Development of a tool to combine rides with time frames efficiently while respecting customer satisfaction

van Zutphen, K.J.H.

Award date:
2014
Development of a tool to combine rides with time frames efficiently while respecting customer satisfaction.

By
K.J.H. (Kevin) van Zutphen

BSc Industrial Engineering – TU/e 2012
Student identity number: 0634442

In partial fulfillment of the requirements for the degree of

Master of Science
In Operations Management and Logistics

Supervisors:
Dr. Ir. H.P.G. van Ooijen, TU/e OPAC
Prof. Dr. T. van Woensel, TU/e OPAC
R. van Strien, Tinker
Subject Headings: airport, goods transportation, linear programming, logistics, passenger transportation, taxi
Abstract

This Master Thesis project investigates the ability to combine taxi rides at Tinker, a fast-growing company who uses an online booking system for customers who reserve taxi rides in advance to transport them to airports or from airports back to home. This project researches opportunities to transport multiple passengers in one ride and opportunities to reduce empty transport kilometers to reduce the total transportation.

The current situation of Tinker has been analyzed to find out to what extent combining rides can be profitable. A tool has been developed to allocate passengers to a taxi ride in an efficient way. The effect of changing the current setting in order to increase the possibilities has been investigated as well and recommendations will be given for further improvements.
Management Summary

In this Master Thesis Project is investigated how Tinker, a company who uses an online bookings system to schedule taxi rides to airports, could gain money by combining rides. They believe that by transporting two customers together in one taxi reduction in costs will be reached. Two types of combining rides are considered: transporting customers in one taxi to the airport by combining their bookings into one ride, defined as Combination and planning rides in such a way that after bringing customers to the airport, other customers are brought back home with the same taxi, which is defined as Matching. The following problem was formulated:

Develop a methodology to determine whether a (taxi) company using an online internet booking system for customers to reserve rides in advance, characterized by (1) time windows, (2) multiple types of transport options, (3) various capacity constraints, using (4) multi depots, can generate more profit by combining bookings, still adhering to the performance agreed upon.

After investigating the dataset, it is concluded that combination and matching are possibilities to reduce costs. The bookings are to the airport or from the airport back home, only bookings with the same direction can be combined. For matching, only rides with a different direction can be matched.

Most bookings only have limited number of passengers and customers, meaning that it is possible to plan a different booking in the same taxi. In addition, enough bookings arrive on or depart from the airport at similar times, meaning that combination will not result in many additional travel times.

Tinker lets customers choose to allow combinations or not, to keep the service open for customers who are absolutely not willing to have their rides combined. In addition, they promise to customers that they only have an maximum additional travel time (defined by $m_i$) of 0.25 hours, or a maximum additional travel fraction of 20% (denoted by $f_i$) of the original travel time. These promises limit the number of possible combinations, but do not affect the number of possible Matches since in that case customers do not travel together.

First, the effect of these parameter settings has been investigated. In addition, the effect of introducing an allowed customer waiting time ($\xi_i$) is investigated. When it is allowed to combine bookings with other bookings which should for example arrive 0.5 hours later, the number of possible combinations is increased. The value of $\xi_i$ is related to customers, but for the Matching process, a value $\eta$, which is defined as the maximum allowed additional waiting time of the vehicle driver, is introduced to make it possible to match a To ride with a Back ride which departs later.

Combination reduces kilometers by saving kilometers with customers in the taxi. Instead of transporting customer A to destination 1 and customer B to destination 2, first both customers are picked up and then transported to their destinations (which could be the same). In addition, empty kilometers (with no customers in the taxi, but required to get to the customer / destination or back) are spared, since only one taxi need to travel to/back from one of the customers and destinations. Matching does not influence kilometers with customers in the taxi, but reduces empty kilometers since it is not necessary to drive empty to or back from the destinations.

The parameters $m_i, f_i, \xi_i$ and $\eta$ have a large effect. Enabling a 0.5 hours waiting time almost doubles the number of possible combinations. Dropping the fraction triples the number of
possibilities. The maximum additional travel time only has effect in combination with increasing the fraction. Also the value of $\eta$ has a big influence, allowing a 0.5 hours waiting time for the vehicle increases the number of possible matches by 50%. It can be concluded that making these parameters more flexible will result in higher savings.

Combination and Matching have to be executed separately, since they depend on each other. If two rides are combined, only one taxi arrives at the location instead of two, resulting in limited possibilities to match rides. This also works the other way around. Combination is considered first since the expected savings per combination are higher as the expected savings per match.

Two models are built:

1. A simple Excel model, which can easily be used, but is limited since bookings are only compared to limited other bookings, not resulting in the optimal solution. In addition, the Excel models cannot deal with large values of $\eta$;
2. A Linear Programming model, which will result in the optimal schedule and can deal with all combinations of parameters, but requires software and knowledge to use. This model has long calculation times when many bookings are added in the model.

The total savings have been investigated for three settings for the Excel model:

1. The original situation;
2. A flexible situation which allows 0.5 hours waiting time, an additional travel time of 0.5 hours and has a fraction of 1;
3. A mixed setting in which the fraction is only considered for rides that take more than 0.25 hours, with a maximum additional travel time of 0.25 hours for rides taking less than 1.25 hours and a maximum additional travel time of 0.5 hours for rides taking longer than 1.25 hours, not allowing additional waiting time.

The results for the Excel model are given for a value of $\eta = 0$ (situation 1, matching To and Back rides with similar time), a situation in which To rides are also compared with Back rides departing 0.5 hours later and a situation in which also To rides minus 0.5 hours are compared with Back rides. This is based on a 6 month period, with a total distance of 1.155.843 kilometers when no bookings are combined or matched. Values of $m$ and $\xi$ are given in hours.

<table>
<thead>
<tr>
<th>$(f, m_1, (m_2), \xi)$</th>
<th>Scenario</th>
<th>Combination savings in km</th>
<th>Matching savings in km</th>
<th>Total savings (incl. Combination)</th>
<th>Total savings fraction</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0.2; 0.25; 0)</td>
<td>1</td>
<td>12.689</td>
<td>44.307</td>
<td>56.996</td>
<td>4.9%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>12.689</td>
<td>64.397</td>
<td>77.086</td>
<td>6.7%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>12.689</td>
<td>50.902</td>
<td>63.591</td>
<td>5.5%</td>
</tr>
<tr>
<td>(1; 0.5; 0.5)</td>
<td>1</td>
<td>76.982</td>
<td>39.619</td>
<td>116.601</td>
<td>10.1%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>76.982</td>
<td>59.627</td>
<td>136.609</td>
<td>11.8%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>76.982</td>
<td>55.814</td>
<td>132.796</td>
<td>11.5%</td>
</tr>
<tr>
<td>(0.2; 0.25; 0.5; 0)</td>
<td>1</td>
<td>36.245</td>
<td>43.633</td>
<td>79.878</td>
<td>6.9%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>36.245</td>
<td>63.457</td>
<td>99.702</td>
<td>8.6%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>36.245</td>
<td>49.405</td>
<td>85.650</td>
<td>7.4%</td>
</tr>
</tbody>
</table>
In the current situation, the savings are almost 5%. This is an interesting value, but it can be increased by using different parameter settings. For the LP problem, which has been implemented in AIMMS, these scenarios are tested as well. The differences between the models are as follows, considering data of December 2013 (171.862 kilometers in original situation without combination and matching):

<table>
<thead>
<tr>
<th>Setting</th>
<th>Model</th>
<th>Value of $\eta$</th>
<th>Combination To Rides</th>
<th>Combination Back Rides</th>
<th>Matching</th>
<th>Total Savings</th>
<th>Fraction</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(0.2; 0.25; 0)$</td>
<td>Excel</td>
<td>0</td>
<td>1.683</td>
<td>584</td>
<td>4.551</td>
<td>6.818</td>
<td>4.0%</td>
</tr>
<tr>
<td></td>
<td>LP</td>
<td>0</td>
<td>1.683</td>
<td>584</td>
<td>6.751</td>
<td>9.018</td>
<td>5.2%</td>
</tr>
<tr>
<td></td>
<td>LP</td>
<td>0.5</td>
<td>1.683</td>
<td>584</td>
<td>9.440</td>
<td>11.707</td>
<td>6.8%</td>
</tr>
<tr>
<td></td>
<td>LP</td>
<td>1</td>
<td>1.683</td>
<td>584</td>
<td>12.436</td>
<td>14.703</td>
<td>8.6%</td>
</tr>
<tr>
<td>$(0.2; 0.25; 0.5; 0)$</td>
<td>Excel</td>
<td>0</td>
<td>4.523</td>
<td>1.260</td>
<td>4.384</td>
<td>10.167</td>
<td>5.9%</td>
</tr>
<tr>
<td></td>
<td>LP</td>
<td>0</td>
<td>4.325</td>
<td>1.260</td>
<td>6.604</td>
<td>12.189</td>
<td>7.1%</td>
</tr>
<tr>
<td></td>
<td>LP</td>
<td>0.5</td>
<td>4.325</td>
<td>1.260</td>
<td>9.615</td>
<td>15.200</td>
<td>8.8%</td>
</tr>
<tr>
<td></td>
<td>LP</td>
<td>1</td>
<td>4.325</td>
<td>1.260</td>
<td>12.416</td>
<td>18.001</td>
<td>10.5%</td>
</tr>
<tr>
<td>$(1; 0.5; 0.5)$</td>
<td>Excel</td>
<td>0</td>
<td>8.578</td>
<td>2.956</td>
<td>3.805</td>
<td>15.339</td>
<td>8.9%</td>
</tr>
<tr>
<td></td>
<td>LP</td>
<td>0</td>
<td>10.017</td>
<td>2.984</td>
<td>6.406</td>
<td>19.407</td>
<td>11.3%</td>
</tr>
<tr>
<td></td>
<td>LP</td>
<td>0.5</td>
<td>10.017</td>
<td>2.984</td>
<td>10.954</td>
<td>23.955</td>
<td>13.9%</td>
</tr>
<tr>
<td></td>
<td>LP</td>
<td>1</td>
<td>10.017</td>
<td>2.984</td>
<td>12.601</td>
<td>25.602</td>
<td>14.9%</td>
</tr>
</tbody>
</table>

Based on this, the differences between the models in combination are small, but for more flexible parameter values, the difference increases. For matching, the Linear Programming model performs about 30% better as the simple Excel model. Only the values for $\eta = 0$ can be compared, since the calculations are different for $\eta = 0.5$ and the simple Excel model cannot deal with $\eta = 1$ since the Excel model can only compare a limited number of rides.

The results for $\eta = 1$ (allowing that vehicle drivers wait 1 hour at the airport) increase the savings a lot, therefore LP is recommended. If Tinker does not intend to use these values and accepts a solution close to the optimal, the Excel model can be enough, however if they have the expertise to implement the LP model, they should do that since this will result in the best solution.
Preface

In this report results of my graduation project to receive the master’s degree for the master Operational Management and Logistics in Industrial Engineering at the Eindhoven University of Technology are presented. This project was executed from February 2014 to July 2014 at Tinker in Schiphol, Amsterdam.

This project provided me an opportunity to apply the knowledge learned in several courses during my bachelor Industrial Engineering and master Operational Management & Logistics at the University of Technology in Eindhoven in reality. It was very educative to execute this project at a very interesting new company who has many opportunities to grow even more in the future. As the company was very new, many new developments happened during my project and I am sure that Tinker will become even more successful if they continue to operate based on their viewpoint. I would like to thank my supervisor Rob van Strien who find time during his daily busy life to help me with the project and the meetings we had offered me many insights which helped me with the project.

I also want to thank my supervisors from Eindhoven, Henny van Ooijen and Tom van Woensel. Both were always available to help me when I needed it and offered important feedback which helped me to complete the project.

I also want to thank everyone who helped me during my study, my group members during the study, friends, family and everyone else involved. It was a time which I will not forget.
<table>
<thead>
<tr>
<th>List of concepts</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Back ride</td>
<td>Situation in which customers are transported from airport to their destination.</td>
</tr>
<tr>
<td>Booking</td>
<td>A single online booking, made by a group or a single passenger.</td>
</tr>
<tr>
<td>Booking time</td>
<td>Time in which the customer registers a booking (not the time they travel).</td>
</tr>
<tr>
<td>Booking time in advance</td>
<td>Date of travel minus date of booking.</td>
</tr>
<tr>
<td>Comfort Class</td>
<td>Situation in which the customer does not allow combined rides.</td>
</tr>
<tr>
<td>Customer distance</td>
<td>Total travel distance with customers in the taxi.</td>
</tr>
<tr>
<td>Drive</td>
<td>Executing one (combined) ride To the destination and travel back to the depot afterwards, execute one (combined) Back ride starting at the depot or doing both a To and Back ride directly after each other. Moreover, all kilometers from the depot till the taxi returns to the depot.</td>
</tr>
<tr>
<td>Economic Class</td>
<td>Situation in which the customer allows combined rides.</td>
</tr>
<tr>
<td>First customer</td>
<td>Customer in a combined ride who is picked up as second in a To ride, and brought back home first in a Back ride. This customer might not have any disadvantage from a combined ride; it could only be the case that the customer has to wait for the second customer at the airport.</td>
</tr>
<tr>
<td>LP</td>
<td>Linear Programming</td>
</tr>
<tr>
<td>LU</td>
<td>Luggage Units</td>
</tr>
<tr>
<td>Matching</td>
<td>Executing a Back ride (almost) directly after a To ride with one taxi.</td>
</tr>
<tr>
<td>PDP</td>
<td>Pickup and Delivery Problem</td>
</tr>
<tr>
<td>Ride</td>
<td>Executing one booking To the destination or Back, or combining two bookings To or back.</td>
</tr>
<tr>
<td>Second customer</td>
<td>Customer in a combined ride who is picked up as first in a To ride, and brought back home second in a Back ride. This customer faces the disadvantages of a combined ride for sure.</td>
</tr>
<tr>
<td>Taxi distance</td>
<td>Travel distance with no customers in taxi (time to get from taxi depot to location to pick up customer).</td>
</tr>
<tr>
<td>TB</td>
<td>Taxi Bus</td>
</tr>
<tr>
<td>TC</td>
<td>Taxi Car</td>
</tr>
<tr>
<td>To ride</td>
<td>Situation in which customers are transported from their destination to the airport</td>
</tr>
<tr>
<td>Travel time</td>
<td>The time in which the customer travels to the airport, or back to home.</td>
</tr>
<tr>
<td>VRP</td>
<td>Vehicle Routing Problem</td>
</tr>
</tbody>
</table>
# Table of Contents

1. Introduction.......................................................................................................................... 1
   1.1. The research assignment.................................................................................................. 1
   1.2. The Approach .................................................................................................................. 1
   1.3. Structure of the report...................................................................................................... 2
2. The company of the case study.............................................................................................. 3
   2.1. The history of the company ............................................................................................ 3
   2.2. The process...................................................................................................................... 3
      2.2.1. Customer viewpoint................................................................................................. 4
      2.2.2. Tinker viewpoint....................................................................................................... 5
   2.3. Characteristics of the company....................................................................................... 5
3. Description of the research assignment.................................................................................. 8
   3.1. Literature research recap.................................................................................................. 8
   3.2. Linking literature study with characteristics of Tinker .................................................. 9
   3.3. The general assignment................................................................................................ 10
      3.3.1. Increasing the number of combinations and matches .............................................. 12
      3.3.2. Combining and Matching rides efficiently ............................................................... 12
      3.3.3. Combining the aspects............................................................................................ 13
4. Analysis.................................................................................................................................. 14
   4.1. The current price strategy............................................................................................... 14
   4.2. Dataset of Tinker............................................................................................................ 15
      4.2.1. Number of passengers............................................................................................. 15
      4.2.2. Luggage Units......................................................................................................... 16
      4.2.3. The booking time in advance.................................................................................. 16
      4.2.4. Travelling from or to the airport .......................................................................... 16
      4.2.5. Distance to the airport .......................................................................................... 17
      4.2.6. Class type............................................................................................................... 17
      4.2.7. Information about carriers..................................................................................... 18
      4.2.8. Speed of the taxi..................................................................................................... 19
      4.2.9. Other observations................................................................................................. 19
   4.3. Flight data....................................................................................................................... 19
   4.4. Summary of the analysis............................................................................................... 20
Appendices
References

