous chapter. This solution should maximize total VOT for all bids. Integer programming can be used to develop a model which maximizes the total VOT and ensuring that the requirements of the government are not violated. The following formulation gives the objective function, which is the function which maximizes the total VOT, and the constraints which will make sure that the requirements given by the government will be satisfied. These requirements are that the capacity of the road is never violated and that every company gets the number of trucks they posses to send on a certain trajectory.

Objective function

$$max \sum_g \sum_s VOT_{g,s}\, y_{g,s}$$

subject to:

$$\sum_g \sum_s N_{g,s}\, b_{g,a}\, y_{g,s} = N_a \qquad \text{for all } a \tag{3.1}$$

$$\sum_g \sum_s P_{g,s}^{h,k}\, N_{g,s}\, I_{g,k}\, y_{g,s} \leq C_{h,k} \qquad \text{for all } h, k \tag{3.2}$$

$$y_{g,s} = \{0, 1\} \qquad \text{for all } g, s \tag{3.3}$$

This formulation uses five different indices. These indices represent the following sets:

Sets:

G	=	Groups of trucks
S	=	Time slots for departure
A	=	Companies
H	=	Observation times
K	=	Segments of the road

An element of some set will be denoted by the same, small letter. The set G contains for every bid a company makes the corresponding group of trucks as explained in the previous chapter. Further does the set H contain the times on which the road is observed. With these sets the parameters and variables of the formulation can be explained. Parameters stand for data which is assumed to be known and the variables are supposed to be calculated by the model.

Parameters:

$VOT_{g,s}$ = The valuation of a bid corresponding to a group trucks g, departing on time slot s, in terms of Value Of Time.

$N_{g,s}$ = Number of trucks in group g, departing on time slot s.

N_a = Number of truck that company a wants to send on the road, which is referred to as the number of trucks the company possesses.

$P^{h,k}_{g,s}$ = The percentage of trucks that will be on segment k on time h, if group g departs on time slot s.

$C_{h,k}$ = The capacity of segment k of the road on time h.

$$b_{g,a} = \begin{cases} 1 & \text{if group } g \text{ belongs to company } a. \\ 0 & \text{else} \end{cases}$$

$$I_{g,k} = \begin{cases} 1 & \text{if group } g \text{ has segment } k \text{ in their trajectory.} \\ 0 & \text{else} \end{cases}$$

Variables:

$$y_{g,s} = \begin{cases} 1 & \text{if the bid for group } g\text{, departing on time slot } s\text{, is selected by the auction.} \\ 0 & \text{else} \end{cases}$$

Constraints:

The first constraint (3.1) makes sure that each company exactly gets the number of trucks to send on the road as the number of trucks they posses. For every group is checked whether it is chosen by the auction and if it belongs to company a. This is done by multiplying the number of trucks in a group with the control parameters $b_{g,a}$ and the variable $y_{g,s}$.

The second constraint (3.2) is to prevent exceeding the capacity of every segment of the road, on every point of time. First it has to be checked whether a group is picked and whether it uses the particular segment k of the road. This is done by multiplying the the number of trucks in a group by the control parameter $I_{g,k}$ and the variable $y_{g,s}$. From an assigned group of trucks is known that a percentage $P_{g,s}^{h,k}$ will be at time h on segment k.

The third constraint (3.3) tells that y is a binary variable.

3.1 Relations with other Combinatorial Optimization Problems

The previous described integer programming problem is related to other well-known combinatorial optimization (CO) problems. In this section we will describe these CO problems and explain what the similarities and differences are with the IP formulation used in this research.

3.1.1 The Knapsack Problem

The IP formulation is strongly related to the knapsack problem. The knapsack problem is based on the idea of filling a knapsack with different objects. This knapsack has a certain capacity, which results in having to make choices regarding to which objects should be put into the knapsack. Every object has its own weight and profit where these choices are based on. The objective is to maximize the profit for the whole knapsack. The standard knapsack problem is formulated as follows (Dantzig (1957) [8] and Kellerer, Pferschy and Pisinger (2004) [11]):

$$\max \sum_{j=1}^{n} p_j x_j \tag{3.4}$$

subject to:

$$\sum_{j=1}^{n} w_j x_j \leq c \tag{3.5}$$

$$x_j \in \{0, 1\}, \qquad j = 1, \ldots, n \tag{3.6}$$

In equation (3.4) p_j stands for the profit of object j and x_j represents whether a object is chosen or not. x_j is 1 when object j is selected and 0 otherwise. In equation (3.5) w_j stands for the weight of object j.

Although our IP formulation is also based on maximizing the profit (in terms of VOT) and on the restriction of capacity (the capacity of the road), it still differs much from the standard knapsack formulation. First of all our IP formulation also has other, extra constraints. These constraints are to make sure that all companies get the desired number of trucks. Further do the capacities depend on the number of segments of the road and on the number of observation times. This means there are more than one capacity in our problem and so the IP formulation has not one, but more constraints about the capacity (specifically: $k * h$). For this reason we take look at some extended versions of the standard knapsack problem.

3.1.2 Multidimensional Knapsack Problem

The multidimensional knapsack problem (MDKP) is also based on the idea of filling the knapsack, but in this problem the knapsack has more dimensions. This can be interpreted as that for every object j a number of r_{ij} resource units are required. Each resource i having its own dimension in the knapsack. So when an object is chosen, more than one capacity should be taken into account. The problem was formulated by Weingartner and Ness (1967) [18] as follows:

$$\max \sum_{j=1}^{n} p_j x_j \tag{3.7}$$

subject to:

$$\sum_{j=1}^{n} r_{ij} x_j \leq c_i, \qquad i = 1, \ldots, m \tag{3.8}$$

$$x_j \in \{0, 1\}, \qquad j = 1, \ldots, n \tag{3.9}$$

In this equation (3.8) r_{ij} stands for the resource units i that are needed for object j. In our problem the objects j indicate the group of trucks, which means that about this group is known which trajectory it uses, the number of trucks that drive in the group and to which company the group belongs. The dimensions of the knapsack can be seen as the different segment of the road. When a group of trucks is selected, the trucks use a number of segments for the trajectory that is known for the group. So for a chosen group different dimensions of the knapsack are filled. The size of the object is the number of trucks that use the road segment, because the capacity is in terms of usage. When to the object is also added information about at which time slot the group the MDKP resembles the IP formulation which is used in this research. The last similar aspect of this MDKP formulation is that possibility of filling the different dimensions depends on all the capacities of the resources that are needed for an object j. This is similar to our problem because when a group of trucks is send on a segment, the group will also use another segment of the road. For example when the capacity is smaller for the segment which follows next, it is not possible to put more trucks on the current segment than that can be driven on the next segment.