5. Model formulation .............................................................................................................. 21
   5.1. Model description ..................................................................................................... 21
   5.2. Effect of maximum times ....................................................................................... 23
       5.2.1. Calculating the distances ............................................................................... 24
       5.2.2. Constraints ..................................................................................................... 25
       5.2.3. The Models ..................................................................................................... 27
           5.2.3.1. Combination model to optimize parameters ............................................. 27
           5.2.3.2. Matching model to optimize parameters ................................................. 28
   5.3. Selecting the most optimal drives ............................................................................ 28
       5.3.1. Calculating the savings .................................................................................... 29
       5.3.1.2. Matching savings ....................................................................................... 30
       5.3.2. The savings optimization model ..................................................................... 31
   5.4. The linear programming model ................................................................................ 33
       5.4.1. Combination problem ..................................................................................... 33
       5.4.2. Matching problem ......................................................................................... 35
   5.5. Output of the models ............................................................................................... 37
       5.5.1. Output of the combination model .................................................................... 37
       5.5.2. Output of the matching model ........................................................................ 38
   6. Case Study ................................................................................................................... 39
       6.1. Input data of Tinker ............................................................................................. 39
       6.2. Changing the allowed travel and waiting times .................................................. 40
       6.3. Savings estimations ............................................................................................. 43
           6.3.1. Combining Bookings ................................................................................... 43
           6.3.2. Matching ....................................................................................................... 44
       6.4. Linear Programming model results ........................................................................ 46
   7. Implementation ............................................................................................................. 48
   8. Conclusions and recommendations ............................................................................. 49
       8.1. Conclusions Research Questions / Assignment ................................................... 49
       8.2. Recommendations to Tinker ................................................................................. 50
References .......................................................................................................................... 51
Appendices ........................................................................................................................ 52
   Appendix 1: Cost structure analysis .............................................................................. 53
Appendix 2: Data characteristics of Tinker

Appendix 3: Schiphol flights fraction per time

Appendix 4: Allocation Example
   Appendix 4a: Location/destination method
   Appendix 4b: Travel date and time
   Appendix 4c: Number of luggage and passengers
   Appendix 4d: Moment of booking
   Appendix 4e: Conclusions

Appendix 5: Formulas Maximum Additional Travel Time

Appendix 6: Savings
   Appendix 6a: Combination Savings for each setting
   Appendix 6b: Matching Savings for each setting

Appendix 7: Excel models
   Appendix 7a: Model to find how many bookings can be combined
   Appendix 7b: Model to find how many rides can be matched
   Appendix 7c: Combination model to calculate savings
   Appendix 7d: Matching model to calculate savings

Appendix 8: Dutch law for taxi drivers

Appendix 9: Models applied for Tinker situation

Appendix 10: Savings analysis of simple Excel model

Appendix 11: Savings per Combination / Match of LP model

Appendix 12: Results of savings calculations for Excel model

Appendix 13: Speed formula determination
Table of Tables

Table 1: Transportation Problem Heuristics ................................................................. 8
Table 2: Luggage volumes grouped per number of pax ................................................. 16
Table 3: Flights per session in 2013 ............................................................................... 19
Table 4: Trade-off between combination methods ....................................................... 23
Table 5: Effect of allowing waiting time ....................................................................... 40
Table 6: Effect of increasing maximum absolute waiting time and fraction ............... 41
Table 7: Combining Probabilities for different settings ................................................ 42
Table 8: Effect of \( \eta \) .................................................................................................. 43
Table 9: Savings fraction per Setting, first Economic Class only and secondly all rides ... 43
Table 10: Estimation of Results of Matching Rides ......................................................... 45
Table 11: Savings in the Linear Programming Model compared to the Simple Excel Model ...... 46
Table 12: Cost Calculations ......................................................................................... 53
Table 13: Fractions Class grouped per time and direction ............................................ 60
Table 14: Bookings Allocation Example ........................................................................ 65
Table 15: Distances of Bookings Allocation Example .................................................... 65
Table 16: Bookings Allocation Example sorted on capacity .......................................... 66
Table 17: Bookings Allocation Example sorting on booking on advance time .............. 66
Table 18: Results Bookings Allocation Example ............................................................ 67
Table 19: Additional travel time constraints per setting ................................................. 68
Table 20: Combination Savings per setting ................................................................... 69
Table 21: empty kilometers per setting ......................................................................... 69
Table 22: Combinations grouped in kilometer slots for the original situation .............. 77
Table 23: Combinations grouped in kilometer slots for the flexible situation ................ 78
Table 24: Distance effect on vehicle distances for matching for both settings .............. 79
Table 25: Average savings for Combination / Matching for LP model .......................... 80
Table 26: Combination probabilities for various settings ............................................. 81
Table 27: Combinations in mixed setting ..................................................................... 82
Table of Figures

Figure 1: Luggage screen .......................................................... 6
Figure 2: Taxi Car and Bus ........................................................ 6
Figure 3: Transportation steps To ride ....................................... 10
Figure 4: Transportation steps Back ride ................................... 10
Figure 5: Matching drive .......................................................... 11
Figure 6: Scatter plot of customer locations ............................... 17
Figure 7: Graphical distribution carriers ................................... 18
Figure 8: Cost effect of number of passengers .......................... 55
Figure 9: Cost effect of increasing luggage ............................... 55
Figure 10: Cost effect of booking days in advance ..................... 56
Figure 11: Cost effect of distance .............................................. 56
Figure 12: Number of people travelling per booking ................. 57
Figure 13: Luggage quantities per booking ............................... 57
Figure 14: Bookings per day in advance over long term ............. 58
Figure 15: Bookings per day in advance (0-10 days) ................ 58
Figure 16: Bookings grouped by distance, given in kilometers .... 58
Figure 17: Percentage Economic Class grouped per number of passengers and LU .... 59
Figure 18: Effect of distance on Economic Class fraction .......... 59
Figure 19: Effect of Time on Economic Class fraction ............... 60
Figure 20: Effect of Time on Economic Class, grouped in To and Back rides .......... 60
Figure 21: Effect of distance on average speed ......................... 61
Figure 22: Speed and distance related to time ......................... 62
Figure 23: Flight pattern Schiphol in hours in percentages .......... 63
Figure 24: Demand Tinker resulting from Departures (rides to the airport) ........ 64
Figure 25: Demand Tinker resulting from Landings (Back rides, rides to the customer) .... 64
Figure 26: Speed Formulas ...................................................... 83
1. Introduction

This chapter describes the project. The research assignment will be described in the first section and in section 1.2 the approach to investigate the assignment will be worked out. The third and final section of this chapter describes the structure of the report.

1.1. The research assignment

The assignment will be done for Tinker. Tinker is a company which uses an online booking system at which people can book a taxi to travel to or back from airports like Schiphol. People are picked up at a selected location and they travel directly to the airport, or the other way around. However, this transportation process can be improved by combining bookings. Customers who are located close to each other or are on each other’s route and travel almost at the same time could travel together in one taxi, reducing the total transportation costs. Tinker believes that combining bookings will lead to a more profitable schedule. They do not use combination yet, but since the number of taxi bookings that they have each day is growing fast, they believe that combining bookings is an attractive possibility to make their process more efficient. The following research assignment has been defined.

Develop a methodology to determine whether a (taxi) company using an online internet booking system for customers to reserve rides in advance, characterized by (1) time windows, (2) multiple types of transport options, (3) various capacity constraints, using (4) multi depots, can generate more profit by combining bookings, still adhering to the performance agreed upon.

The tool must be able to combine bookings in a profitable way, without dropping customer satisfaction. The customers must accept this new development, since they are needed to make the combinations possible. The person responsible for the schedule must be able to use it easily.

1.2. The Approach

This section will describe the most important steps taken in this project to get to the results. These steps are based on the literature research and research proposal which are made in advance of this project.

In the first step, information about the current process is retrieved. In order to find out how the current process of the company can be improved it must be clear how the company currently operates. Limitations of the current process will be addressed and possible improvements are discussed. Current data is analyzed and discussed. This data is derived using SQL in the database of Tinker. Additional information was gathered by scheduling meetings with the company supervisor and using sources from internet. For example, information about flight patterns of Schiphol was derived from the web.

Based on this data, the characteristics of the company are worked out. Chapter 4 will summarize this data, and these characteristics will be taken into account in the mathematical models which are presented in chapter 5. These models will be tested on the situation of Tinker.
1.3. **Structure of the report**
The report continues with an introduction of the company in chapter 2. The history of the company, the current booking process and characteristics related to the research assignment will be described. Chapter 3 describes the research assignment. The main assignment and additional sub assignments are given. Chapter 4 is about the analysis of the company. Data of the company and related information is summarized in this chapter.
In chapter 5 (mathematical) models related to the research assignments given in chapter 3 are described. These models are built for general situations and are not limited to the situation of Tinker. Chapter 6 links Tinker with these models, in this section the results are discussed. Chapter 7 is about the implementation of the models. Chapter 8 concludes this report, recommendations will be given and ideas for further research are discussed.
2. The company of the case study
This chapter will describe the general characteristics of the company that gave the inspiration for this problem. In section 2.1, the history of the company will be described. Section 2.2 continues with the current booking process. Finally, section 2.3 provides insights in the characteristics of the company related close to the research assignment.

2.1. The history of the company
Tinker was founded in 2012. At 28 November 2012, the company started to use an online booking system to offer taxi transport from a customer location to Schiphol Airport or the other way around (Tinker: Smarter in Travel, 2013). Tinker does not own taxis but uses the current taxi network to offer their service.

The idea of Tinker started by thinking about phone lines. It is possible to call from home directly to other people since there are phone lines at your house, but if this was not the case; people should visit the main office of their provider to call. When people fly this is the case; they must travel to the airplane to catch their flight. Tinker has the view that the vacation should start when people leave their house, so a good service should be available to take people to the airport. Based on this idea, Tinker was developed (Hulshof, 2012).

The company started by operating at Schiphol only, but expanded their service later to several other airports. In the beginning of 2014, they operated from Amsterdam (Schiphol), Eindhoven Airport, Rotterdam The Hague Airport, Weeze Niederrhein Airport, Brussels, Oostende (Brugge) and Antwerp and they are considering expanding their position even further. Tinker has offices located in Enschede and close to Schiphol, where they investigate opportunities to improve the service and receive customer calls.

2.2. The process
Customers use to book taxis very late, even on the day when they travel to the airport. They call a taxi company which picks them up. This is not a very efficient way, since it is hardly possible for taxi companies to make an efficient planning in advance as a result of the last minute bookings. Tinker wants to change this by offering customers the opportunity to book a taxi early via internet at the webpage tinker.travel. They can book till 24 hours before they want to use the service, but they get a cheaper price if they book early. Customers can also book at the day they travel, but in that case they should call Tinker by phone to book a ride. Customers can book already one year before they fly.

Tinker schedules the bookings and sends the schedule to taxi companies as Atax, Dortmans and Somotax. These taxi companies take care of the rides and maintenance of the taxis.

The process related to Tinker will be described from two different viewpoints. These are the customer viewpoint (paragraph 2.2.1) and the global viewpoint of the company (paragraph 2.2.2). In these views all steps done by the customer to book a taxi and use the service are described as well as the required input by employees of Tinker and the carriers. The process starts when the customer visits the site and makes the booking. Actions which are not related to Tinker directly, like booking the air flight or vacation by the customer are not considered in these views.
2.2.1. Customer viewpoint

Customers who will use the service first recognize the need to be transported to or back from the airport. They will compare the possibilities (train, taxi and car are the most appropriate possibilities) and based on their situation they will decide what option to use. For Tinker, only customers choosing for taxi transport are relevant, however they can influence this decision with the price of the service. Based on information (on websites, experiences of other people and so on), customers make a decision for the company they will use. If they travel back, customers could choose to not do anything in advance at all but take a taxi when they are at the airport. Customers who choose to select Tinker will book in advance and go through the following stages to book a ride:

1. Visit the site of Tinker;
2. Filling in the booking form;
   a. Step 1: address, airport, date, time, from/to or both, number of persons;
   b. Step 2: luggage information (type of luggage, number per type);
   c. Step 3: selection of class: Economic or Comfort (allow combined rides or not);
   d. Step 4: personal contact information, e-mail, mobile phone number;
   e. Step 5: confirmation.
3. Receiving the confirmation / bill;
4. Pay Tinker for the service;
5. Printing the booking ticket.

At this moment, the service is booked. From here, two cases are considered. The first case is customers travelling to the airport; the second case is customers travelling back to home. They also could select both cases; in that case two booking tickets are printed. When the customer has booked the service to travel to the airport, the following steps are taken.

6. Waiting at the entered address for the taxi;
7. Give the printed booking ticket to the taxi driver;
8. Leave the taxi when arrived on the airport;
9. Continue to the airport.

When the customer has chosen to be brought home by Tinker, the following steps are taken.

6. Leave airplane, collect luggage and meet with Tinker employees at the airport;
7. Follow them to the taxi which is waiting for them;
8. Give the printed booking ticket to the taxi driver;
9. Leave the taxi when arrived at location.

Tinker reduces costs by not waiting at the general taxi places at Schiphol, but at a cheaper place close to Schiphol. An employee of Tinker brings the customers to that location. There could be some steps in between, for example when a customer has chosen to allow for combined rides with other customers it might be the case that the customer has to wait for the other customers. This description assumes that everything went as planned. Obviously, deviations are possible. For example, the customer could cancel the booking. Also delays can occur, which result in corrective actions to pick the customers up later.
2.2.2. Tinker viewpoint

Tinker must act correct to make sure that the customers are satisfied with the service. Tinker reacts on the input by the customer. The following actions are taken to help the customer. General actions (like making a workforce planning for the Tinker employees and keeping the site running) are left out of scope. This also holds for automated actions (as sending the payment to the customer):

Prior to the transportation of customers (planning staff: 1+2+4, Tinker service employees: 3+4).
1. Planning the ride;
2. Allocate the right vehicles to transport the customers;
3. Inform customers about a possible combination, especially if this results in a different pick-up time (when travelling to the airport);
4. On the travel day: check for possible delays and take corrective actions;

When transporting to the airport (taxi drivers).
5. Pick-up the customer at the correct location;
6. Load luggage in the taxi, check if the volumes are correct;
7. Receive the booking form;
8. Travel to airport and unload passengers and luggage at the airport;

When transporting back (Tinker bells: 9+10, taxi drivers: 11-14).
9. Receive customers at the airport;
10. Guide customers to the taxi;
11. Load luggage in the taxi, check if the volumes are correct;
12. Receive the booking form;
13. Check whether all customers are in the taxi (when combining rides);
14. Bring customers to (home) address and unload them.

Not many employees are involved during the process; the customer meets employees at the airport (the Tinker bells), taxi drivers and might get in contact with the service employees before they travel. The customer will not meet employees who are responsible for planning the rides.

2.3. Characteristics of the company

As described before, Tinker does not own taxis but uses the current taxi network to offer their service. These taxi companies use two different taxi types. The first taxi type is a car and can take at most 18 Luggage Units (LU) and 4 people. 1 LU has a cumulative size of approximately 50 cm. The second type, a small bus, can take at most 32 LU and 8 people. Tinker knows from experience that in most cases people take on average 5 LU per person. Tinker limits the maximum number of passengers to 6 to avoid that many bookings exceed the luggage limit. However, Tinker will help customers who want to travel with more than 6 passengers as well by transporting them in multiple taxis. It happened once that a large group wanted to travel, Tinker organized a bus for them.
Tinker differentiates between the following types of luggage:

- **Type 1**: 4 LU. A big suitcase, with a maximum cumulative size (length + width + height) of 180 cm;
- **Type 2**: 3 LU. A suitcase with a maximum cumulative size of 150 cm;
- **Type 3**: 2 LU: A suitcase with a maximum of 55 x 25 x 35 (length x width x height) for each suitcase;
- **Special luggage**: 4 LU (all types).

Combinations which exceed the maximum capacity of the taxi bus cannot be selected at the online booking system, the customer will get a notification to change it. They also can call Tinker to find out whether an exception is possible. On the site of Tinker, the customer gets the following screen to select the luggage given in Figure 1:

![Luggage screen](image)

**Figure 1: Luggage screen**

If possible, Tinker uses cars since these are more comfortable. Busses are used if cars are too small or when it is more efficient for schedule reasons. Figure 2 shows a car and a bus. The cost difference is small, but Tinker prefers using cars since this more comfortable for customers.

![Taxi Car and Bus](image)

**Figure 2: Taxi Car and Bus**
Customers can choose whether they allow Tinker to combine their rides with other people. If people allow that (by selecting Economic Class), they get a lower price for their booking, however they might have to deal with additional travel time. They can choose Comfort Class if they do not allow combination. The difference between the prices depends mainly on the number of people, luggage and the distance. The price reductions will be discussed in section 4.1.

Tinker uses the current taxi network by leasing taxis at carriers. Atax and Dortmans are the most important connections of Tinker. Tinker has contracts in which they can hire a taxi if there is demand in the area around the garage. The carriers are located at several places spread through the countries in which Tinker operates. If demand occurs, the closest garage (or closest available taxi) will be selected to execute the booking. After executing the ride, the taxi makes another ride or returns to the garage. The taxi always has to return to the same garage at the end of the day.

Tinker has arrangements with these carriers, each day they can schedule taxi rides, taxis are always available. It hardly happens that there is no taxi available (based on Tinker, this only has happened once). They pay a fixed price to the garage to use the taxi. The garage is responsible for maintenance and execution of the service.
3. Description of the research assignment

In this chapter the research assignment will be described. The assignment is based on the literature study and research proposal which are executed prior to this report. First a short summary of the literature research will be given. This literature study described several types of transportation problems and heuristics to solve these heuristics. In section 3.2, the findings of the literature study are linked to the characteristics of Tinker. After that, the general assignment is provided in section 3.3.

3.1. Literature research recap

In the literature study (Van Zutphen, 2014), several types of transport problems are discussed. These transport problems are Traveling Salesman Problem (TSP), Vehicle Routing Problems (VRP) (Cordeau, Gendreau, Laporte, Potvin, & Semet, 2002), (Fisher & Jaikumar, 1981) and Pickup and Delivery Problems (PDP) (Dumas, Desrosiers, & Soumis, 1991). Heuristics to solve these algorithms are the Nearest Neighbor algorithm, Sweep algorithm, Fisher & Jaikumar algorithm, Clarke & Wright heuristic and Tabu Search (Laporte, 1992), (Nagy & Salhi, 2003), (Pisinger & Ropke, 2007). Table 1 describes advantages and disadvantages of these heuristics:

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Context</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nearest Neighbor</td>
<td>Most appropriate for TSP, less for VRP, PDP</td>
<td>Easy to use and modify for new data.</td>
<td>Less appropriate for VRP and extensions.</td>
</tr>
<tr>
<td>Sweep</td>
<td>Most appropriate for TSP, less for VRP, PDP</td>
<td>No crossings in ride, easy to use and modify for new data.</td>
<td>Requires coordinates of customers, less optimal for high density areas. Less appropriate for extensions.</td>
</tr>
<tr>
<td>Fisher &amp; Jaikumar</td>
<td>Suitable for VRP, PDP</td>
<td>Splitting customers in groups (TSP can be applied here, appropriate when using Multi-depot).</td>
<td>Requires many calculations when, adding / removing customers. Less appropriate for extensions.</td>
</tr>
<tr>
<td>Clarke &amp; Wright</td>
<td>Suitable for VRP, PDP</td>
<td>Can cope with extensions.</td>
<td>Many calculations required.</td>
</tr>
<tr>
<td>Tabu Search</td>
<td>TSP, VRP, PDP</td>
<td>Finds (local) optimum.</td>
<td>Requires many calculations.</td>
</tr>
</tbody>
</table>

In a general transport problem, people or goods are transported with a vehicle to a location without any limitations. In the literature study, several extensions to the general transport problem are considered. Time Windows (in which the customers must be visited in given time ranges), Multi-Depot (where the ride can start at different positions), (Lim & Wang, 2005) and Heterogeneous Fleet (in which different types of vehicles are used) (Dondo & Cerdá, 2007) are examples of extensions. These extensions limit the number of possibilities to solve the transport problem.

Most available literature focuses on single extensions, not many sources consider multiple extensions. Here is an important gap, since these combinations are more reasonable for many companies. For example, companies may have to deal with time windows and have multiple depots. Also capacity constraints need additional research, most research focuses on bringing...
packages to customers or picking customers up. Not much research is done about cases in which both customers and packages are transported. Most literature is tested with a given (unchanging) dataset, but several companies deal with changes in the dataset at the last minute. Here is an important research gap.

Another gap is the optimization context. Many research articles focus on minimizing the total distance, but companies can be more interested in costs, or customer satisfaction for example. Even though sparing distances reduces costs, other factors influence the price as well.

Also there is not much attention for different kind of customers in the literature. In most common cases all customers are treated the same, although in reality customers have different expectations and they might require a different treatment. Considering situations in which there is differentiating among them would be a relevant addition to the current literature.

3.2. Linking literature study with characteristics of Tinker

Based on chapter 2 and section 3.1, the characteristics of the company are best described by a mix of the Vehicle Routing Problem (VRP) and Pickup and Delivery Problem (PDP). A VRP considers several customers which must be visited by a vehicle with limited capacity. This vehicle is located at a fixed depot. In a PDP the ride alternates between Pickup and Delivery locations. A customer is picked up at one location and brought to a different location. From that location the vehicle travels to another customer who must be brought to a different location. This situation can be seen as a mix of these two concepts since it brings people from a couple of central locations (the airports) to different locations or the other way around. Multiple customers can be taken to that location in one vehicle, but the vehicle has capacity constraints (which are characteristics of a VRP). In addition, a taxi that brought a customer to the airport might be able to take a different customer back home directly after that (a PDP characteristic). This means that the problem can be described both as a VRP and PDP.

Several extensions apply to the situation. The company, Tinker, has taxis located through the whole country in which they are operating, located in garages. Therefore, taxis can depart from different locations. To operate efficiently, the closest available taxi will be selected to pick up the customer at home or at the airport. Hence, a multi depot extension must be considered. In addition, multiple types of taxis are used. As described before, cars (which can carry at most 4 persons, 18 LU) and busses (carrying at most 8 persons, 32 LU) can both be used for transport. To consider this effect, the heterogeneous fleet extension applies.

Since customers must catch a flight, they must (while travelling to the airport) reach their destination on time for sure. Delays are not allowed since these results in unsatisfied customers, missing their flight. When customers travel from the airport back to home the time restriction is less hard, although customers probably are not willing to wait very long before they can travel back. Moreover, a Time Windows extension must be considered. The maximum additional waiting and travel time of the customer must be respected.

The general VRP or PDP deals with one capacity constraint. In most cases this is related to the number of people or number of inventory (packages). In this setting, both the number of passengers and the amount of luggage (measured in LU) can be a restrictive constraint. Therefore the original problem formulation must be adapted in order to cope with this limitation.
3.3. The general assignment
Currently combination is not used, meaning that for each booking one vehicle (taxi) is used. In order to combine rides efficiently, a model must be developed. Therefore, the following assignment is defined:

Develop a methodology to determine whether a (taxi) company using an online internet booking system for customers to reserve rides in advance, characterized by (1) time windows, (2) multiple types of transport options, (3) various capacity constraints, using (4) multi depots, can generate more profit by combining bookings, still adhering to the performance agreed upon.

Two ways of combining bookings will be distinguished in this report:
1. Combination;
2. Matching.

Ad.1. The first way is combination. Combination takes place when two bookings are transported from home To their destination (presented in Figure 3), or from the Destination back to home (Figure 4). It also could be the case that customers are transported to different destinations. Combination will result in additional travel time and/or waiting time for at least one of the customers. Combination transforms the list of bookings in a list of rides, a ride is a single booking, or two combined bookings.

![Diagram of Transportation steps To ride](image1)

![Diagram of Transportation steps Back ride](image2)
Ad.2. The second way of combining bookings is defined as matching rides. This procedure links a ride To the destination with a ride Back by executing the Back ride (almost) directly after the To ride. Matching transforms the list of To rides and the list of back rides in a list of drives, which could exist of a single To or Back ride, or a matched To and Back ride. There are four possibilities to match rides given in Figure 5, taking into account combination effects:

![Diagram of matching drive](image)

**Figure 5: Matching drive**

A **booking** will be denoted as a single online booking, made by a group or a single passenger. This is visualized in the No Combination settings of Figures 3 and 4 and will be notated by lowercases ($i,j,k,...$). A **ride** is a complete To or Back ride as presented in Figures 3 and 4 and will be denoted by capital letters ($I,J,K,...$). This could be a combination, or a single not combined booking. A **Drive** is defined as a matched To and Back ride, as presented in Figure 5.

The goal is to reach a cost reduction in the process by reducing distances. The reduction must be interesting to implement. Currently, Tinker offers customers a choice, they can book Economic Class in which they allow Tinker to combine, they also can choose Comfort Class if they do not want to travel with other people. Obviously, the offered price is lower for Economic Class, although the price reduction depends on other aspects like luggage and number of passengers. In section 4.1 these price differences will discussed in more detail. By offering lower prices Tinker also tries to attract more people, which can attract other people as well by Word of mouth.

To do the assignment, first the current aspects of the company are investigated. Tinker offers guarantees to the customer. There is a maximum absolute additional travel time and a maximum additional travel fraction. The maximum absolute additional travel time is a value which indicates how much the difference between the travel time in the situation in which a booking is combined and the original situation without combination may be. The maximum additional travel fraction is the maximum allowed value of the travel time in the combined situation divided by the travel time in the current situation. First a research question is defined in which the effect of this will be investigated. To execute this step, first is investigated how many bookings could have been combined with the current settings, and afterwards how many bookings could have been combined with changed settings. The model must be created in such
a way that this could be done easily for different values of the parameters. In step 2, a research question will be answered in order to get a view what kind of savings could have been achieved in the past if bookings were combined or matched.

By executing steps 1 and 2 it should be clear whether combining and matching is attractive for the company. Finally, a model must be realized which can provide a schedule for a given amount of time in which is indicated which bookings/rides should be combined and matched.

3.3.1. Increasing the number of combinations and matches

Even though Tinker offers a reduction to customers who allow combined rides it currently does not happen that customers indeed travel with other (unknown) people. In the starting phase of Tinker, this was the result of too few bookings. Since the company is currently growing fast, it is likely that the possible number of combinations will increase. The number of possibilities is limited since passengers likely depart or arrive at different time slots. Some agreements are set by Tinker in order to respect customer satisfaction. One offered guarantee is that customers who allow combined rides face an additional travel time of at most 15 minutes (0.25 hours), or 20% of the travel time. This reduces the number of possibilities, since the additional waiting time for people living very close to the airport is limited (when a customer for example lives 15 minutes away, this results in an additional waiting time of at most 3 minutes). To investigate this effect, the following research question is defined:

RQ1: To what extent does the maximum additional travel/waiting time affect the abilities to combine taxi rides, respecting customer satisfaction?

Constraints which are not directly related to these settings must be taken into account as well. If two groups travel together and are willing to combine this might not be possible due to this other factors. During this first step, the effects of the constraints will be investigated.

3.3.2. Combining and Matching rides efficiently

Bookings must be combined/matched in such a way that the total savings are maximized. It must be investigated what kind of cost reductions could be reached by combining or matching rides. The gain must have a certain level. The following research question is defined.

RQ2: What kind of cost reduction can be achieved by combining multiple To and or Back bookings and scheduling rides in such a way that multiple rides are executed with the same vehicle by doing a Back ride (almost) directly after a to ride?

Two concepts will be introduced to answer this research question. **Customer distances** are the distances in which the customer is in the taxi, given in kilometers. **Vehicle distances** are the additional required kilometers to get the taxi to the customer and to get from the customers destination back to the depot of the taxi (or vice versa). The sum of these should be minimized in the final model. In the research question must be found whether it makes sense to combine or match rides by giving an indication of the savings in a tool.
3.3.3. Combining the aspects
When the total of savings is sufficient, the allocation process must be optimized. Since the process involves picking up multiple people and bringing them to one location or moving people from one point (depot) to different locations, a transportation heuristic must be implemented, considering limitations which apply in this situation.

Implement a Transportation Heuristic dealing with extensions to improve the allocation of combined / matched rides.

As described in section 3.2, the context is closely related to characteristics of the Vehicle Routing Problem (VRP) and Pickup and Delivery Problem (PDP). Combination can be viewed as a VRP, multiple customers travel together in a vehicle with limited capacity. The matching procedure can be viewed as a PDP, since a Back ride is executed (almost) directly after a To ride here. When both combination and matching takes place, it is a mc-PDP (multi capacity Pickup and Delivery).
Combination and Matching must be executed separately, since the number of possibilities depend on each other. The method with the highest expected savings per kilometer must be executed first.

The next chapter describes the characteristics of the data of Tinker. Based on these findings, a simple model is set up in chapter 5 to get an idea whether combination / matching could be attractive under various conditions and a linear programming model will be presented to combine bookings and match rides in an efficient way.
4. Analysis

In the previous chapter the research assignment has been described. Based on the characteristics of the company related information will be discussed in this chapter. In section 4.1 is described how prices for bookings are calculated and what kind of reductions are offered for customers who allow Combination. In section 4.2, data about booking information will be introduced. Each factor of the bookings will be described. Section 4.3 summarizes characteristics of the environment related to Tinker. This mainly involves the airports. Flight patterns of Schiphol Airport and the fractions of delays will be presented. These directly influence the demand pattern at Tinker since people book taxis related to the flights on the airport. Most bookings are from or to Schiphol Airport, therefore data of this airport will be used. The situation at other airports is comparable to Schiphol.

4.1. The current price strategy

In this section is described how the price reduction for customers who allow combination is affected by the characteristics of bookings. As described in section 2.2.1, bookings have several aspects. There are two classes from which customers can choose: Economic Class (allow combined ride with other customers) and Comfort Class (no combination). Since the amount of customers selecting Economic Class depends on the price reduction which offered, it is important to understand the behavior of the price system. The effect of these aspects on the price reductions for customers who allow combination is discussed in this section (Tinker: Smarter in Travel, 2013). The following aspects are discussed:

1. Travel distance;
2. Number of passengers of Amount of luggage;
3. Booking days in advance and direction of the flight;

In Appendix 1, an example is given to illustrate the effects which will now be described.

Ad. 1. The price reduction is affected by the distance to the airport. The offered reduction based on distance only has a minimum value of 0,95 euro and increases linear to a maximum of €9,95 for rides which are at least 219 kilometer.

Ad. 2. The number of passengers (pax) and luggage (LU) also have an effect on the offered reduction. Depending on the amount of pax and LU, the reduction based on the distance is multiplied. For each setting this is a fixed value. For large kilometers, there is no real effect of pax and LU and the price is multiplied by 1,13. For smaller distances, the offered price reduction is multiplied with 2,11 for situations with only 1 passenger, changing to a value around 1,73 when the amount of pax / LU reaches the maximum. There is a linear decrease in the reduction.

Ad. 3. Since Tinker uses early booking discount, the price for Back rides is a fraction cheaper as the price for To rides. The difference is 1 euro, independent of the other inputs like distance and luggage. The booking days in advance also influence the price. For customers who book 2 days in advance the reduction of choosing Economic Class is zero. For other factors, the discount
which is a result of the distance and pax + luggage is multiplied by a factor. The factor increases linear from 0 to 1.96.

To conclude, the price reduction is calculated as follows:

\[
\text{Economic Price} = \text{Comfort Price} - \left( \frac{\min(x, 219)}{219} \times 9.95 \times \left(1.13 + \frac{6 - \text{pax}}{6} \times \frac{32 - \text{LU}}{\text{LU}} \times 0.98 \right) \times 1.96 \times \frac{\min(\text{Booking Days in Advance}, 61)}{61} \right) - 1 \times \text{direction},
\]

in which direction equals 1 if the booking is a Back ride and 0 otherwise.

It must be noted that Tinker continuously improves its price strategy, which means that the current strategy can differ slightly from the described strategy.

Tinker uses different prices for different settings because the maximum willingness to pay is different for these types. Based on market research, The described patterns are used to match the maximum willingness to pay as good as possible to attract more customers.

### 4.2. Dataset of Tinker

General characteristics of the bookings will be described in this section. A data set of 10,000 bookings between the start of Tinker (November 2012) and September 2013 is used. People who booked both a To and a Back ride are entered twice in the data. The data has been checked for errors first and corrective actions are taken. Microsoft Excel and SPSS are used for the analysis. 8 inputs are discussed.