The differences however are that the capacities of the knapsack do not depend on the observation times, which means for our problem that the trucks drive on several segments on the same time. This is not possible, but we can change the interpretation of the dimensions. The dimensions can also be indicated as the segments of the road at a certain time. Then the objects still remain the groups of trucks with a associated starting time. Then the only missing thing compared to our problem is the restriction about assigning the groups of trucks to the companies.

3.1.3 Multiple Knapsack Problem

The multiple knapsack problem (MKP), in stead of more dimensions in one knapsack, has i different knapsacks. The difference with the MDKP is that the objects are not split into different resources. It also does not matter into which knapsack the object is packed. This means that for an object, every knapsack can be chosen to pack the object, but the object can only be packed once. One of the first to describe the MKP were Hung and Fisk in 1978 [9]:

$$\max \sum_{i=1}^{m} \sum_{j=1}^{n} p_j x_{ij} \tag{3.10}$$

subject to:

$$\sum_{j=1}^{n} w_j x_{ij} \leq c_i, \qquad i = 1, \ldots, m \tag{3.11}$$

$$\sum_{i=1}^{m} x_{ij} \leq 1, \qquad j = 1, \ldots, n \tag{3.12}$$

$$x_{ij} \in \{0, 1\}, \qquad i = 1, \ldots, m,\ j = 1, \ldots, n \tag{3.13}$$

In the MKP the x_{ij} is 1 when object j is put into knapsack i and zero otherwise. Equation (3.11) shows that there are more knapsack and that when object j is put into knapsack i this give weight w_j to knapsack i. This problem has an extra constraint, which is given in equation (3.12). This constraint makes sure that every object j is maximal once in one of the knapsack. So it is not possible to put the objects more than once into a knapsack.

In our problem the objects can be seen as the groups of trucks and in this case the different knapsack can be seen as the different time slots. When it is seen like this, it means that when a group of trucks is set to start on time slot i it is packed in knapsack i. This knapsacks then still have capacities in terms of number of trucks. However this gives a problem. When the trucks are put into the knapsack corresponding to the starting time slot, they will not be in the knapsack corresponding to the next time slot, however the trucks are possibly still driving on the road. But this could be changed by changing equation (3.12), to make sure that a group of trucks can fit in more than one knapsack. This can be done by reformulation (3.12) as:

$$\sum_{i=1}^{m} P_{ij} x_{ij} \leq 1, \qquad j = 1, \ldots, n \tag{3.14}$$

In equation (3.14) P_{ij} is the probability that group j, for which is know the trajectory, the start time slot and a road segment, is still driving on time slot i. This probability sums up to one for every segment of the road. However even if we change equation 3.12 into equation 3.14 in the MKP problem, the capacity of the road segments is still not taken into account. This means that the formulation still has to be changed entirely.

When this MKP is seen with the knapsacks as the different segments of the road and the objects are the group of truck with an associated starting time slot, then according to the MKP formulation the groups of trucks can only be send onto one segment. The capacity again is the number of trucks on the road. However the trucks do not only use one segment, so also in this interpretation equation (3.12) can be changed so that it possible to use more than one segment of the road. This can be done as:

$$x_{ij} \leq I_{ij}, \qquad j = 1, \ldots, n \tag{3.15}$$

In equation 3.15 the I_{ij} is 1 when segment i is used by group j an zero when not. This gives that the groups can drive over more than one segment and only on the segments they use. However also in this case this still means it is not possible to keep in track of the trucks in terms of time which is one of main things about the auction. So after all this formulation does not look very useful, a lot needs to be changed.

3.1.4 Multiple-Choice Multidimensional Knapsack Problem

Hifi, Michrafi and Sbihi (2004) [8] discuss a model which is a combination of the above two problems. It is called the multiple-choice multidimensional knapsack problem (MMKP). This problem can be formulated as follows. There are given n classes J_i of items. Every class has r_i items. Every object j of class i has a profit of p_{ij}. For object j of class i, k resources are needed which means that the weight of every resource becomes w_{ij}^k. Again the resource k should be put in dimension k of the knapsack. Every dimension of the knapsack has its own capacity which gives the capacities are denoted by c^k. The formulation is given to be:

$$\max \sum_{i=1}^{n} \sum_{j=1}^{r_i} p_{ij} x_{ij} \tag{3.16}$$

subject to:

$$\sum_{i=1}^{n} \sum_{j=1}^{r_i} w_{ij}^k x_{ij} \leq c^k, \qquad k = 1, \ldots, m \tag{3.17}$$

$$\sum_{j=1}^{r_i} x_{ij} \leq 1, \qquad i = 1, \ldots, n \tag{3.18}$$

$$x_{ij} \in \{0, 1\}, \qquad i = 1, \ldots, n,\ j = 1, \ldots, r_i \tag{3.19}$$

Equation (3.16) makes sure that the capacities of every knapsack dimension is not violated and equation (3.17) states that only one object can be chosen for every class.

In this MMKP formulation the objects are interpreted as the groups of trucks, the classes interpreted as the starting time slots and the resources are seen as the segments of the road. x_{ij} represents then whether group of trucks j is chosen for starting time slot i. The equation (3.16) makes sure that the capacity of the segments is not violated. However there is also a capacity on the time, so when we assume that the observation are equal to the starting time we could change c^k into c_i^k.

Another is thing is that this formulation states that only one object can be chosen for every class. This means that only one group of trucks can drive at a certain time slot and this is not the case. However by changing the capacities into c_i^k there is already covered for how many group can start at a certain time slot. If you would sum over all time slots, it is known how many groups are selected. With this there can be made a restriction for the number of trucks are needed for the company. However this means that the problem changed completely.

After discussing these CO problems it seems that the most important difference with our problem is the fact that there are extra constrains for assigning the groups of trucks to the companies.

In the next chapter we will investigate some characteristic of the IP formulation as given in this chapter.

4. CHARACTERISTICS OF THE SOLUTION

In this section we will characterize the solution of the discussed model for the time slot allocation (from now on we will refer to this as the IP solution). First of all we are interested in the fairness of the solution. In particular towards the small companies. This is interesting for future heuristic development, because this can then be taken into account. In the first section this is tested using some data examples and by using one-way lay out ANOVA.