1. **Number of passengers**: The number of people travelling together (the number of passengers for one booking). This integer number varies from 1 to 6 people for the online bookings, however there are samples in which more than 6 persons are travelling;
2. **Luggage Units**: the volumes per type of luggage;
3. **The booking time in advance**: given in days;
4. **From/to**: indicates whether the booking is from the home location to the airport, the other way around or both.
5. **Distance**: the distance from the customers location to the airport;
6. **Economic Class or Comfort Class**: if the customer allows combination he selects Economic Class, otherwise Comfort Class is selected;
7. **Carriers**: companies which are collaborating with Tinker to execute the rides;
8. **Travel speed**: the average speed of the taxi during a ride.

In paragraph 4.2.1 to 4.2.8 these inputs will be described and in paragraph 4.2.9 additional conclusions are given.

#### 4.2.1. Number of passengers

Based on this dataset, most people use the service while travelling alone (2,564 bookings) or with two people (4,401 bookings). The number of 3 people (1,189) and 4 people (904) is much lower. As described before, bookings with at least 5 people must be executed by a taxi bus, this dataset contains 942 bookings for which this is the case. In 160 of them, 7 or 8 people are transported. The complete pattern is presented in Appendix 2 in Figure 12.
4.2.2. Luggage Units

As mentioned in section 2.3, Tinker distinct four types of luggage: type 1, type 2, type 3 and special luggage. The sum of these (in Luggage Units) will be considered. Most customers take 12 LU with them (20.6%). 2.4% of the customers did not take any luggage with them. The average of all bookings is 11.34 LU. In 13.6% of the bookings, the total LU was above 18, requiring a taxi bus. A Figure with all totals per number of LU is given in Appendix 1 in Figure 13. Looking at both luggage and number of passengers, it can be found that bookings in which more passengers travel the total luggage units are higher as well. Comparing luggage for different number of passengers results in an increasing pattern. The averages are given in Table 2, next to the average LU, the number of bookings with more than 18 LU are given:

<table>
<thead>
<tr>
<th>Number of Passengers</th>
<th>Total Bookings</th>
<th>Average LU</th>
<th>Number &gt;18</th>
<th>Fraction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2564</td>
<td>4.9</td>
<td>8</td>
<td>0.003</td>
</tr>
<tr>
<td>2</td>
<td>4401</td>
<td>10.0</td>
<td>29</td>
<td>0.007</td>
</tr>
<tr>
<td>3</td>
<td>1189</td>
<td>14.1</td>
<td>92</td>
<td>0.077</td>
</tr>
<tr>
<td>4</td>
<td>904</td>
<td>18.9</td>
<td>460</td>
<td>0.509</td>
</tr>
<tr>
<td>5</td>
<td>465</td>
<td>23.3</td>
<td>354</td>
<td>0.761</td>
</tr>
<tr>
<td>6</td>
<td>317</td>
<td>25.4</td>
<td>277</td>
<td>0.874</td>
</tr>
<tr>
<td>7</td>
<td>62</td>
<td>25.2</td>
<td>55</td>
<td>0.887</td>
</tr>
<tr>
<td>8</td>
<td>98</td>
<td>25.6</td>
<td>81</td>
<td>0.827</td>
</tr>
<tr>
<td>All bookings</td>
<td>10000</td>
<td>11.34</td>
<td>1356</td>
<td>0.136</td>
</tr>
</tbody>
</table>

For 1-3 passengers, it is hardly necessary to use a taxi bus, but for 4 passengers it was required in 51% of the cases. In total 1.531 of the 10.000 bookings had to be executed by bus, the remaining 8.469 bookings could be executed by car.

4.2.3. The booking time in advance

Many people still book at the last minute. From the sample set, 45% of the customers booked between 0-10 days before they fly, which is quite a lot. When looking at monthly data, this fraction is more or less constant, so it is not likely that this fraction will differ in the future. 22% of the customers book between 10 and 20 days in advance. Only 2% of the bookings are booked at least 100 days in advance (with a maximum of 269 days). In Appendix 2, additional Figures (14+15) are given to describe the booking behavior of customer. There is no relation between the booking days in advance and other booking characteristics.

4.2.4. Travelling from or to the airport

From the dataset, 5.738 people booked a ride to the airport and 4.262 people booked a back ride. The higher fraction of people travelling to Schiphol is probably the result of the fact that people need to call a taxi when they are at home to get to the airport but they can wait at the taxi standing place of the airport to get a taxi to travel back.
4.2.5. Distance to the airport

The distances between the customer and airport are entered in the dataset as well. The maximum distance in the dataset is 228 kilometers and the minimum 1.7 kilometers. While clustering the data in groups of 10 kilometers it can be found that most bookings (26.9%) are between 10-20 kilometers. There is a decreasing pattern (with an outlier in 40-50 kilometers, which is a result of large cities in that area). Only 51 bookings are 200 kilometers or longer. In Appendix 2, the results per cluster are given in Figure 16. Making a scatter plot of the data, using coordinates of the postcodes of the customers results in Figure 6:

![Figure 6: Scatter plot of customer locations](image)

Looking at Figure 6, it can be concluded that the scatter plot looks like the Netherlands, indicating that bookings come from all places. There is no relation between distance and other characteristics of bookings.

4.2.6. Class type

Customers can choose to allow having their booking combined or not. Combining rides reduces costs and customers who allow combination get a cost reduction. However, not all people are willing to travel together with other people. To be able to offer taxi transport to these people as well, Tinker decided to introduce Economic Class (allow combinations) and Comfort Class (not allowing combinations).

The class type was introduced on 31 October 2013, so in the discussed dataset the class type was not used. To find information about this parameter, data with a booking date between November 2013 and March 2014 has been used. In this period (5 months), 7,377 bookings had Economic class (62%) and 4,462 Comfort class (38%). Looking at monthly data, the percentage of customers choosing Economic Class varied between 56% and 70%. Since the price reduction offered to customers (as described in section 4.1) depends on the booking characteristics, it is likely that the fraction of people choosing Economic class depends on the booking characteristics. The most important relations are:
1. Passengers and luggage;
2. Distance;
3. Time.

Ad.1. As mentioned in section 4.1, for each added passenger or LU the offered price reduction for customers choosing Economic Class increases. The probability that a booking will be combined decreases, so it is more attractive for customers to select Economic class if they travel with more passengers / luggage. Due to this, it is observed that 60% for customers travelling with limited quantities allow combinations, increasing to about 80% for customers with more passengers / luggage. In Appendix 2 this effect is presented in Figure 17.

Ad.2. The offered compensation for selecting Economic Class increases when distance increases. However, people who live close to the airport will likely be the ‘first customer’, meaning that they will be brought home first or picked up last and therefore are not facing additional travel time while travelling with other customers. The percentage selecting Economic Class is high for people living close (0-20 kilometers) to the airport (90%), reducing to around 40% for people living further away (with a large likelihood being the ‘second’ customer and facing additional travel time). In Appendix 2 this is illustrated in Figure 18.

Ad.3. The time of the day has an effect as well, however it appears to be not very big. In the night, the fraction of people allowing combinations is a little below the average (50%), and the percentage is a bit higher in the morning (70%). Appendix 2 shows the percentages over the day in Figure 19 (all bookings) and 20 (bookings split in To/Back) and Table 13. Based on the data it can also be concluded that there is no real difference between people travelling back or to the airport in these time slots.

4.2.7. Information about carriers
Another important aspect is the carriers. In Figure 7, the locations of the carriers are indicated (Groningen, Eindhoven and Schiphpol are added as reference points:

Figure 7: Graphical distribution carriers

There are no carriers located in the North of the Netherlands. For a booking in Groningen, many additional kilometers must be travelled to get to the customer from the closest garage (Arnhemsetaxi). This also holds for bookings from Limburg (which will probably be executed by Dortmans). Especially for these bookings, combining is an attractive way to reduce costs.
4.2.8. Speed of the taxi
The speed could influence the time as well. For shorter distances, the average speed is likely to be smaller since for longer distances, large parts of the ride are on the highway. There is an increasing pattern from about 40 km/h for small distances to 100 km/h for long distances. In Appendix 2 (Figure 21) presents the increase clustered in groups of 10 km. Also for different times the speeds might differ, since there is less traffic on the roads during the night. However, the influence of time is minimal, if the used time unit is an hour, the difference between the time with the minimum speed (5:00-6:00, 69.4 km/h) and the maximum speed (9:00-10:00, 76 km/h) is minimal. This difference also can be the result of the difference in distance. It can be concluded that distance has an influence on the average speed, but the time of the day from the ride does not have a large effect. Figure 22 of Appendix 2 shows this effect.

4.2.9. Other observations
9.827 of the 10.000 bookings are linked directly to Schiphol. From the remaining 173 rides, 70 travelled to Rotterdam Airport, 45 from Rotterdam back, 32 to Eindhoven and 20 back, leaving 6 rides open. These are special bookings, for example two families who are travelling together to the airport, but are picked up at a different location. The bookings are related to the flight patterns of these airports. These patterns are discussed in the next section.

4.3. Flight data
The demand pattern of Tinker is related to flight patterns of Schiphol and other airports. The arrivals and departures on Schiphol follow a regular pattern (Schiphol Amsterdam Airport, 2014). Most landings (arrivals) take place between 8 and 9 in the morning and between 19 and 20 in the evening. For the take-offs (departures) the pattern has a different peak. Here, most action takes place between 10 and 11 in the morning at between 21 and 22 in the evening. The total flights per time of the day are given in Table 3:

<table>
<thead>
<tr>
<th>Session</th>
<th>Time</th>
<th>Landings</th>
<th>Take-offs</th>
<th>Take-offs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Night</td>
<td>23:00 – 05:59</td>
<td>14.866</td>
<td>5.878</td>
<td>20.744</td>
</tr>
<tr>
<td>Early Morning</td>
<td>06:00 – 06:59</td>
<td>6.303</td>
<td>3.756</td>
<td>10.059</td>
</tr>
<tr>
<td>Day</td>
<td>07:00 – 18:59</td>
<td>150.843</td>
<td>154.852</td>
<td>305.695</td>
</tr>
<tr>
<td>Evening</td>
<td>19:00 – 22:59</td>
<td>40.781</td>
<td>48.286</td>
<td>89.067</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>212.793</td>
<td>212.772</td>
<td>425.565</td>
</tr>
</tbody>
</table>

In Appendix 3, the flights are clustered per hour in Figure 23 (Schiphol Amsterdam Airport, 2014). The exact destination of the flight will be left out of scope, since this will not affect the taxi ride from Tinker. Only continental and intercontinental flights need to be differentiated for departures since Schiphol suggests passengers to be 2 hours before the flight on the airport for continental flights and 3 hours for intercontinental flights. In 2013, 52,6 million passengers travelled from or to Schiphol; 35,7 million (68%) of them travelled to an airport in Europe (domestic flights), 16,9 million (32%) travelled to an intercontinental destination. It will be assumed that customers will follow these time suggestions by Schiphol (Schiphol Amsterdam Airport, 2014). Arriving customers require approximately 40 minutes after landing being ready to go home. Currently, the ride back is planned one hour after the actual landing.
Based on the sample of chapter 4 it is found that Tinker is more used in the night hours, it does not follow the landing pattern of Schiphol exactly, although the number of customer of Tinker is also higher around peak times of Schiphol. Since To rides are executed 2-3 hours before the depart, the pattern is shifted with a couple of hours. The Back rides patterns is shifted one hour as well. In Appendix 3, Figure 24 and 25 illustrate this effect in more detail.

Another important characteristic from Schiphol which influences the process at Tinker is the punctuality. Punctuality is defined as “the percentage of flights departing/arriving within no more than 15 minutes of its scheduled time on/off blocks” (Schiphol Amsterdam Airport, 2014). The percentage of punctually in 2013 was 81,9% for departures and 87,6% for arrivals. It does not happen that a plane departs too early for more than 15 minutes, therefore this does not have a big impact on the schedule of Tinker for the To Rides. For Back rides, this is an issue, since taxis are scheduled to pick up customers at a desired time. Based on Tinker, delays are communicated in advance (at least 4 hours), so Tinker and the taxi carriers have a reasonable time to change their schedule based on delays. Last-minute delays hardly happen (Schiphol Amsterdam Airport, 2014), therefore Tinker is able to take corrective actions on time.

4.4. Summary of the analysis

In this chapter the price strategy of Tinker, data of Tinker and flight patterns of airports are discussed. Tinker offers a reduction for customers who allow combination which depends on the number of passengers, luggage, booking days in advance, distance and direction. Especially distance and booking days in advance have a large effect on the reduction.

Based on the current data, most bookings are with a few passengers and limited luggage, only 15% of the bookings had more than 4 passengers or more than 18 luggage units meaning that the ride must be executed by a taxi bus. Most customers book their flight short before they fly and most bookings are from customers living close to the airport (10-20 km), but bookings come from all places in the Netherlands, even from people living over 200 kilometers.

Customers are more likely to allow combination when they take more luggage or travel with more people or live within 20 kilometers of the airport. Reasons are that these customers have a smaller probability that their bookings are combined, and if there is a combination they are likely to be picked up last, meaning that they do not face additional travel time.

The pattern of the taxi rides is related to the flight patterns of airports. While looking at hour slots, it can be concluded that peak hours for Tinker are 2-3 hours before peak hours of the airports for To rides (which is a result of the fact that customers must be 2-3 hours on the airport before they fly) and 1 hour after peak hours of the airport for Back rides (a result of the fact that customers require about 45 minutes to collect their luggage and move to the taxi).

There is an interesting amount of bookings who allow combination (62%). So combining rides could be a tool to reduce costs. Based on this information, a mathematical model will be formulated to combine bookings in the next chapter.
5. Model formulation

In this chapter, a model will be formulated to find answers to research question 1 and 2 presented in chapter 3. In section 5.1, a general description of the model will be given about which input is required and which input will be used to combine rides. In section 5.2, the maximum times will be discussed to answer research question 1 in paragraph 3.3.1. The effect of a maximum additional travel distance/time and waiting time will be investigated in this model. Section 5.3 discusses the potential savings of combining and matching bookings. A simple model will be presented to give an indication what kind of savings could be gained for each setting, to answer research question 2 of paragraph 3.3.2. The model will have two steps, first a Combination model is presented to combine rides as efficient as possible and second a Matching model is given to match rides. This section builds on the results of section 5.2, since the maximum times affect the combination probabilities. These are simple Excel models which have limitations, but can be used easily. In section 5.4 a Linear Programming model will be given which does not have the limitations of the Excel model, but is more complex to use. This model will be able to calculate the optimal solution, resulting in higher cost reductions.

5.1. Model description

The following input must be available to use the models. A list of bookings must be available, containing the following information:
- Start location (given in coordinates, longitude and latitude);
- Destination (to where the customer must travel, given in coordinates);
  - To calculate travel times, the travel speed must be known. Alternatively, instead of the coordinates, the model can be used as well if all distances / travel times between all customers and destinations are known;
- Number of passengers (pax);
- Number of luggage units (LU);
- Arrival / departure time, the moment at which the customer must arrive at the destination or is ready at the destination to travel back home;
- Direction: whether the customer travels to the destination or back;
- Class: to what extent does the customer allow to have a ride combined with other customers;
- Carriers: The carriers which can execute the rides. The coordinates of them must be known, or the distances/travel times to the related customers / destinations;
- Fleets: A fixed number of different fleets of vehicles is available, the maximum number of passengers and luggage must be known of each fleet type;
- Load and unload times: given in time per passenger and/or time per LU.

This is a list of input which will be used in the final model. In addition, booking time: the moment that the booking is registered, could be used however this value does not affect the results. Some effects can be ignored by setting the values to zero or considering only 1 type (for example if loading times can be neglected by setting these to zero and if there is only one carrier, only that one is listed).

As mentioned before, there are two possible ways to combine bookings. One way is defined as combination, two bookings are executed simultaneously in one vehicle. Customers must have
the same travel direction to be able to use this type. In addition, both customers must allow combination. An alternative is matching, in which after bringing customers to a location a ride to bring different customers from his location back to home is executed almost directly afterwards. Here, the direction of the two bookings must be the opposite of each other. To combine or match customers, first a decision must be made based on which parameters the customers are combined or matched. The following parameters can be used:

1. Distance: Location and/or destination of the customer;
2. Arrival / Departure time of the customers;
3. Number of luggage and passengers;
4. Moment of booking.

Ad.1. The first way is to combine customers based on the distance. This can be done at several ways, depending on how the model is set up. One way is to combine customers based on the distance between customers, using a Nearest Neighbor algorithm to combine customers who live close to each other. Alternatively, a customer living far away can be selected and a customer living on the route to the airport is combined with this customer. Both ways result in limited additional travel times for the customers and high savings but in long waiting times since this could combine a customer who travels in the morning with a booking in the evening.
Ad.2. When the arrival and departure times are used as decision parameter (combining customers who have the same arrival or departure time), waiting times will be small. However, this results in an inefficient combination, since customers living far away from each other might be put in one ride, resulting in much additional travel time and limited savings.
Ad.3. When luggage quantity and number of passengers are used, bookings will be combined in such a way that the occupation of the vehicles is as high as possible. This is a Bin Packaging Problem which can be solved by the First Fit decreasing algorithm to minimize the number of vehicles required, as explained in Appendix 4. The total luggage and passengers is as close as possible to its maximum of the best (cheapest) fleet. Even though this results in an efficient use of the taxis, this results in a less efficient route since customers living far away or customers with different arrival / departure times could be matched, reducing customer satisfaction.
Ad.4. Combining based on the moment of booking (allocating the customer who booked first to the customer who booked second) is also a possibility, however this will have a worse impact on additional travel time, waiting time and occupations of the taxis.

Based on the described possibilities it can be concluded that the allocation of rides is a trade-off between the following factors:

1. Kilometer savings;
2. Additional travel distance/time, distance/time that customers travel longer in the vehicle related to the situation in which they are not combined;
3. Waiting time, time that customers arrive too early at the destination, or must wait when they travel back in a combined setting, and/or the time that a taxi driver / customer must wait between two matched rides.
4. Occupation of the taxi, optimizing the use of cheaper fleets.
In Table 4, a trade-off between these factors is given. A minus (-) indicates a worse score on the factor, a plus (+) indicates a good score. The ranking indicates which method is preferred.

Table 4: Trade-off between combination methods

<table>
<thead>
<tr>
<th>Combining method</th>
<th>Kilometer Savings</th>
<th>Additional Travel time customers</th>
<th>Waiting Time customers</th>
<th>Occupation of the taxi</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distance</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>Arrival/Depart time</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>2</td>
</tr>
<tr>
<td>Pax + LU</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>2</td>
</tr>
<tr>
<td>Booking in Advance</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>3</td>
</tr>
</tbody>
</table>

Distance scores the best. This method results in the highest savings, but it has a worse impact on the waiting time of customers. To reduce this effect, maximum values for waiting times will be considered, meaning that bookings only can be combined if the arrival/depart times are close to each other. In addition, a maximum allowed travel time will be considered to avoid long additional travel times for customers. Since the costs between the different types of taxis are small, factor 4 is less important, therefore this effect will not be taken into account in the model. In Appendix 4, a small example indicates how rides are combined for a given dataset.

5.2. Effect of maximum times

In this section it will be investigated what the effect is of the maximum additional travel and waiting times on the number of possible combinations and matches. Data of the past will be used to do this analysis. To indicate whether two rides could be combined, constraints must be checked. Microsoft Excel will be used to measure these effects. The data is split in two groups based to the destination (to or back) since only rides in the same direction can be combined. Additional travel time can be considered absolute or relative. The absolute travel time is the difference in time between the situation in which the ride is combined and the original travel time.. The relative additional travel time is a fraction, which is the travel time in the combined situation divided by the time in the original situation, introduced to keep the additional travel time for customers living close to the airport limited. In addition, maximum waiting times are considered. These are related to the time that the customer or taxi driver waits at the destination to be able to have a combination or match with a customer travelling at a different time. A model has been built to calculate the number of rides which could have been combined/matched. The following assumptions are used:

1. At most two bookings are combined;
2. Travel distances/time from A to B and from B to A will be assumed to be equal (symmetry is assumed);

The first assumption reduces the number of possibilities. However, it is hardly possible to combine more than two customers (due to maximum allowed travel times) and since complex calculations are required to combine multiple bookings, at most two bookings are combined. The effect of this assumption depends on the situation in which the model is used.
Assumption 2 does not hold in reality, however the effect of assuming symmetry is in most cases limited to a few kilometers / minutes. Therefore, the effect of this assumption is not large. If distances/travel times are entered instead of using coordinates to calculate the distances, this assumption is not used.

5.2.1. Calculating the distances
To measure the effect of the maximum travel time and its fraction and the maximum waiting time on combining probabilities, it is necessary to know the distances. If all distances between all customers / destinations are known, these can be imported from the dataset. If this is not the case and only coordinates are given, the coordinates can be used to estimate the distances. The following notation is used:

- $d_{i,a_i}$: distance from customer $i$ to the destination of that customer (denoted by $a_i$)
- $d_{i,j}$: distance between customer $i$ and customer $j$.
- $d_{i,a_j}$: distance from customer $i$ to the destination of customer $j$.

Since symmetry is assumed (assumption 2), these distances are equal to $d_{a_i,i}$, $d_{j,i}$ and $d_{a_j,i}$ respectively. The calculation of the distances will be done using a distance formula derived by Pythagoras (Broekmeulen, 2011). Here, the formula is given for customer $i$ to his related airport $a_i$, the formula also can be used for the other settings:

$$d_{i,a_i} = \sigma \sqrt{(\text{long}_i - \text{long}_{a_i})^2 + (\text{lat}_i - \text{lat}_{a_i})^2}$$

$\text{long}$ and $\text{lat}$ represents the longitude and latitude, presented in degrees. Depending on how the coordinates are presented in the dataset, they might need to be divided by 1,000 or 1,000,000 to get the correct distances in kilometers. The format of the coordinates should be (a,bb) and (cc,dd), for example, the values could be given as \(\text{long}_i = 5^\circ 23'11", \text{lat}_i = 50^\circ 46'27'\), or 5.230, 50.460, or other forms. These values must be denoted as 5,23 and 50,46. $\sigma$ is a scaling factor, which represents the number of kilometers per degree. The value of $\tau$ depends on the road density, hills, rivers and other forms of noise in a field.

In addition to the distance, average speed should be calculated to be able to calculate the travel times. Times are used since maximum allowed additional travel times are communicated to customers. The average speed, denoted by $v_{i,a_i}$ (while considering customer $i$ and his destination) is given in km/h and depends on the travel distance. As described in paragraph 4.2.8, larger distances are likely to have a higher average speed since more kilometers can be driven at highways. A formula for the average speed will be based on historical data, the travel distances ($d$) between the customer and destination and related travel times ($t$) of bookings in the past are known and based on these a formula can be derived which also can be used to calculate the average speed for other settings (like distances between customers). The average speed can be found by dividing the distance by the speed, $t_{i,a_i} = \frac{d_{i,a_i}}{v_{i,a_i}}$. Alternatively, route planners as Google Maps could be used to calculate the distances and travel times, but the general formula will be used since this can be added in a model.

While travelling, customers must be ready at a time to be picked up. They enter the vehicle, travel to a location and then leave the vehicle with their luggage. Customers with more luggage
will require more time to enter or leave the vehicle than customers with fewer volumes. Therefore loading and unload times will be taken account in addition to the travel times. The leading time of booking $i$ can be defined as $l_i = \text{pax} \times l_{\text{pax}} + LU \times l_{LU}$. The unload time can be defined similar: $u_i = \text{pax} \times u_{\text{pax}} + LU \times u_{LU}$. For single passengers with a small number of luggage these times will have a small effect, but the model will also hold for other applications as vehicles transferring containers to supermarkets for example. In that setting, load and unload times are more important.

5.2.2. Constraints

Bookings can only be combined if all constraints are met. Constraints of combination and matching are different. First the combination constraints are discussed in paragraph 5.2.2.1, followed by the matching constraints in paragraph 5.2.2.2.

5.2.2.1. Combination constraints

The combination constraints are split in the following categories:

1. Date and time. The arrival/depart times of bookings may differ at most the maximum allowed waiting time for the customer;
2. Capacity. The sum of both luggage units and passengers for two bookings which are related may not exceed the maximum values;
3. Distance/time. Additional travel times may not exceed the maximum allowed values.

Ad.1. To measure whether the waiting time is not exceeded, a variable is introduced: $\xi_i$: maximum additional waiting time in hours at the destination for customer $i$.

The arrival time ($at$) of two bookings for a ride to the destination or the departing time ($dt$) of two bookings for a back ride may differ at most the value of the maximum allowed waiting time of the customer, denoted by $\xi_i$ for customer $i$. This must hold for both customers ($i$ and $j$) involved in the combination. Thus:

$$\begin{cases} at_i - at_j \leq \xi_i & \text{to ride} \\ dt_j - dt_i \leq \xi_i & \text{back ride} \end{cases}$$

Ad.2. It must hold that number of luggage ($LU$) and the number of passengers ($\text{pax}$) cannot exceed their maximums. These formulas are not related to maximum waiting or travel times, but these still must be met. In mathematical notation this equals:

$$LU_i + LU_j \leq LU_{\text{max}}$$

$$\text{pax}_i + \text{pax}_j \leq \text{pax}_{\text{max}}$$

Ad.3. The last constraint deals with the travel times. To keep customer satisfaction in account, it must hold that the total travel time in the new setting (in which the ride is combined) is not larger as the original travel time plus the maximum additional travel time. In addition, the travel time in the new setting divided by the original setting may not exceed the maximum additional fraction distance/time. This must hold for both customers. The following variables are used:
The exact formula depends on the setting of how rides are combined. Considering To rides, if customer A travels to destination 1 and customer B travels to destination 2, the following combinations are possible:

1. AB12 (from customer A to customer B to the destination of customer A to the destination of customer B)
2. AB21 (from customer A to customer B to the destination of customer B to the destination of customer A)
3. BA12 (from customer B to customer A to the destination of customer A to the destination of customer B)
4. BA21 (from customer B to customer A to the destination of customer B to the destination of customer A)

In addition, A1B2 and B2A1 could be considered, but in that case customers do not travel together, therefore these combinations are left out of scope for this model. In the Back scenario, the formulas are similar, however in that case A and B represents the destination of the customers and 1 and 2 the location of the customers. For the first setting (AB12), the formulas are as follows for the scenario travelling from the customers' location to the destination. $t$ denotes the travel times between the locations, $l$ reflects the load time and $u$ the unload time.

**Absolute maximum additional travel time**
Customer 1: $t_{A,B} + t_{B,1} + l_B + u_B \leq t_{A,1} + m_A$
Customer 2: $t_{B,1} + t_{1,2} \leq t_{(B,2)} + m_B$

**Maximum additional travel fraction**
Customer 1: $\frac{t_{A,B} + t_{B,1} + l_B + u_B}{t_{A,1} + l_A + u_A} \leq 1 + f_i$
Customer 2: $\frac{t_{B,1} + t_{1,2} + l_B + u_B}{t_{B,2} + l_B + u_B} \leq 1 + f_j$

The left hand side of the absolute maximum additional travel time will be defined as $t_{comb,i}$, which is the required time in the combined setting. The formulas for the other settings are given in Appendix 5 in Table 19. For each setting, these two constraints are checked and must both be respected. If this is the case for at least one setting, the third constraint is met.

### 5.2.2.2. Matching constraints

When the combination step is executed, the output is used for the matching procedure. As a result, capacity constraints do not have to be checked again since this already has been done in the combination step; the required capacity is the maximum of the capacity of the selected To and back ride. The difference between the arrival date and time ($at$) of the (combined) To ride and the departure time ($dt$) of the (combined) Back ride must be checked. For both the customer and vehicle driver, this difference must be acceptable. The following variables are used:
\( \eta \): maximum time in hours for a vehicle to wait at the destination. This is not related to the customer, but to the vehicle driver.

\( t_a \): travel time from the end location of the To ride to the starting location of the Back ride.

\( \rho_i \): slack time of booking \( i \). \( \rho_i = m_i + \xi_i \) minus the waiting times and absolute additional travel and (un)load times resulting of the combination step.

\( \rho_f \): the slack time of ride \( f \) (which is the minimum of the slack times related to the bookings in that ride, \( \rho_f = \min(\rho_i, \rho_j) \)).

For the vehicle driver it must hold that the difference between the arrival and departure times is not larger than \( \eta \).

\[ dt_j - at_i \leq \eta \]

For the customer(s) in a To ride, it must hold that the customer arrives as earliest on \( at_i - \rho_i \) and as latest on \( at_i \). The customer(s) in the Back rides may not have to wait longer than the slack time left, so the arrival time of the matched To ride, so: \( at_i \leq dt_j + \rho_j \). Possible travel times between the end location of the To ride and start location of the Back ride are not considered, only matches of rides from the same airport are allowed in this model.

In the next paragraph models to decide on the optimal values for \( m_i, f_i, \xi_i \) and \( \eta \) are presented for both Combination and Matching. These constraints are added in those models.

5.2.3. The Models

In this paragraph models to measure the effect of the values of \( m_i, f_i, \xi_i \) and \( \eta \) will be described. In subparagraph 5.2.3.1 the combination model to determine the optimal settings is described and in subparagraph 5.2.3.2 the corresponding matching model is given.

5.2.3.1. Combination model to optimize parameters

To calculate whether two bookings can be combined, all constraints stated in paragraph 5.2.2.1 must be met. In section 5.1 is decided to combine bookings based on distance. To combine bookings in such a way, bookings are first sorted in the following order to have bookings which are likely to be combined with each other close to each other:

1. Postcode of the customer;
2. Postcode of the destination;
3. Time (arrival time for To rides, departure time for Back rides);
4. Class of the booking.

In addition, the data is split up in a list with bookings To the destination and a list with bookings Back, the model is executed for both lists. To determine whether bookings can be combined, bookings are compared with their neighbors (the booking just above and just under each booking), as a result of sorting, bookings which are likely to be combined are close to each other. So it can be indicated how many bookings can be combined with at least 1 other booking. The model starts with booking \( i = 1 \). This booking is compared to booking \( i + 1 \). The date and time and capacity are checked first. Also, the maximum allowed absolute travel time and fraction are checked. If all these values are OK, the rides can be combined. In the next step,
booking \(i + 1\) (which already is compared with booking \(i\)) is compared to booking \(i + 2\), continuing till the end of the dataset (when booking \(n - 1\) is compared with booking \(n\), the last booking of the list). Afterwards, the number of bookings that can be combined can be found. In Appendix 7a, this approach is described in mathematical formulation.

5.2.3.2. Matching model to optimize parameters

To find how many rides can be matched, for each ride of the To rides list (of all classes, including classes which do not allow combination) it must be checked with how many rides it could be matched with the Back rides list. As mentioned before, the output of the combination step is used for the matching step, which means that bookings which are combined are considered as one ride.

Each booking (in both the To rides and Back rides list) has a range in which the ride must arrive/depart. The earliest arrival time of a To ride booking is equal to the arrival time minus the minimum slack time (\(\rho_{i}\)). The latest arrival time equals the original arrival time.

The arrival time of the vehicle driver does not have any constraints, but affects the depart time. The latest depart time is the original arrival time plus the time that the vehicle driver is allowed to wait (\(at_{i} + \eta\)). For Back ride bookings, the earliest depart time is the original depart time and the latest depart time is the original depart time plus the minimum slack time (\(\rho_{j}\)). So the range to arrive at the destination for the vehicle driver is \((at_{i} - \rho_{i}, at_{i} + \eta)\) and the range to depart is \((dt_{j}, dt_{j} + \rho_{j})\). If there is overlap between the ranges, the rides can be matched.

Each To ride must be compared with all Back rides (and the other way around). The result is a number of possible matches for each ride, the sum of this are the total number of possible matches. In Appendix 7b, this approach is described in mathematical formulation.

The models described in 5.2.3.1 and 5.2.3.2 can be used to test the effects of the parameters \(m_{i}, f_{i}, \xi_{i}\) and \(\eta\). In the next section, a model is described to calculate the savings.

5.3. Selecting the most optimal drives

In section 5.2 a method is described to investigate the effect of the maximum absolute additional travel time (\(m_{i}\)), the maximum additional fraction (\(f_{i}\)), the maximum allowed waiting time of the customer (\(\xi_{i}\)) and the maximum waiting time of the vehicle drivers (\(\eta\)). The next step is to combine and match bookings in such a way that the costs are minimized. Related costs in this context are costs to travel from the depot of the closest carrier to (or back from) the customer/destination and transport the customer(s) between their location and the destination, expressed in kilometers. These costs involve distances and travel time. Since the travel time depends on the distance, minimizing the distance of the rides also minimizes the travel times.

Since the combination and matching possibilities depend on each other, two models are considered. When two bookings to the destination are combined, only one vehicle is required, which means that as a result from this combination, only one vehicle is available to be matched with a Back ride. The first model deals with the combination of bookings, which is related to transport customers in one vehicle to their destination and/or transporting customers back.