Secondly it is interesting to see what happens with the IP solution when the capacity halfway or at the end of a trajectory decreases. This capacity decrease can be interpreted as a road for example going from a three lane road to a two lane road, which is the case in our real based data at the end of the road. Besides this straight forward interpretation, the capacity decrease can also be seen as effects of weather conditions. Different weather conditions are also very common and so we like to investigate the effects of capacity decreases halfway or at the end of a trajectory on the solution of the model.

Furthermore we are interested in the influence of the the size of the data on the running time in the computer program. This program is AIMMS and is a software designed for solving optimization problems. The number of constraints and the number of variables are used as a measures for this influence in the last section. The number of variables depends, as can be seen in the previous chapter, on the number of groups and the number of time slots. The number of groups it self depends on the number of companies, the number of trajectories and the number of time slots. The number of constraints depend on the number of companies, the number of road segments and the number of observation times. In the last section those two measures are used to see how large both can become to still make sure the model is solved within reasonable time.

4.1 Data Sets

All sections use example data sets. In the data sets considered, there will be two types of companies, small and large. The difference in the size of the company is made by the number of trucks the company posses. The specific number of trucks will differ in all sections.

The data is set to be the same for all possible situations, in order to get a clear idea about changes in the IP solution. For this there is only one

Tab. 4.1: Fixed numbers

# Road segments	**# Time slots**	**# Observation times**	**Max VOT per truck**
10	5	5	100

trajectory which is used, which contains all road segments. This is in order to have no side effects due to the difference in used trajectories between the groups. This similarity is also in the preferences of the companies. The companies are set to all prefer the same time slots. This means that the data is set such that the first time slot is favored most, the second time slot secondly and so on. By all companies the time slots are preferred by the same value. There is only one data part that is different for every situation: the percentages $P_{g,s}^{h,k}$. These are generated randomly in order to break the symmetry of the solution, however it will stay the same in all data examples. Table 4.1 presents the data that are set to be fixed. With these numbers the preferences per time slots per truck can be given in terms of VOT and they are presented in Table 4.2.

Tab. 4.2: Valuation of time slots per truck in terms of VOT

Time slot 1	**Time slot 2**	**Time slot 3**	**Time slot 4**	**Time slot 5**
100	90	80	70	60

A last feature that is important for the whole chapter is the capacity. In order to get a useful IP solution, the capacity can not be fixed for all different situation. A certain capacity can have three different consequences for the IP solution. It can either be that the capacity is that large that all companies can have their first choice and thus will drive all on the same time (because they all prefer the same time slot most). Or it could be that a certain capacity is too small and results in no possible solution. This means the problem is infeasible. As last it can be that the capacity results in a IP solution, which is not the trivial one. This kind of IP solution is the only one which is useful to say anything about the characteristics of the IP solution, otherwise there will be nothing to say. The more companies there will be in the data examples, the more trucks there will be on the road, the higher the capacity will have to be set in order to get a feasible solution other then the trivial. By this should be noted that in in all realistic situation the capacities are a given fact and we can now only change the capacity because we make our own data examples.

4.2 Fairness of the Solution

In this section we are interested in the fairness of the IP solution towards the small companies. It might be that the large companies are favored over the small companies. The model maximizes the total VOT and therefor it might be that higher bids are favored over smaller bids. When the difference between the bids becomes larger, bids from small companies might become unimportant because bids of large companies give much higher values. To test whether this is the case we will use some small data examples.

The small data example consist the following data:

- One large company.
- Two small companies.
- The small company possesses 10 trucks.
- The number of trucks that the large company possesses will be changing from 50 to 500.
- For every company, the number of trucks in the bid will be the maximum number of trucks and half of that maximum number for every time slot.
- The capacity will be the same for all road segments.

To test whether the bids of the small companies become unimportant we investigate what the effect is of increasing the difference in bid between the two types of companies. When the number of trucks of the large company (this from now on will be called the *size*) will change this will also cause a change in the size of the bids of the large companies. The bids for all companies are set to the maximum number of trucks they can bid for, and half of that maximum number. This is in order to have a structured bid system. The capacity is the set to be the same to make sure there are no side effect due to the capacity differences.

Now the first step is to see how the IP solution reacts on an increase in the size of the large company. The results are given in Table 4.3.

In Table 4.3 the first column gives the number of the runs. In the second column the size of the large company is presented, as can be seen it is set to be increasing. In the third column the capacities are given, which are used for every run. These capacities are as low as possible and are the first two that lead to a feasible solution (recall from the beginning of the chapter). The next column gives the total VOT of the solution in euros. Then the fifth column gives the calculation time it took AIMMS to solve it

Tab. 4.3: Data set 1

	Size large company	Capacity of road	Solution	Time (sec)	VOT S1(%)	VOT S2(%)	VOT L1(%)	AC S1	AC S2	AC L1
1	50	60	€6,950	0.20	95	100	100	1.5	1.0	1.0
2	50	50	€6,150	0.22	75	90	90	3.5	2.0	2.0
3	70	80	€8,950	0.19	85	100	100	2.5	1.0	1.0
4	70	70	€8,450	0.25	85	95	95	2.5	1.5	1.5
5	100	100	€11,500	0.17	100	100	95	1.0	1.0	1.5
6	100	90	€10,650	0.25	85	80	90	2.5	3.0	2.0
7	200	180	€21,000	0.17	100	100	95	1.0	1.0	1.5
8	200	170	€19,600	0.19	90	70	90	2.0	4.0	2.0
9	500	430	€49,500	0.19	100	100	95	1.0	1.0	1.5
10	500	420	€26,800	0.19	90	90	90	2.0	2.0	2.0

in seconds. The next three columns present the percentage VOT the solution gives of the maximal possible VOT per company. S1 and S2 represent the small companies and L1 the large company. The last three columns give the average choice (AC) per company, which gives an idea of how far the solution is from the first choice for each company.

The results of the runs in Table 4.3 have some notable properties. First, the difference in capacity leads to differences in the solution. By this we mean that when the size of the large company remains the same and only the capacity decreases, the IP solution also decreases. This seems reasonable because when the capacity decreases the trucks will have to be more divided over all hours and so the total VOT will decrease. Secondly, the calculation times are very short, this fact is promising because this is a very small example.

Another thing to note is what difference the percentages $P_{g,s}^{h,k}$ make for the solution. When we would not have these percentages the value of the optimal solution could be higher. For example look at run 5 in Table 4.3. The large company does not get the best possibility unlike the small companies, who do get the best possibility. However the capacity of the road is even to the size of the large company. So when we would have no percentages and use the greedy algorithm we would get that the large company gets the best possibility (which is the first time slot) and the two small companies get the second best bid (which is the second time slot). This would give a IP solution of €11,800. This is a larger solution than the IP solution in Table 4.3. But due to the percentages this is not possible because the capacity of the second time slot will be violated.