Since combinations reduces both customer distances (kilometers with customers in the taxi) and vehicle distances (empty kilometers), and matching only reduces vehicle distances, the expected savings per combination are higher for combination as the expected savings per
match for matching, therefore the combination model is executed first and its output is used in the matching model, which is executed after combination.

The reduced distance is the difference between the sum of the distance of the two bookings when the bookings are not combined minus the total distance when the rides are combined. Obviously, combining rides only makes sense when this difference is positive. The result of combination is a list of rides To the destination and a list of rides Back.

In the second model, which is applied after the combination step, rides are matched. To do this, the combination databases for both To and Back rides must be used and corrected for rides which are combined.

The following assumptions are used for this model:

1. Both the vehicle and vehicle driver must to return to the same depot as where it started;
2. There are always enough vehicles and drivers available at any carrier;
3. Travel times are set in such a way that these are always met;
4. Delays of arrival/departure times are left out of scope;
5. A carrier will take care for a complete drive;
6. Carriers only execute rides related to this dataset, not having other rides scheduled in between these rides;
7. It is assumed that a vehicle is ready to start a new ride directly at the time of arrival;
8. Carriers are located at a single location;
9. Break times for vehicle drivers are neglected;
10. Costs are equal at each carrier.

Assumptions 1 and 5 are added as a constraint in the model. Depending on the context, assumption 2 can hold in reality. Assumption 3 and 4 might result in delays or requiring using additional vehicles. Since it could be the case that delays are known hours in advance, the schedule can be changed before a ride starts. Assumption 6 does not affect the results of the model. Assumption 7 might require a change in the arrival and/or departing time, which should be modified with the required time. This could be added in the loading and unloading times or in a similar variable. Assumption 8 could be solved by considering the different locations as different depots. Assumption 9 can be accounted for by shifting the arrival and departure times. Assumption 10 could be solved by multiplying the distances/savings with a correction factor; however it is not likely that the cost differences between carriers are big.

Paragraph 5.3.1 gives an explanation about how the savings are calculated. In paragraph 5.3.2 models for combination and matching to optimize the savings will be presented. Paragraph 5.3.3 describes the output of the model.

5.3.1. Calculating the savings

The model in section 5.2 can be used to find the most suitable values for the parameters \( m_i, f_i, \xi_i \) and \( \eta \). The savings will be optimized in the model in this paragraph. First the savings are expressed in paragraph 5.3.1.1 for combination and in paragraph 5.3.1.2 for matching.

5.3.1.1. Combination savings

For combination, the savings can be split up in customer savings and vehicle savings and are presented in kilometers. To calculate the customer savings, for each allowed setting described
in paragraph 5.2.2.1, the savings must be calculated and the largest value must be selected. The customer savings involve the savings of kilometers in which customers are in the vehicle. The vehicle savings are the reductions of kilometers in which no customers are transported (the empty kilometers).

Rides can be executed by several carriers. For each ride, the most optimal carrier must be selected to execute the ride. This can be done by calculating the vehicle distance of a ride for all available carriers and selecting the minimum. Therefore, in addition to the distances from the customers to the destinations \((d_{i,a_i}, d_{i,j} \text{ and } d_{i,a_j})\), the distances \(d_{i,k}, d_{a_i,k}, d_{j,k} \text{ and } d_{a_j,k}\) should be considered in which \(k\) represents the carrier. The customer savings are not affected by the carrier selection, the carrier selection only affects distances before and after the ride.

In no combined settings, each booking is executed by a different vehicle. In combined settings, this distance is reduced, since then only one vehicle is required to accomplish two bookings. Instead of travelling empty to/back from both customers and to/back from the destination, the vehicle only travels empty to/back from one of the customers and one of the destinations. Like the customer savings, the vehicle savings depend on the setting.

To summarize, the customer savings for combination are equal to \(d_{i,a_i} + d_{j,a_j} - d_{i,j} - d_{a_i,a_j}\) minus the distance from one of the customers to one of the destinations (this depends on the setting). The vehicle savings are equal to \(d_{k,i} + d_{k,a_i} + d_{l,j} + d_{l,a_j}\) minus the vehicle distance in the combined setting, executed by carrier \(m\). Here \(k, l\) and \(m\) represents the most optimal carrier to execute booking \(i\), booking \(j\) and the combination of booking \(i\) and \(j\) respectively. It could be the case that \(k = l, k = m\) and/or \(l = m\). The savings on customer distances and vehicle distances for each setting are given in Appendix 6a (Table 20).

When all possible savings are calculated, the most optimal allowed setting must be selected for each booking. A booking can only be combined with one other booking.

5.3.1.2. Matching savings

Like in the combination model, the goal of the matching model is to give an indication of the savings. In this model only vehicle kilometers are saved since the matching procedure does not affect customer kilometers. While rides are not matched, a taxi drives empty to the customer, brings him to the destination and drives back to the depot empty. A second vehicle drives to the destination of the second customer, brings him to his location and drives empty back to the depot. If these rides are matched, only the distances between the customers and depot are driven empty, plus the distance between the destinations of the customers (which could be the same). The realized savings in a match are:

\[
(d_k + d_{a_i,k}) + (d_{i,a_j} + d_{j,l}) - (d_{m,i} + d_{m,j}) - d_{a_j,a_j}
\]

The values of the terms for each setting are given in Appendix 6b (Table 21). The terms between the left brackets are the vehicle distances of the To ride in the not matched setting, the terms between the middle brackets are the vehicle distances of the Back ride in the not matched setting and the terms between the right brackets are the vehicle distances of the matched setting. The last term is the distance between the end location of the To ride and the starting location of the Back ride, which is 0 when these are the same. In the next paragraph, models will be described in which these savings are maximized for both Combination and Matching.
5.3.2. The savings optimization model

To use the list of bookings, these must be sorted in such a way that bookings which are likely to be combined are close to each other in the list. By doing so, rides which are likely to be combined are above or below each other. The sorting is method is similar as in section 5.2:

1. Postcode of the customer;
2. Postcode of the destination;
3. Date and Time;
4. Class of the booking;

Like before, the data is split in a list with bookings To the destination and a list with Back bookings. Data of customers who did not allow combined rides must be taken into account as well since these bookings can be used in the matching procedure. To maximize the savings again two models are used: one model for combination in paragraph 5.3.2.1 and a model for matching in paragraph 5.3.2.2. Microsoft Excel is used to build these models.

5.3.2.1. Combination model to calculate savings

This model builds on the model presented in section 5.2. As mentioned in paragraph 5.2.2.1, there are multiple settings in which two bookings can be combined (AB12, AB21, BA12 and BA21), the savings are calculated for each setting and the most optimal allowed setting will be selected. Bookings can be combined with at most 1 other bookings.

The model starts with the first booking \(i = 1\) of the list and calculates all distances and speed averages depending on these distances related to booking \((i + 1, i + 2, ... i + z)\), in which \(z\) reflects the number of bookings to which bookings are compared. In the model of 5.2, the value of \(z = 1\) is used, since that model was built to find how many bookings could be combined. While optimizing the savings, it is be better to consider more alternatives, since for example it is better to combine booking \(i\) with \(i + 2\) instead of booking \(i + 1\). However, the more alternatives are compared to a booking, the slower the model will work, so the value of \(z\) should be limited.

Since bookings are sorted on postcode and time, and only a limited number of bookings have the same arrival / departure time, bookings which are not close to each other in the list cannot be combined, so comparing a many bookings will not result in lots of additional combination possibilities.

To calculate the savings a First Come First Serve strategy is used. This means that first for booking \(i\) the most optimal value is selected (considering the next \(z\) bookings only). As a result, the solution is likely to be not optimal, since not all possibilities are compared. In section 5.4, a linear programming model will be presented which can compare all possibilities and results in an optimal solution. If no combinations are allowed for a booking, the savings are set to 0 and no combination takes place for that booking.

The model starts with booking \(i\) and savings will be calculated for the situation in which a booking is compared with \(i = i + 1, i + 2, ..., i + z\). If a combination is made (for example with \(i = i + 2\)), the booking which is combined with booking \(i\) \((i + 2)\), will be skipped in the execution since at most two bookings are combined, so booking \(i\) is only compared with other bookings if it has not been selected while \(i - z, ..., i - 2\) and \(i - 1\) were considered and if for example booking \(i + 2\) was combined with booking \(i\), booking \(i + 1\) will only be compared with booking \(i + 3, ..., i + z\), skipping booking \(i + 2\). The distance between the two bookings, the distance
between the bookings and the destinations, the distance between the two destinations and distances to the depot of the listed carriers of the customers and destinations are calculated. Next, the model checks whether all constraints of paragraph 5.2.2.1 are met: it is checked whether time matches, the maximum absolute additional travel time and fraction are not exceeded, capacity does not exceed the maximum values and the classes of the bookings are such that they allow combination. In the next step, for each possible combination the most optimal setting and the most optimal carrier for that setting is selected. Then the possible combinations are compared and the most optimal one is selected. The model then continues with the next booking (skipping bookings which are already combined) and does the same calculations till the last booking of the list is reached. The result is a list of rides in which bookings are combined. In Appendix 7c, this model is described in mathematical formulation when the number of compared bookings \( z = 3 \).

One value which is a result of this model is the minimum slack time of a ride. This is the difference between the sum of the allowed waiting and additional travel time minus the additional waiting and travel time as a results of combination. The minimum of the slack time for combined bookings will be selected, since the values must be met for both customers. This value will be used as input value in the second step, the matching model. For To rides the value is as follows:

\[
p_1 = \min \left[ \xi_i + m_i - (t_{\text{comb},i,j} - t_{i,a_i} - l_i - u_i) + (a t_i - \min (a t_i, a t_j)), \xi_j + m_j - (t_{\text{comb},i,j} - t_{j,a_j} - l_j - u_j) + (a t_j - \min (a t_i, a t_j)) \right]
\]

For back rides, the formula equals:

\[
\rho_j = \min \left[ \xi_i + m_i - (t_{\text{comb},i,j} - t_{i,a_i}) + (\max (d t_i, d t_j) - d t_i), \xi_j + m_j - (t_{\text{comb},i,j} - t_{j,a_j}) + (\max (d t_i, d t_j) - d t_j) \right]
\]

Here, \( t_{\text{comb},i,j} \) is the total required time, including load and unload time, in which customer \( i \) is in the vehicle when there is a combination with customer \( j \). \( t_{i,a_i} \) is the time in the not combined setting excluding load and unload times. \( a t_j \) is the arrival time of booking \( i \) (for To rides) and \( d t_i \) is the departure time of booking \( i \) (for Back rides). If a booking is not combined, this will result in \( \xi_i + m_i \), in a combined ride the minimum slack time is the minimum value of this formula from customers \( i \) and \( j \), denoted by \( \rho_j \). If a customer does not allow combination, the slack time is zero instead of the function above (in fact, customers who do not allow customers can be seen as customers with \( m_i = \xi_i = 0 \)).

### 5.3.2.2. Matching model to calculate savings

After executing the combination model of the previous section, the matching model must be executed. The model uses the To rides list and looks for related Back rides. The output of the model of 5.3.1.1 is used, which means that bookings which are combined in that model are listed as one value. The list is divided in two Classes: Class 1 equals the list with bookings which allow combination, starting at row 1, and Class 2 is the part of the list which does not allow combination, starting at the first booking of the list which does not allow combination. So, the To rides database is split in two groups as well as the Back rides database.
The model starts at the first ride of the To rides list and looks for the first booking in the Back rides list of Class 1 for which the travel time match exactly. The distance between the end location of the To ride and the starting location of the Back ride is calculated and it is checked whether the time to travel this distance does not exceed the minimum slack time left of the customers of the Back ride. If this is the case, the savings are calculated.

In the next step, the booking just below the just compared Back ride is checked with the To ride. If the travel time matches again the distance between the destinations is checked with the slack time and savings are calculated, moving to the next booking. If the travel times do not match, a booking is searched in Class 2. This process continues till three bookings are found, or when there are no more matches possible. In addition, the model can compares bookings from the To list with bookings from the Back list departing 0.5 hours later, or compare bookings from the To rides minus 0.5 hours with the Back rides list. The last option results in an additional waiting time for the customer(s), so this can only be used if this is allowed. These options are subsets of the possibilities with $\eta = 0.5$. The model cannot deal with values of $\eta > 0.5$.

Since the number of alternatives to match is limited and the First Come First Service policy is used (first the optimal solution is searched for the first ride, which can limit the results of the second ride), this will not result in optimal solution. Also, this model only compares a limited amount of the possibilities. A mathematical model is described in Appendix 7d.

In the next section (5.4), an alternative model will be described. This is a linear programming model, with the characteristics of a Vehicle Routing Problem and Pickup and Delivery Problem. Results and recommendations based on the data of Tinker are presented in chapter 6 and 7.

5.4. The linear programming model

In this section a Linear Programming model will be presented. AIMMS has been used to implement and test the model. The linear programming model will lead to better solution as the simple Excel model of 5.3, since it compares all possible combinations instead of a limited amount of possibilities. Like the previous model, this model consists of two sub models as well.

As before, bookings are combined first and based on the results, rides are matched in the second step. In paragraph 5.4.1 the combination problem is given, paragraph 5.4.2. describes the matching model. Output is described in section 5.5.

5.4.1. Combination problem

First, the combination model will be presented. The model maximizes the total distance savings. The maximum number of combinations per customer is 1 since an assumption has been made that at most two bookings can be combined. Two bookings can only be combined if:

- The direction is the same (realized by separating the data set in a list of To rides and a list of Back rides);
- The date and time of the bookings differ at most the maximal additional waiting time;
- Both bookings allow combinations (done by leaving bookings who do not allow combinations out of scope);
- Luggage does not exceed the maximum allowed quantity;
- Number of passengers does not exceed the maximum allowed quantity;
- The additional absolute travel time for the customers is at most the maximum allowed quantity;
- The additional travel fraction is at most the maximum allowed quantity.
In mathematical terms, the model is as follows:

Maximize: \( \sum_i \sum_j x_{i,j} s_{i,j} \)

Where

\[
\sum_i x_{i,j} + \sum_j x_{i,j} \leq 1, \quad \forall i, j: i \neq j
\]

\[
x_{i,j} = 1 \text{ if} \sum_i x_{i,j} + \sum_j x_{i,j} = 0, \quad \forall i, j: i \neq j
\]

\[
x_{i,j} \leq \lambda_{i,j}
\]

\[
\lambda_{i,j} = 1 \text{ if}!
\]

\[
\begin{aligned}
& \{ \begin{aligned}
& at_i - at_j \leq \xi_i, \quad at_j - at_i \leq \xi_j, \\
& dt_j - dt_i \leq \xi_i, \quad dt_i - dt_j \leq \xi_j,
\end{aligned} \\
& \text{to rides data}
\end{aligned}
\]

\[
LU_i + LU_j \leq LU_{\text{max}}
\]

\[
pax_i + pax_j \leq pax_{\text{max}}
\]

\[
c_i = c_j = 1
\]

\[
t_{\text{comb,i,i,j}} \leq t_{i,a_i} + m_i
\]

\[
\frac{t_{\text{comb,i,i,j}}}{t_{i,a_i}} \leq 1 + f_i
\]

\[
s_{i,j} = cd_{i,j} + vd_{i,j}
\]

\[
x_{i,j}, \lambda_{i,j} \text{ binary}
\]

\[
LU_i, pax_i \in \mathbb{R}
\]

\[
\xi_i \geq 0
\]

Where:

\( s_{i,j} \): Savings in kilometers when customer \( i \) and \( j \) are combined, which is the difference of the distance in the not combined setting and the combined setting for the most optimal setting for both customer distances (\( cd_{i,j} \)) and vehicle distances (\( vd_{i,j} \)). The formulas per setting are given in Appendix 6.

\( cd_{i,j} \): Difference in kilometers with customers in the vehicle between the original situation and combined setting. The values per setting are given in Appendix 6a, Table 20.

\( vd_{i,j} \): Difference in empty kilometers between the original situation and combined setting. The values per setting are given in Appendix 6a, Table 20.

\( c_i \): A binary variable indicating whether combination is allowed for the booking or not.

\( t_{i,a_i} \): Time to execute the original (not combined) ride from customer \( i \) to airport \( a_i \) or the other way around.

\( t_{\text{comb,i,i,j}} \): The total time with customer \( i \) in the taxi when its ride is combined with customer \( j \) (as given in Appendix 5).

\( \lambda_{i,j} \): A binary variable which indicates whether booking \( i \) and \( j \) could be combined.

\( x_{i,j} \): A binary variable which determines whether customer \( i \) and \( j \) will be combined in one ride, in which customer \( i \) will be the customer not facing additional travel time.
\( dt_i \) is the date and time that booking \( i \) starts the ride, presented in the format \( \text{dd-mm-yyyy hh:mm} \) (day-month-year hour: minute).

\( at_i \) is the date and time that booking \( i \) ends the ride, presented in the format \( \text{dd-mm-yyyy hh:mm} \) (day-month-year hour: minute). This is the arriving time at the destination for To rides.

\( \xi_i \): Maximum allowed waiting time for customer \( i \).

\( m_i \): Maximum additional travelling distance/time in hours for customer \( i \).

\( f_i \): Maximum additional fraction in relation to the current travelling distance/time for customer \( i \).

\( d_{x,y} \): Distance between location \( x \) and \( y \). Indices \( i,j \) represents customers, \( k,l,m \) represents carriers and \( a_i, a_j \) destinations. Values are calculated by using the formula in paragraph 5.2.1.

The value of \( \lambda \) determines whether two bookings \( (i,j) \) could be combined. The value of \( x_{i,j} \) indicates whether bookings \( i \) and \( j \) will be combined, which is only possible if \( \lambda_{i,j} = 1 \).

**5.4.2. Matching problem**

Like in the combination model, distance savings are maximized. The results of the combination model are used as input of this model and are respected, meaning that the constraints for single bookings are checked already and combinations remain intact. Each To ride can be matched with at most one Back ride. Rides can only be matched if:

1. The difference between the arrival time of the To ride (plus the travel time between the To ride end location and start location of the Back ride) and the departure time of the Back ride is at most the maximum allowed waiting time for the vehicle driver;
2. The real arrival time of the To ride is at most the minimum slack time of the To rides earlier as the original planned arrival time;
3. The real departure time is at most the minimum slack time of the Back rides later as the original planned departure time.

In contradiction to the Excel model of 5.2, this model can match rides from different airports. The time to travel between the airports is not considered as waiting time, so the arrival and depart time may differ at most \( \eta + t_a \). The maximal allowed working time for a vehicle driver \( (t_{\text{max}}) \) is set in a law and must be respected (Ministerie van Sociale Zaken en Werkgelegenheid, 2014). The most important characteristics of this law are presented in Appendix 8. For small distances only it is not necessary to plan a break during a ride. By considering only one match, the maximum working time is in that case not exceeded. Even though the vehicle driver also has additional tasks during a shift like refilling gas for example, the maximum working time is not likely to be exceeded. It will be assumed that vehicle drivers are allocated in such a way that this will not be a problem. Break time might be an issue, but it is assumed that the effect can be neglected.

The selected taxi type depends on the maximum of the luggage and passengers of the single rides. If at least one requires a bus, the bus will be used. If both can be done with a car, the car is selected. The used output of this problem is similar to the output of paragraph 5.2.3.
In mathematical formulation, the model for the matching procedure is defined as follows. The second constraint does not have to be added, since this will be realized by the definition of the first constraint.

Maximize: \[ \sum \sum X_{i,j} r_{i,j} \]

Where

\[ \sum_i X_{i,j} + \sum_j X_{i,j} \leq 1, \quad \forall i, j: i \neq j \]

\[ x_{i,j} = 1 \text{ if } \sum_i X_{i,j} + \sum_j X_{i,j} = 0, \quad \forall i, j: i \neq j \]

\[ X_{i,j} \leq \mu_{i,j} \]

\[ \mu_{i,j} = 1 \text{ if }! \]

\[ \begin{cases} 
  dt_j - (at_i + t_a) \leq \eta \\
  dt_j + \rho_j - (at_i + t_a) \leq \eta \\
  dt_j - (at_i - \rho_i + t_a) \leq \eta \\
  dt_j + \rho_j - (at_i - \rho_i + t_a) \leq \eta 
\end{cases} \quad \text{(one of these must hold)} \]

\[ at_e + t_a - dt_j \leq \rho_j \]

\[ r_{i,j} = d_i + d_j - d_{match,i,j} - d_a \]

\[ X_{i,j}, \mu_{i,j} \text{ binary} \]

\[ \eta \geq 0 \]

The variables are as follows:

- \( at_i \): The date and time that ride \( i \) is planned to arrive at the destination presented in the format dd-mm-yyyy hh:mm (day-month-year hour: minute).
- \( d_i \): Total distance of ride \( i \), including empty kilometers to get to the customer/destination.
- \( d_{match,i,j} \): Total distance of ride \( i \) and \( j \) together when these are matched, including empty kilometers.
- \( dt_i \): Is the starting date and time of ride \( i \) presented in the format dd-mm-yyyy hh:mm (day-month-year hour: minute).
- \( \eta \): Maximum allowed waiting time for taxis to wait at the destination, given in hours.
- \( \rho_j \): The minimum slack time of the customers related to ride \( j \), which is the minimum value of the slack times of the customer involved to ride \( j \)
- \( r_{i,j} \): Savings in kilometers when ride \( i \) and \( j \) are matched.
- \( t_a \): Is the total required time to drive from the end location of ride \( i \) to the start location of ride \( j \).
- \( \mu_{i,j} \): A binary variable which indicates whether (combined) ride \( i \) and \( j \) can be matched.
- \( X_{i,j} \): A binary variable which determines whether ride \( i \) and \( j \) are matched.

The variable \( \mu_{i,j} \) is comparable to the \( \lambda_{i,j} \) for the combination model. For notational reasons \( i \) and \( j \) are used for the rides, combined bookings in the combination model are considered as one ride. In the combination problem a list of bookings (indicated with lowercase alphabets) is
considered, the results of that model is a list of rides (indicated with capitals) which are used in this matching model. The result of the matching model is a list of drives. 

In the next and final section of this chapter, output values of the models of section 5.2, 5.3 and 5.4 will be discussed.

**5.5. Output of the models**

When the models are executed, output is available. The values can be calculated for both the simple Excel models and the Linear Programming models. First the output related to the Combination models is discussed in paragraph 5.5.1, output related to Matching is presented in paragraph 5.5.2.

**5.5.1. Output of the combination model**

In this section output of the combination model is defined. The most important output value is the total savings ($s$). The total savings are as follows:

$$s = \sum_{i,j} x_{i,j}s_{i,j} = \sum_{i,j} x_{i,j}(cd_{i,j} + vd_{i,j})$$

The fraction of the total savings can be found by dividing this value by the distance in the case that no bookings are combined. The total savings exist of savings on customer distances ($cd$) and savings on vehicle distances ($vd$), formulas for these values can be found in Appendix 6.

In addition to the savings, other values are set. The extra time, involving both additional waiting time and additional travel time, is an important measure. In mathematical notation, the extra travel time of customer $i$ for combination $i,j$ equals:

$$ET_{i,i,j} = \begin{cases} 
(t_{comb,i,i,j} - t_{i,a_i} - l_i - u_i) + (at_i - \min(at_i,at_j)), & \text{to} \\
(t_{comb,i,i,j} - t_{a_i,l} - l_i - u_i) + (\max(0,dt_i - \max(dt_i,dt_j))), & \text{back} 
\end{cases}$$

If a ride is not combined with another booking, $x_{i,i} = 1$ and $t_{comb}$ is set to $t_{i,a_i} + l_i + u_i$, this results in a zero in the Extra Time function. $l_i$ and $u_i$ are the values of load and unload times. The extra time related to both combined customers is defined as $ET_{i,j} = ET_{i,i,j} + ET_{j,i,j}$.

The maximum of these values, the total additional travel time, number of customers facing disadvantages and the average waiting time of these customers can be expressed as follows:

$$\max_{i,j} (ET_{i,i,j} + ET_{j,i,j})$$

$$\sum_{i,j} (ET_{i,i,j} + ET_{j,i,j})$$

$$\sum_{i,j} x_{i,j} | ET_{i,i,j} + ET_{j,i,j} > 0$$

$$\frac{\sum_{i,j} (ET_{i,i,j} + ET_{j,i,j})}{\sum_{i,j} x_{i,j} | ET_{i,i,j} + ET_{j,i,j} > 0}$$
The number of possible combinations is represented as follows

\[ \sum_{i,j} \lambda_{i,j} \]

\( \lambda_{i,j} \) indicates whether booking \( i \) and \( j \) could have been combined. The average number of combinations per booking is as follows:

\[ \frac{\left( \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_{i,j} \right) - n}{n} \]

The value of \( n \) represents the number of bookings in the dataset.

5.5.2. Output of the matching model

For the matching procedure, some output can be considered as well. The total savings \( r \) equal:

\[ r = \sum_{i,j} X_{i,j} r_{i,j} = \sum_{i,j} X_{i,j} v d_{i,j} \]

The number of possible matches is as follows, in which \( \mu_{i,j} \) indicates whether ride \( i \) and \( j \) could have been matched:

\[ \sum_{i} \sum_{j} \mu_{i,j} \]

The average number of possible matches per ride (for To or Back rides) can be found by dividing this value by the number of To or Back rides.

\[ \frac{\sum_{i} \sum_{j} \mu_{i,j}}{n_{to}}, \quad \frac{\sum_{i} \sum_{j} \mu_{i,j}}{n_{back}} \]

A different output variable is the number of rides which could be matched with at least one other ride. This equals the following:

\[ \sum_{i} \mid \sum_{j} \mu_{i,j} \geq 1, \quad \sum_{j} \mid \sum_{i} \mu_{i,j} \geq 1 \]

To calculate the fraction of this, this value must be divided by \( n \).

The models of this chapter are applied to the situation of Tinker, in the next chapter the results of these models will be discussed.
6. Case Study
The models of chapter 5 are applied to the situation of Tinker. In this section the results of these models will be presented. The used data of Tinker will be introduced in section 6.1. In section 6.2 is described how many rides could have been combined, considering different values for the maximum allowed (fraction) of travel time and maximum allowed waiting time. This is done by using the Excel model presented in section 5.2.
Section 6.3 gives an estimation of the savings, using the simple Excel model described in section 5.3 which does not consider all possible combinations. In section 6.4, the results of the linear programming model of section 5.4 are presented.

6.1. Input data of Tinker
To answer the research questions, data with a travel date between December 2013 and May 2014 are used, since Tinker did not use different classes (allowing customers to choose to have their rides combined or not) before November and the dataset of November was not reliable. This dataset consisted of 10.842 bookings (5.936 To rides, 4.906 Back rides). 6.483 (3.610 To rides, 2.873 Back rides) of these have Economic Class. In the combination models, first only Economic Class bookings will be used. Second, all bookings will be considered, ignoring class constraints. This is done to get an idea of the effect of a higher number of bookings per day, which is expected by Tinker since the number of bookings is growing fast.
The distance calculation will be based on the coordinates of the Tinker database. For each booking, Tinker stored the latitude and longitude coordinates. For example, the coordinates of Schiphol are 4.750 (lat) and 52.300 (long). The values need to be divided by 1.000 to get the correct distances. These bookings are all related to bookings in or close to the Netherlands. In this area, the scaling factor, $\sigma = 88$ and the road density factor $\tau = 1.34$ (Broekmeulen, 2011).

To determine a formula for the speed, data of the distance and meters and the travel time (which are available for executed bookings in the past) are compared to get a formula. Since these values are only available for distances and times between customers and destinations, a formula is required to predict the speed for other values. The formula is derived using regression software (using SPSS or Excel), the used logarithmic formula is $17.47 \times \ln(x) + 12,852$ in which $x$ is the distance in kilometers. This formula has a $R^2$ of 0.78, which is high. Alternatively an exponential or polynomial function could be used, but these result in a lower $R^2$ and have high deviations for distances above 200 kilometers, as shown in Figure 26 of Appendix 13.
If the values are in general too high the formula results in exceeding the maximum additional absolute travel time or fraction (leading to unsatisfied customers), and if the values are lower as in reality, rides which could have been combined are not combined (resulting in fewer savings). The logarithmic function has relatively good results for all values of distance. In reality, travel speed differs as well as a result of external factors like traffic, weather, behavior of taxi driver, number of kilometers driven on highways and so on, it is hard to know the exact speed in advance. Route planners as Google Maps could be used as well, but this formula can be implemented in the models to make it possible to calculate the expected travel times automatically.
6.2. Changing the allowed travel and waiting times

In this section the effect of the maximum additional waiting time ($\xi_i$), maximum absolute additional travel time ($m_i$), maximum travel fraction ($f_i$) and maximum allowed waiting time for the vehicle driver ($\eta$) is investigated. In the current situation of Tinker, the maximum additional travel time is equal for all bookings and set to $m_i = m = 0.25$. The maximum travel fraction is set to $f_i = f = 0.2$ and maximum waiting time $\xi_i = \xi = 0$ for all bookings. Values are given in hours. Bookings are sorted first on postcode and finally on travel time (arrival / departure time).

First the effect of the maximum additional waiting time ($\xi$) is presented, using the model of section 5.2. This value indicates how many hours difference of arrival/departure times between bookings are allowed for combination. Since bookings only differ in half hours in the dataset steps of 0.5 are taken, $\xi = 0.5$ has the same result as $\xi = 0.99$. As mentioned before, each booking is compared with the booking on the next list. If combination is allowed, a value 1 is given, otherwise a zero. The sum of these values is the number of bookings which could be combined. Only considering time limitations and class constraints (ignoring luggage and distance constraints), the allowed combinations, considering the situation with only Economic Class bookings and the situation in which all bookings are considered, are given in Table 5. The results only depend on the value of $\xi$, $m$ and $f$ do not affect the results.

<table>
<thead>
<tr>
<th>Value of $\xi$</th>
<th>Only Economic Class Bookings</th>
<th>All Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Combinations Possible</td>
<td>Fraction</td>
</tr>
<tr>
<td>0</td>
<td>1.871</td>
<td>0.29</td>
</tr>
<tr>
<td>0.5</td>
<td>3.650</td>
<td>0.56</td>
</tr>
<tr>
<td>1</td>
<td>4.571</td>
<td>0.71</td>
</tr>
<tr>
<td>1.5</td>
<td>5.084</td>
<td>0.78</td>
</tr>
<tr>
<td>2</td>
<td>5.409</td>
<td>0.83</td>
</tr>
<tr>
<td>2.5</td>
<td>5.656</td>
<td>0.87</td>
</tr>
<tr>
<td>3</td>
<td>5.830</td>
<td>0.90</td>
</tr>
<tr>
<td>3.5</td>
<td>5.929</td>
<td>0.91</td>
</tr>
<tr>
<td>4</td>
<td>6.016</td>
<td>0.93</td>
</tr>
</tbody>
</table>

It can be found that the number of combinations based on travel time only increase a lot when the value of $\xi$ is increased. Customer satisfaction must be taken into account as well, customers are not willing to wait more than an hour. However, after 1 hour, the effect of adding another 30 minutes is smaller and therefore not attractive anymore. When more rides are included, the combinability fraction based on time increase. Therefore, the fraction of bookings which can be combined increase when there are more bookings in the system.

Secondly, the effect of changing the values of $m_i$ and $f_i$ is tested. As described before, these values are currently the same for all customers ($m_i = m = 0.25, f_i = f = 0.2$ for all customers). For the analysis it is assumed that this remains the case, since Tinker must communicate these values to the customers to keep them satisfied. Increasing $m$ and/or $f$ will result in more possibilities. To show the effect, values of 0.25, 0.5 and 1 hour for $m$ and 0.2, 0.5, 1 and $\infty$ for $f$
are considered. The results are given in Table 6. Only the constraint of exceeding maximum absolute travel time and the fraction is considered here, the value of \( \xi \) does not affect this constraint and does therefore not have to be considered. Again, the situation with only Economic Class bookings and the situation with all bookings are considered.