To test whether the small companies are in a disadvantage against the large company we use the ANOVA one-way lay-out, which tests whether the mean of the observations in columns nine till eleven in Table 4.3 do not differ significantly.

- H0: $\mu_{S1} = \mu_{S2} = \mu_{L1}$
- Ha: at least one of the means differ.

The results are in Table 4.4.

Tab. 4.4: ANOVA one-way lay out for differences between companies

Test Statistic	**Degrees of Freedom**	**Significance Level**	**Critical Value**	**Conclusion**
0.48	2, 27	0.05	3.35	not reject H0

From Table 4.4 it follows that the null Hypothesis is not rejected which means there is no significant difference in the results of Table 4.3. By this it should be noted that the number of observations is small for this test, therefor it could be that the test is not completely correct. However a good reason for not rejecting H0 could be that when the difference between the size of the companies becomes larger also the differences between the bids become larger. By this is meant that, with taking the bids as maximum and half of that when the size increases from 50 to 200 the bids increase from 25 and 50 to 100 and 200. Keeping in mind the fact, as mentioned before, that if the capacity is low in comparison with the number of trucks the problem may be infeasible. Or if the capacity is too high the solution is trivial since everyone gets the desired time slot. This might result in such a large capacity that it is always possible to fit in the trucks of the small companies. This is a nice result for future heuristic development but we like to investigate what happens when the bids do become closer to each other.

In order to test with less difference between the bids we use the same data with as only difference the number of bids. The bids that are used in the next runs all begin with the bid of ten trucks and then increase every bid with 10 when possible. This means that the small companies have for every time slot only the bid of ten trucks and the large company has bids from ten, twenty, up to the maximum number of trucks. We call this increase in bids of ten trucks an *bid step* of ten trucks. The results are in Table 4.5.

The columns in Table 4.5 represent the same as in Table 4.3. Again we test whether there is no difference between the results in Table 4.5, columns nine till eleven.

- H0: $\mu_{S1} = \mu_{S2} = \mu_{L1}$
- Ha: at least one of the means differ.

The results are presented in Table 4.6.

Tab. 4.5: Data set 2

	Size large company	Capacity of road	Solution	Time (sec)	VOT S1(%)	VOT S2(%)	VOT L1(%)	AC S1	AC S2	AC L1
11	50	60	€7,000	0.17	100	100	100	1.0	1.0	1.0
12	50	50	€6,600	0.23	95	100	94	2.0	1.0	1.6
13	70	70	€8,900	0.33	100	95	100	1.0	2.0	1.0
14	70	60	€8,300	0.27	100	100	90	1.0	1.0	2.0
15	100	90	€12,000	0.19	100	100	100	1.0	1.0	1.0
16	100	80	€11,700	0.19	100	100	97	1.0	1.0	1.5
17	200	140	€20,500	0.79	100	60	94.5	1.0	5.0	1.6
18	200	130	€17,400	2.34	90	100	77.5	2.0	1.0	2.2
19	500	290	€47,100	14.32	80	100	90.6	3.0	1.0	1.9
20	500	280	€43,700	12.06	100	80	93.8	1.0	3.0	2.6
21	500	280	€44,600	14.67	100	100	85.2	1.0	1.0	2.6

Tab. 4.6: ANOVA one-way lay out for differences between companies

Test Statistic	Degrees of Freedom	Significance Level	Critical Value	Conclusion
7.49	2, 30	0.05	3.32	reject H0

From Table 4.6 it follows that the null hypothesis is rejected and that there is a difference between the companies. The results in Table 4.5 however suggest that the large company is not significantly better off. This is not a bad result because we wanted to know whether the small companies are in a disadvantage and this is not the case. By this test it should also be noted that the number of observations is small, and it therefor could be that the test is not completely correct.

It should be noted that in Table 4.5 there are eleven runs, which means that an extra run is added. This run is exactly the same as the one before (so number 20) the only thing that differs is the way the data is put into the program AIMMS. All runs that are done before had the data matrices sorted in the way that first all data from the first small company, next from the second small company and last all data from the large company was in it. It sometimes looks like the two small companies were favored in the program because they were in the top of the data matrices. This could be the case because of the Branch and Bound algorithm the program is partly build on. When the start position is always on the top of the data matrix it might be the first data is favored by the Branch and Bound algorithm. To investigate this the matrix was sorted differently. First all first choices were put into the matrices and after that all second choices and so on. This means for this data example that first all bids for the first time slot are filled in, followed by all bids for the second time slot and so on. Table 4.5 shows that the program gives a different solution. So it might be that the Branch and Bound issue is influencing the solution, but it is also due to the fact that the percentages ($P_{g,s}^{h,k}$) did change in order since the data was generated randomly. For similar phenomena see the article of Jans and Desrosiers (2011) [10], but we will not investigate this subject any further and the data will be sorted in the way we used to order it before.

4.3 Influence of Road Capacity

In this section it is investigated what the influence is on the optimal solution when there is a decrease in the capacity at the end of the road. An example of a capacity decrease is the A15, which at the end of the road has a decrease in capacity. Recall from the introduction that a decrease in capacity can also be seen as the consequences of the weather conditions. It is possible that when the capacity at the end of the trajectory decreases that it is no longer possible to find a feasible solution.

The example data we will use for this consist the following numbers:

- 50 small companies.
- 10 large companies.
- The small companies possesses 50 trucks.
- The large companies possesses 200 trucks.
- The bid size (recall from previous section) is 50.
- The capacity is the same for every segment of the road, except for the last one. The capacity of the last segment will be changing.

In order to see what the influence of the capacity change is, the other data are kept the same. In Table 4.7 the results are shown. The program is run with different road capacities and it is stopped after half an hour running time. The first column represents the number of the corresponding run. The second column contains the capacity of all segment without the last segment. The column 'Decrease' shows the percentage of which the capacity of the last road segment decreases. The next column presents the IP solution for that run. If the solution in NO, this means that no solution is possible or no solution is found within the half hour. The fifth column gives the gap of the solution. Recall that when the program is running half an hour, we stop it. When the program has found a solution this is not necessarily the optimal solution. The gap represents the distance between the solution and the Linear Programming solution in percentages. This should give an idea how close the solution is to the optimal solution. The last column gives the average number of time slots the solution assigns to the large companies.

Only of the large company because the small companies can only be assigned one time slot due to the fact that the bid step is the number of trucks the small companies possesses.

Tab. 4.7: testing effect of capacity decrease

	Capacity	**Decrease (%)**	**Solution**	**Time (sec)**	**Gap**	**Average # time slots**
1	2,500	0	€450,000	3.14	0	1
2	2,500	5	€449,500	0.69	0	1,1
3	2,500	10	€449,000	1.04	0	1
4	2,500	15	€446,500	1.22	0	1,2
5	2,500	20	€442,000	1.78	0	1,2
6	2,500	25	€435,000	6.71	0	1,3
7	2,500	30	€424,500	43.49	0	1,4
8	2,500	35	€407,500	66.28	0	1,5
9	2,500	40	€371,000	130.37	0	1,7
10	2,500	45	NO	0.39		
11	2,400	0	€447,500	1.54	0	1,2
12	2,400	5	€447,000	4.82	0	1,3
13	2,400	10	€445,000	2.92	0	1,2
14	2,400	15	€442,000	7.21	0	1,2
15	2,400	20	€436,000	45.75	0	1,4
16	2,400	25	€426,500	500.65	0	1,4
17	2,400	30	€413,000	89.67	0	1,3
18	2,400	35	€387,000	1800.08	0.14	1,5
19	2,400	40	NO	0.36		
20	2,300	0	€442,000	20.81	0	1,4
21	2,300	5	€441,000	19.58	0	1,5
22	2,300	10	€438,000	311.69	0	1,3
23	2,300	15	€433,000	695	0	1,5
24	2,300	20	€424,500	1,800	0.23	1,3
25	2,300	25	€412,000	1,800	0.43	1,4
26	2,300	30	NO	1,800		
27	2,300	35	NO	0.59		
28	2,200	0	NO	1,800		

Table 4.7 starts with a run with no decrease so it is possible to analyze the effect of decreasing capacity. In run 1 we get to the maximum solution possible. When decreasing the capacity of the last segment, the table shows that the solution also decreases. An important thing to notice is that the larger the capacity starts, the easier it is to solve the problem when the end capacity decreases. This is what is expected. When the start capacity is getting smaller it takes longer calculation times. Table 4.7 shows that when the end capacity decreases with the percentages in the third column, it will come to the point where the data no longer has a feasible solution. It can also be seen that when the start capacity decreases, there are less possibilities to

decrease the end capacity. This is what we expected, however even by the smallest start capacity it is possible to decrease the end capacity with 25%. Another important thing to note is that when the end capacity decreases the average number of time slots that are assigned to the companies increases. For furthers heuristic development this is a result that should be taken into account.

So we can conclude that when decreasing the end capacity of the road, this influences the solution. It might be possible that the solution is no longer feasible. Also can be concluded that the calculation times and the average number of time slots assigned to the companies increase when the end capacity decreases.

4.4 Solvability with AIMMS

In this last section we are interested in the solvability with AIMMS. For this research the program AIMMS version 3.11 is used with an AMD® Athlon™ 64 x2 Dual-Core processor with 2GB DDR2. Because it is known that the problem is NP-Hard, it would be nice to see with how many variables and constraints AIMMS is able to calculate a solution within reasonable time. And so with how many variables and constraints is AIMMS not able to do this. Reasonable calculation time is set to be half an hour. Recall also from the beginning of this chapter, that we will use two measures: the number of variables and the of constraints of the problem. These numbers will change depending on the data. The number of variables depends on the number of groups and the number of time slots. The number of groups itself depends on the number of companies, the number of trajectories, the number of time slots and the bid step size. The number of constraints depend on the number of companies, the number of road segments and the number of observation times. The number of constraints and the number of variables are easy to calculate:

$$\#\,\text{Variables} = \#\,g\,*\,\#\,s \tag{4.1}$$

$$\#\,g = \#\,a\,*\,\#\,\text{trajectories}\,*\,\#\,s\,*\,(\text{company size/bid step}) \tag{4.2}$$

$$\#\,\text{Constraints} = \#\,a\,*\,\#\,k\,*\,\#\,h \tag{4.3}$$

In equations (4.1) till (4.3) the g stand for groups, s for departure time slots, a for companies, k for road segments and h for observation times. # means 'number of', so # g means the number of groups.

The example data which will be used contains the following data:

- The small companies possesses 50 trucks.

- The large companies possesses 200 trucks.
- The capacity is the same for every segment of the road.

The number of companies will be changing in order to change the number of variables and the number of constraints. The bid step size will also be changed, which changes only the variables. Recall that *bid steps* are the step within the bids of a companies. So if the bid step is ten, a small company bids for 10, 20, 30, 40 and 50 trucks for every time slot.

In Table 4.8 the results of the tests are proposed. The first column gives the corresponding run number. The second and third column denote the number of companies that are used in that run. Followed by the number of constraints and the number of variables. The sixth column presents the bid step and the seventh column the capacity of the road. Again recall that the capacity changes to get an useful solution. In the last three columns present the solution, the running time and the gap of the solution.

Tab. 4.8: Testing solvability with AIMMS

	# Small companies	**# Large companies**	**# Constraints**	**# Variables**	**Bid step**	**Capacity**	**Solution**	**Time (sec)**	**Gap**
1	2	1	53	300	25	180	€25,000	0.6	0
2	2	1	53	750	10	170	€27,400	3.3	0
3	50	10	110	1,250	50	2,200	€442,000	19.9	0
4	50	10	110	4,500	25	2,300	€449,250	30.2	0
5	50	10	110	11,250	10	2,150	€450,000	977.4	0
6	50	20	120	2,250	50	3,300	€644,500	201.8	0
7	50	20	120	6,500	25	3,300	€650,000	56.0	0
8	50	20	120	16,250	10	3,100	€650,000	57.9	0
9	50	30	130	3,250	50	4,250	€841,500	1,800	0
10	50	30	130	8,500	25	4,200	€849,250	1,800	0.09
11	70	10	130	1,350	50	2,900	€545,000	11.1	0
12	70	10	130	5,500	25	2,800	€548,000	77.6	0
13	70	20	140	2,350	50	3,800	€741,000	392.7	0
14	70	20	140	7,500	25	3,800	€750,000	153	0
15	70	30	150	3,350	50	4,800	€941,000	1,800	0.15
16	70	30	150	9,500	25	4,700	€947,000	1,800	0.32
17	100	20	170	2,500	50	4,500	€887,500	594.1	0
18	100	20	170	9,000	25	4,500	€895,000	1,800	0.14
19	100	40	190	4,500	50	6,400	€1,284,000	1,800	0.20
20	100	40	190	13,000	25	6,400	€1,295,250	1,800	0.33
21	100	60	210	6,500	50	8,300	€1,674,500	1,800	0.64
22	100	60	210	17,000	25	8,250	€1,698,000	1,800	0.12

As mentioned before, it can be seen in Table 4.8 that the size of the companies and the bid step differ. By this, the number of constraints and variables also differ. This is shown in the first six columns. The capacities are set to the smallest capacity where the model can find an optimal solution. When there is a gap, it is not possible to find an optimal solution within half an hour. There are four runs which have the trivial answer as optimal solution. Runs 5, 7, 8, 14 have the maximal solution possible for that problem. This means for run 7 that when the capacity is set smaller, within half an hour we get an answer but with a gap. For the other three is this not the case. For runs 5, 8, 14 is it is not possible to get another solution (even with a gap) than the trivial one in half an hour. This is also the reason why the calculation time of run 14 is shorter than that of run 13. Run 13 can get to a solution with a gap but this takes half an hour and run 14 will give no answer for a smaller capacity after half an hour.

We started with a bid step of ten and twenty-five, but because run 5 and 8 show that with that number of variables it is not possible to get an solution other than the trivial, the bid step is increased.

After knowing this, we can seen that runs 12, 13 and 17 are the runs with the highest number of constraints/ variables which can still be solved within half an hour (and not having the trivial solution). So we can state that when the number of constraints are 130 or less the problem can be solved within half an hour with a maximal number of variables around 5000. When the number of constraints are more than 130 it is save to say that the maximal number of variables is around 2000 to solve the problem. There needs to be noted that when one of the two measures decreases the other can increase. However we can say that the number of constraints have more impact on the solution than the number variables. When the number of constraints increase by 10 the effect on the solution is the same as a increase in number of variables in thousands.

In the next chapter we will describe and run a real based data example to see whether auctions do have any effect on the spreading of driving trucks.

5. REALITY BASED DATA

We started this paper with the problem of the traffic congestion on the A15. We will now apply the described model on the data which is based on the data of that A15. First we will present and explain the known data for the A15. This data will need to be transformed in order to use it with the model. This will be explained in the second section and the results will be presented in the last section.

5.1 Available Data

Fig. 5.1: Picture of A15

The available data is about the A15 from Hoogvliet (exit 17) to Haven (exit 15). Figure 5.1 is a picture of this part of the A15. From this part of the road the following facts are known:

- The traffic demand is known on 8 time slots over 7 road segments.
- 20% of this traffic demand is freight traffic.
- The 8 time slots are divided over the time interval: 6.00am till 8.00am which means there is every 15 minutes a new time slot.
- There are 130 small, 62 medium and 1 large companies that use this road on the 8 time slots.

- The capacity of the road segment is known and stays the same over time.
- The probabilities that the trucks will be on a certain segment on a certain time, given all origin, destination and departure time slot combinations.

The demand is given in a three-dimensional matrix, with the demand for all origin, destination, departure time slot combinations. The percentages are given in a two-dimensional matrix with in the first column the origin, in the second the destination, next the departure time slot, then the number of the road segment, next the observation time and last the percentage.

As mentioned the available data is information about the demand from 6.00am till 8.00am with intervals of fifteen minutes. The investigated part of A15 is split into parts of 500 meters and this gives seven road segments. An graphical representation of the road is given in Figure 5.2.

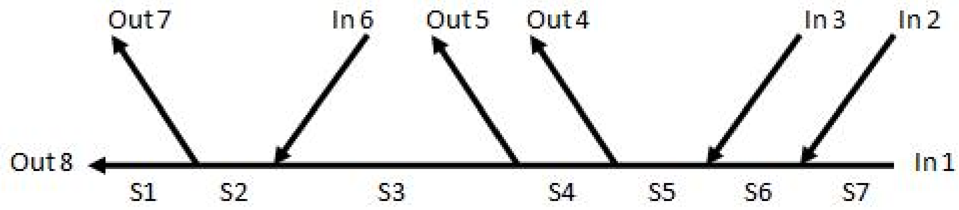

Fig. 5.2: Graphical representation of A15

From Figure 5.2 it follows that the road has four on-ramps, called in 1, in 2, in 3 and in 6, and four off-ramps, called out 4, out 5, out 7, out 8. The on-ramps are defined as possible *origins* and the off-ramps as possible *destinations*. The given probabilities represent the probability that the trucks are on one of the seven segment on one of the eight time slots, given that a group of trucks start at one of the four origins on one of the eight time slots, with one of the off-ramps as destination. These probabilities will be seen as percentages. The distribution of the demand for the eight time slots is given in Figure 5.3. As can be seen there is a peak in the demand around 6.00am and 6.15am. The goal for this data is to flatten peak of the demand within these eight time slots.

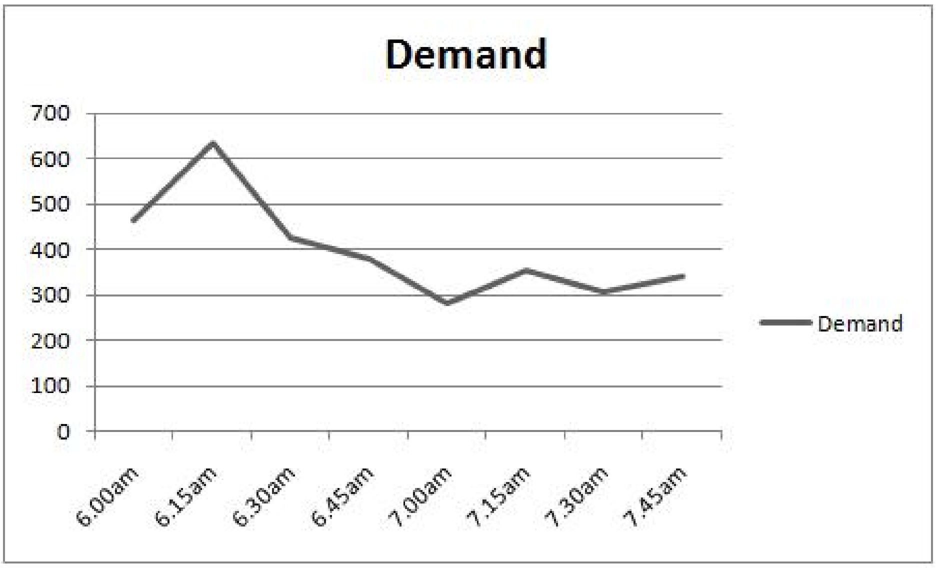

Fig. 5.3: Distribution of demand current situation

5.2 Data Transformation

In order to make it possible to use the data for the IP formulation it needs to be transformed. First the percentages are transformed. We got a two-dimensional matrix which needs to become a four-dimensional matrix for our program. We begin with changing the numbering of the road segment, so that segment 1 is after origin 1. This is done to get no confusing later. After this it is made into a five-dimensional matrix, with the first five columns used as coördinates. In Appendix A the used Matlab function is given. The data is first transformed to a five-dimensional matrix because in the matrix are given the origin and destination. In our IP formulation we only need to know the trajectories they use and so this will be changed later. In this data case we know that there are fourteen possible trajectories, with a known origin and destination. With this we can transform the five-dimensional matrix into a four-dimensional matrix by changing the origin, destination combinations to a numbers 1 till 14 for every trajectory. The Matlab function is given in Appendix B. This matrix is for only one group so for every group in the data, this matrix should be added to the matrix which will be used. This final matrix can be read into AIMMS.

The demand data is used in a different way. It is used to determine the number of trucks a company possesses for every type of company. It is assumed that the demand is distributed over the companies as given in Table 5.1:

The number of trucks per company is calculated simply by taking the per-

Tab. 5.1: Demand distribution

Size company	**Number of companies**	**Total percentage of demand**
Small	130	40%
Medium	62	50%
Large	1	10%

centage of the total demand, dividing it by the number of companies and then rounding off. The result of this are in Table 5.2

Tab. 5.2: Number of trucks per company

Size company	**Number of companies**	**Number of trucks per company**
Small	130	10
Medium	62	25
Large	1	320

In order to develop the bids for every company, a bid step needs to be determined. The bid step for this data will be different for every type of company, because otherwise it would not be possible to get all data into AIMMS. For the small and medium companies the bid step is the number of trucks they possesses, which means that they only bid the maximal number of trucks. However the bids are for every time slot and every trajectory. The bid step for the large company is 40 and also all bids are for every time slot and every trajectory.

The demand data is also used to determine a ranking for the time slots and the trajectories. This is needed because the bids for every company have to be made and there need to be different VOTs for different bid. In order to know how to value different bid we chose to use the demand data to determine a ranking. The demand is summed over all trajectories and then the ranking is based on highest total demand for the time slots. So the time slot with the highest demand has the highest rank and so on. For the ranking of trajectories the same procedure is used. The demand is summed over all time slots and the trajectory with the highest demand gets the highest rank and so on.

After ranking the departure time slots and the used trajectories we had to set a value to all different ranks. The VOT per truck per rank for the trajectories are given in Table 5.3.

Tab. 5.3: VOT per truck per rank for time slots

Rank	1	2	3	4	5	6	7	8	9	10	11	12	13	14
VOT per truck	100	97.5	95	92.5	90	87.5	85	82.5	80	77.5	75	72.5	70	67.5

The VOT as given in Table 5.3 are given to every bid. So now for every time slots the values are the same. So that needs to be changes. We decided to given a penalty to every time slot which is not the best and every lower ranks should give a higher penalty. The penalties per truck per rank for the time slots are given in Table 5.4.

Tab. 5.4: Penalty per truck per rank for time slots

Rank	1	2	3	4	5	6	7	8
VOT per truck	0	-5	-10	-15	-20	-25	-30	-35

When comparing Table 5.3 and 5.4 it can be seen that a difference between the ranks for the trajectories are smaller than the difference between the ranks of the time slots. This is because we do not want punish to hard for different trajectories so these can differ more but we do want to punish for different time slots. We want the possibility for more difference within the trajectories because companies might have to drive certain routes, however we have no information what so ever about this so we cannot make use of it.

In the next section we will present the results of the IP solution.

5.3 Results

The results of the auction are given in Figure 5.4.

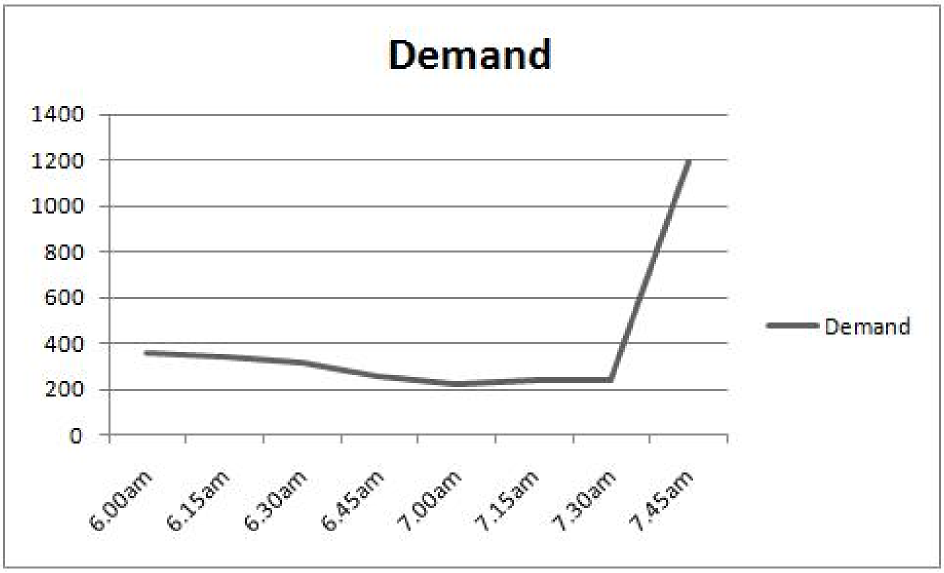

Fig. 5.4: Distribution of demand after the auction

The first thing that falls into attention from Figure 5.4 is that at the end od the graph is a peak. The reason for this is that the groups can start on any time slot, so also on the eighth time slot, and that after time slot eight the road has no capacity any more. This results in a very large number of groups starting at the last time slot. Besides this we can see that the peak around 6.15am is gone. This means the the program works for all time slot except for the last one.

Although the number of variables are 1,792,200 and the number of constraints are 10,808, we were able to find a solution within 300 seconds. The total value of time of this solution is €311,202.50 with only a gap of 0.02%. This has a few reasons. First of all many of the percentages we used were zero, where in the example data every percentage was a random number, so there where far more possibilities. Besides that in order to get every thing into AIMMS the small and medium companies where only able to bid for the total number of trucks they posses, which also means that the number of possible solutions decrease. As last is there no difference between the companies, so it does not matter which company gets a less favored bid. This means that there are many solutions which give the same optimal solution and so AIMMS only has to find one. This does not really matter because there is no information about differences between companies and if there were it would be implemented and then there would be no same possible solution.

6. CONCLUSION AND RECOMMENDATIONS

In this research we were interested in the question what effects an auction have on congestion on the road. We discussed the VCG auction for this research and we introduced a IP formulation which would calculate an optimal solution for the auction. However we know that the problem is NP-hard so we compared our IP formulation with known combinatorial optimization problem for literature. This might be useful for future heuristic development for a other then optimal, but good solution. In order to understand the solution of our IP formulation better we described a number of characteristic in chapter 4. We found that small companies are not is a disadvantage against large companies. Also that when the capacity at the end of the road decreases the solution might become infeasible, but certainly has longer calculation times and the average number of time slots that are assigned to the the companies increases. The solvability with AIMMS is tested and it followed that it is when using under 130 constraints and 5000 variables it is possible to get to a solution within half an hour. When we tried the IP formulation on the real based data we found that the VCG mechanism is possible to divide the trucks of companies more even over the day. However it should be taken into account that road capacities do not stop at the last starting time slot. We would recommend to invent some heuristic for the IP formulation because the problem is NP-hard and in reality there much more data than used in this research.

BIBLIOGRAPHY

[1] Akamatsu, T., Nguyen,L. X. and Sato, S. (2006). Tradable time-of-day bottleneck permits for morning commuters. *JSCE Journal of Infrastructure Planning and Management* 62 605620.

[2] Akamatsu, T. and Wada, K. (2010). An auction mechanism for implementing tradable network permit markets: An intellegent transportation system for eliminating traffic congesture on road networks.

[3] Burkard, R.E., Dell'Amico, M., Martello, S. (2008). Chapter 1. *Assignment Problems*. USA: Siam.

[4] Clarke, E.H. (1971). Multipart pricing of public goods. *Public Choice*, 11, 1733.

[5] Dantzig, G.B. (1957). Discrete-Variable Extremum Problems. *Operations Research* Vol. 5, No.2 (Apr., 1957), pp. 266-277

[6] Dawande, M., et all (2000). Approximation Algorithms for the Multiple Knapsack Problem with Assignment Restrictions. *Journal of Combinatorial Optimization*, 4, 171-186.

[7] Groves, T. (1973). Incentives in Teams. *Econometrica*, Vol. 41, No. 4, p. 617-631.

[8] Hifi, M., Michrafi, M., Sbihi, A. (2004). Heuristic algorithms for the multiple-choice multidimensional knapsack problem. *Journal of the Operational Research Society* 55, 1323-1332.

[9] Hung, M.S., Fisk, J.C. (1978). An algorithm for 0-1 multiple-knapsack problems. *Naval Research Logistics Quarterly*, Vol. 25, No. 3, p. 571-579.

[10] Jans, R., Desrosiers, J. (2011). Efficient symmetry-breaking formulations for the job grouping problem. *Journée de la recherche.*

[11] Kellerer, H., Pferschy, U., Pisinger, D. (2004). *Knapsack Problems.* Berlin: Springer-Verlag.

[12] Martello, S., Toth, P. (1980). Solution of the zero-one multiple knapsack problem. *European Journal of Operational Research*, Vol. 4, No. 4, p. 276-283.

[13] Nisan, N. (2007). Introduction to Mechanism Design (for computer Scientists). *Algorithmic Game Theory*

[14] Nisan, N. and Ronen, A (2007). Computationally feasible VCG Mechanisms. *Journal of Artificial Intelligence Research* 29 1947.

[15] Osborne, M.J. (2004). *An introduction to Game Theory.* New York: Oxford university press.

[16] Parkes, D. (2001). Classic Mechanism Design. Chapter 2, *Iterative Combinatorial Auctions: Achieving Economic and Computational Efficiency* Ph.D. dissertation, Univesity of Pennsylvania.

[17] Vickey, W. (1961). Counterspeculation, Auctions, and Competitive Sealed Tenders. *The Journal of Finance*, Vol. 16, No. 1, p. 8-37.

[18] Weingartner, H.M., Ness, D.N. (1967). Methods for the Solution of the Multidimensional 0/1 Knapsack Problem. *Operations Research*, Vol. 15, No. 1, p. 83-103.

[19] Wolsey, L.A. (1998). *Integer Programming.* New York: Wiley interscience publication.

Appendix A

MATLAB FUNCTION 1: TRANSFORMING 2D MATRIX INTO 5D MATRIX

```
function P = RealP(Pnieuw)

P = zeros(4,4,8,7,8);

for n = 1:size(Pnieuw,1)
    P(Pnieuw(n,1),Pnieuw(n,2),Pnieuw(n,3) ...
        ,Pnieuw(n,4),Pnieuw(n,5))= Pnieuw(n,6);
end
```

In this function the two-dimensional data matrix with the probabilities are used as input. In this matrix the segment numbering is already changed.

Appendix B

MATLAB FUNCTION 2: TRANSFORMING 5D MATRIX INTO 4D MATRIX

```
function Pdef = PdefMaken(P,g)

Pdeff(1,:,:,:) = P(1,4,:,:,:);
Pdeff(2,:,:,:) = P(2,4,:,:,:);
Pdeff(3,:,:,:) = P(3,4,:,:,:);
Pdeff(4,:,:,:) = P(1,5,:,:,:);
Pdeff(5,:,:,:) = P(2,5,:,:,:);
Pdeff(6,:,:,:) = P(3,5,:,:,:);
Pdeff(7,:,:,:) = P(1,7,:,:,:);
Pdeff(8,:,:,:) = P(2,7,:,:,:);
Pdeff(9,:,:,:) = P(3,7,:,:,:);
Pdeff(10,:,:,:) = P(1,8,:,:,:);
Pdeff(11,:,:,:) = P(2,8,:,:,:);
Pdeff(12,:,:,:) = P(3,8,:,:,:);
Pdeff(13,:,:,:) = P(6,7,:,:,:);
Pdeff(14,:,:,:) = P(6,8,:,:,:);

Pdef = Pdeff;
for i = 1:(g/14)-1
    Pdef = [Pdef; Pdeff];
end
```

In this function the input is the output of the previous function and the number of groups. It transforms the five-dimensional matrix, based on the fourteen trajectories we have, into a four-dimensional matrix which only contains information about which segment of the road is used.