<table>
<thead>
<tr>
<th>Value of ( m )</th>
<th>Value of ( f )</th>
<th>Allowed Combinations</th>
<th>Fraction</th>
<th>Allowed Combinations</th>
<th>Fraction</th>
</tr>
</thead>
<tbody>
<tr>
<td>0,25</td>
<td>0,2</td>
<td>788</td>
<td>0,12</td>
<td>1.614</td>
<td>0,15</td>
</tr>
<tr>
<td>0,5</td>
<td>0,2</td>
<td>819</td>
<td>0,13</td>
<td>1.724</td>
<td>0,16</td>
</tr>
<tr>
<td>1</td>
<td>0,2</td>
<td>819</td>
<td>0,13</td>
<td>1.724</td>
<td>0,16</td>
</tr>
<tr>
<td>0,25</td>
<td>0,5</td>
<td>2.393</td>
<td>0,37</td>
<td>3.589</td>
<td>0,33</td>
</tr>
<tr>
<td>0,5</td>
<td>0,5</td>
<td>2.848</td>
<td>0,44</td>
<td>6.553</td>
<td>0,43</td>
</tr>
<tr>
<td>1</td>
<td>0,5</td>
<td>2.901</td>
<td>0,45</td>
<td>4.852</td>
<td>0,45</td>
</tr>
<tr>
<td>0,25</td>
<td>1</td>
<td>2.689</td>
<td>0,42</td>
<td>3.905</td>
<td>0,36</td>
</tr>
<tr>
<td>0,5</td>
<td>1</td>
<td>4.004</td>
<td>0,62</td>
<td>6.417</td>
<td>0,59</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>4.433</td>
<td>0,68</td>
<td>7.517</td>
<td>0,69</td>
</tr>
<tr>
<td>0,25</td>
<td>( \infty )</td>
<td>2.706</td>
<td>0,42</td>
<td>3.922</td>
<td>0,36</td>
</tr>
<tr>
<td>0,5</td>
<td>( \infty )</td>
<td>4.496</td>
<td>0,69</td>
<td>7.001</td>
<td>0,65</td>
</tr>
<tr>
<td>1</td>
<td>( \infty )</td>
<td>5.413</td>
<td>0,83</td>
<td>9.043</td>
<td>0,83</td>
</tr>
</tbody>
</table>

From Table 6, it can be found that increasing the maximum fraction additional time increases the opportunities to combine. When still promising customers a maximum additional travel time of 0,25 hours, dropping the fraction results in an increase from 788 to 2.706 for the number of possible combinations in the case of Economic Class bookings only, an increase of 343%. The results while considering all classes is similar, since this constraint does not depend on the number of bookings. Increasing the maximum absolute additional travel time has impact as well, but only in combination with making the fraction more flexible. The larger the value of \( f \), the larger the effect, leading to more opportunities to combine.

The constraint that the number of passengers and luggage cannot exceed the maximum value must be checked as well. In the current setting, \( \text{LU}(\text{max}) = 32 \) and \( \text{pax}(\text{max}) = 8 \), these values may not be exceeded. In the model there is a difference made between the two taxi types. If the value of \( \text{LU} \) does not exceed 18 and the number of passengers does not exceed 4, a small taxi car can be used. Based on the data, capacity has a small effect, in 5.129 of the 6.483 cases (79%) for Economic Class only, combination is possible when only capacity is considered. 2.495 (49%) of these rides requires a large bus. For all classes, 9.103 (84%) can be combined considering capacity only. 4.445 (49%) of these require a bus.

Next, for several combinations of \( f, m \) and \( \xi \) the number of possible combinations and the fractions (matching all described constraints, except class constraint for the list of all bookings) are given in Table 7. In Appendix 12, the results are given for even more parameter settings. Compared to the previous analysis smaller only values of \( m \) of 0,25 and 0,5 are taken since these are more interesting with respect to customer satisfaction. Values are in hours.
Table 7: Combining Probabilities for different settings

<table>
<thead>
<tr>
<th>$f$</th>
<th>$m$</th>
<th>$\xi$</th>
<th>Economic Class Only</th>
<th>All Bookings</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Number of Combinations</td>
<td>Fraction</td>
</tr>
<tr>
<td>0,2</td>
<td>0,25</td>
<td>0</td>
<td>155</td>
<td>0,02</td>
</tr>
<tr>
<td>0,2</td>
<td>0,5</td>
<td>0</td>
<td>160</td>
<td>0,02</td>
</tr>
<tr>
<td>0,5</td>
<td>0,25</td>
<td>0</td>
<td>576</td>
<td>0,09</td>
</tr>
<tr>
<td>0,5</td>
<td>0,5</td>
<td>0</td>
<td>669</td>
<td>0,10</td>
</tr>
<tr>
<td>1</td>
<td>0,5</td>
<td>0</td>
<td>926</td>
<td>0,14</td>
</tr>
<tr>
<td>0,2</td>
<td>0,25</td>
<td>0,5</td>
<td>301</td>
<td>0,05</td>
</tr>
<tr>
<td>0,2</td>
<td>0,5</td>
<td>0,5</td>
<td>312</td>
<td>0,05</td>
</tr>
<tr>
<td>0,5</td>
<td>0,25</td>
<td>0,5</td>
<td>1.108</td>
<td>0,17</td>
</tr>
<tr>
<td>1</td>
<td>0,25</td>
<td>0,5</td>
<td>1.253</td>
<td>0,19</td>
</tr>
<tr>
<td>1</td>
<td>0,5</td>
<td>0,5</td>
<td>1.794</td>
<td>0,28</td>
</tr>
<tr>
<td>$\infty$</td>
<td>0,25</td>
<td>0</td>
<td>655</td>
<td>0,10</td>
</tr>
</tbody>
</table>

In the current situation, 2% of the bookings can be combined, which is not much. When more bookings are added, the fractions increase with a few percentages for the current setting with $m = 0,25, f = 0,2$ and $\xi = 0$. For higher values of $m, f$ and $\xi$, this increase is higher. It can be concluded that when more bookings are in the system the combination possibilities increase as well, for any setting the values are around 5% higher in the situation with more bookings.

Alternatively, a mixed setting can be used. Since bookings with a small distance are limited by $f$ and bookings with a longer distance are limited by $m$, considering only a maximum absolute additional travel time ($m = m_1$) for small distances and the maximum additional travel fraction ($f$) and a larger maximum absolute additional travel time ($m = m_2$) for large distances probably leads to more possible combinations. So bookings with a small distance to the destination (travel time less than $\frac{m_2}{f}$) are only limited by $m_1$ (setting $f$ to $\infty$), and large rides are limited by $m_2 (\geq m_1)$ and $f$. Settings will be denoted by ($f, m_1, m_2, \xi$). The ($0,2; 0,25; 0,5, 0$) settings leads to 659 combinations, which is 4,25 times as much as the 155 combinations in the current ($0,2; 0,25; 0$) setting given in Table 7. Adding an allowance of 0,5 hours additional waiting time ($\xi = 0,5$) to customers almost doubles the number of possible combinations. Expanding this to an hour even further increases the opportunity. The effect of adding more bookings is similar as in Table 7. Results for other mixed settings are given in Appendix 12 in Table 26.

Finally the effect of the allowed waiting time for the taxis ($\eta$) at the destination is tested. To test the effect of $\eta$, the original values of $m, f$ and $\xi$ are considered. In Table 8, the effect of $\eta$ will be presented. Rides of both classes (Economic / Comfort) are used for matching. Two types of output are given, the number of possible matches (the sum of the possible matches per ride) which is the same for To and Back rides and the total rides which could be matched (which is different for To and Back rides). For each of these values, the average per To / Back ride are given. For similar reasons as for $\xi$, steps of 0,5 are taken for $\eta$. Only rides with a similar destination are matched in this model, which is a limitation of this model.
Table 8: Effect of \( \eta \)

<table>
<thead>
<tr>
<th>Value of ( \eta )</th>
<th>Matches Possible</th>
<th>Average</th>
<th>Number Rides Matchable</th>
<th>Fraction</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3.268</td>
<td>0.60 / 0.70</td>
<td>2.159 / 1.705</td>
<td>0.40 / 0.37</td>
</tr>
<tr>
<td>0.5</td>
<td>4.778</td>
<td>0.82 / 0.98</td>
<td>2.804 / 2.055</td>
<td>0.48 / 0.42</td>
</tr>
<tr>
<td>1</td>
<td>6.810</td>
<td>1.17 / 1.40</td>
<td>3.396 / 2.370</td>
<td>0.48 / 0.49</td>
</tr>
<tr>
<td>1.5</td>
<td>9.049</td>
<td>1.55 / 1.86</td>
<td>3.911 / 2.622</td>
<td>0.57 / 0.54</td>
</tr>
<tr>
<td>2</td>
<td>11.291</td>
<td>1.93 / 2.33</td>
<td>4.243 / 2.818</td>
<td>0.73 / 0.58</td>
</tr>
</tbody>
</table>

When more combinations are possible, there are fewer rides to the destination and back so the number of options to match rides will decrease. When the maximum allowed additional travel time and waiting time are larger, the number of possible matches increases. Large values for \( \eta \) are unrealistic since waiting taxi drivers cost money. So only small waiting times (\( \eta = 0, \eta = 0.5 \) or \( \eta = 1 \)) are appropriate. It can be concluded from Table 8 that increasing \( \eta \) has a strong effect on the number of possible matched.

In this section, the influence of the parameters \( m, f, \xi \) and \( \eta \) was discussed. In the next section, for various settings of the parameters savings will be calculated.

### 6.3. Savings estimations

First the results will be presented for the combination model in paragraph 6.3.1. After that, results for the matching model are discussed in paragraph 6.3.2 are given. The savings will be calculated for the situation in which a booking is compared with 6 other bookings (the value of \( \lambda \) described in section 5.3 has value 3, booking \( i \) is compared with booking \( i - 3, i - 2, i - 1 \) and with booking \( i + 1, i + 2 \) and \( i + 3 \)). This model was built step by step.

#### 6.3.1. Combining Bookings

The combination model described in section 5.3.2.1. can be used to get an overview of the performance under specified input settings. The savings calculations are based on all rides (taking into account both Comfort and Economic Class bookings), but Comfort Class bookings are not combined. The savings for a couple of parameter settings are given in Table 9.

Table 9: Savings fraction per Setting, first Economic Class only and secondly all rides

<table>
<thead>
<tr>
<th>( f )</th>
<th>( m )</th>
<th>( \xi )</th>
<th>Customer Savings Fraction</th>
<th>Vehicle Savings Fraction</th>
<th>Total Savings Fraction</th>
<th>Total Savings in Kilometers</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>0.25</td>
<td>0</td>
<td>0.8%</td>
<td>1.3%</td>
<td>1.1%</td>
<td>12.689</td>
</tr>
<tr>
<td>0.2</td>
<td>0.5</td>
<td>0</td>
<td>0.8%</td>
<td>1.3%</td>
<td>1.1%</td>
<td>13.105</td>
</tr>
<tr>
<td>0.5</td>
<td>0.25</td>
<td>0</td>
<td>1.7%</td>
<td>3.6%</td>
<td>2.9%</td>
<td>32.967</td>
</tr>
<tr>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
<td>1.7%</td>
<td>4.2%</td>
<td>3.2%</td>
<td>36.469</td>
</tr>
<tr>
<td>1</td>
<td>0.5</td>
<td>0</td>
<td>1.4%</td>
<td>5.4%</td>
<td>3.8%</td>
<td>43.569</td>
</tr>
<tr>
<td>0.2</td>
<td>0.25</td>
<td>0.5</td>
<td>1.6%</td>
<td>2.6%</td>
<td>2.2%</td>
<td>25.136</td>
</tr>
<tr>
<td>1</td>
<td>0.5</td>
<td>0.5</td>
<td>2.6%</td>
<td>9.5%</td>
<td>6.7%</td>
<td>76.982</td>
</tr>
<tr>
<td>( \infty )</td>
<td>0.25</td>
<td>0</td>
<td>1.7%</td>
<td>4.1%</td>
<td>3.1%</td>
<td>35.916</td>
</tr>
<tr>
<td>( \infty )</td>
<td>0.5</td>
<td>0</td>
<td>1.1%</td>
<td>6.0%</td>
<td>4.0%</td>
<td>46.112</td>
</tr>
<tr>
<td>( \infty )</td>
<td>0.25</td>
<td>0.5</td>
<td>3.2%</td>
<td>7.4%</td>
<td>5.6%</td>
<td>65.232</td>
</tr>
<tr>
<td>( \infty )</td>
<td>0.5</td>
<td>0</td>
<td>2.0%</td>
<td>10.4%</td>
<td>6.9%</td>
<td>80.252</td>
</tr>
</tbody>
</table>
The total distance if none rides are combined would have been 1,155,843 kilometers (475,724 customer distances, 680,118 vehicle distances). This is the sum of the distances from the customers to their airports plus the distance from the customer to the carrier plus from the carrier to the airport. It is assumed that the most optimal carrier is selected. The savings are split in customer savings (kilometers with customers in the taxi) and vehicle savings (the empty kilometers before or after the ride).

In the current settings, the savings by combination are only 1.1% on all rides (12,689 kilometers). Even though this is not a huge reduction, it is still a promising value since growth is expected. As discussed before, growth increases the combination possibilities with about 5% of percentage, resulting in an increase of the savings as well. Table 7 shows that using higher values for $f$, $m$ and $\xi$ resulted in more possible combinations. Table 10 shows that these higher values also results in more savings. Increasing the fraction to 0.5 increases the savings to 2.9%

The largest part of the savings is in the vehicle distances, the reduction of customer distances are limited for all settings. It can be found that increasing the values of $f$, $m$ and $\xi$ can be worse for the savings of customer distances, but vehicle distances are then reduced in such a way that the total savings are higher. The reason is that negative savings on customer distances are allowed, as long as the total savings of a combination are positive. Dropping the fraction $f$, $(\infty, 0.25, 0)$, result in savings of 3.1%, which is almost three times as much as the original situation. Allowing a 0.5 hours waiting time $(0.2; 0.5; 0.5)$ doubles the savings.

In the current situation, the maximum additional time for the customers is 14 minutes. In total, 155 customers face additional time, with an average time of 8 minutes. When the more flexible $(1; 0.5; 0.5)$ setting is used, meaning that the maximum additional fraction is 1, the maximum additional travel time is 0.5 hours and an additional 0.5 hours waiting for customers who have a different arrive/departing time at the airport is allowed. The maximum additional time increases to 59 minutes, 1,466 customers face the disadvantages, with an average waiting time of 31 minutes. The savings of that setting are 76,982 kilometers. For any setting, the maximum waiting time in hours will be close to $m_i + \xi_i$.

The mixed setting $(0.2; 0.25; 0.5; 0)$ results in total savings of 3.1%, a total of 36,245 kilometers. Here, the maximum waiting time for a customer is 25 minutes. In total, 602 customers will face additional travel / waiting time, with an average of 8.5 minutes. The results are close to the situation in which the fraction is dropped $(\infty, 0.25, 0)$.

As discussed before, this Excel model does not provide the optimal solution, the savings will be higher if the Linear Programming model is used. The results of that model are presented in section 6.4. In the next paragraph the simple Excel model for matching is discussed.

### 6.3.2. Matching

The results of the Matching model presented in paragraph 5.3.2.2. depend on the input and results of the Combination model. Three situations will be compared: the original situation, the flexible $(1; 0.5; 0.5)$ setting and the mixed $(0.2; 0.25; 0.5; 0)$ setting. For the results three scenarios are considered:

1. No additional waiting time for the vehicle is allowed ($\eta = 0$), To rides are only compared with Back rides with the same time ($at_j = dt_j$);
2. To rides are compared with Back rides departing at the same time or 0.5 hours later;
3. To rides are compared with Back rides departing at the same time and To rides departing 0.5 hours earlier are compared with Back rides, resulting in additional waiting time for customers.

In scenario 2 and 3 the value of \( \eta = 0.5 \), which means that it is allowed to let vehicle drivers wait 0.5 hours at the destination. This simple Excel model cannot deal with higher values of \( \eta \). In Table 10, results for each situation in each scenario are given:

<table>
<thead>
<tr>
<th>((f,m_1,(m_2),\xi))</th>
<th>Scenario</th>
<th>Matching savings in km</th>
<th>Savings percentage</th>
<th>Total savings (incl. Combination)</th>
<th>Total savings fraction</th>
</tr>
</thead>
<tbody>
<tr>
<td>((0.2; 0.25; 0))</td>
<td>1</td>
<td>44.307</td>
<td>3.8%</td>
<td>56.996</td>
<td>4.9%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>64.397</td>
<td>5.6%</td>
<td>77.086</td>
<td>6.7%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>50.902</td>
<td>4.4%</td>
<td>63.591</td>
<td>5.5%</td>
</tr>
<tr>
<td>((1, 0.5, 0.5))</td>
<td>1</td>
<td>39.619</td>
<td>3.4%</td>
<td>116.601</td>
<td>10.1%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>59.627</td>
<td>5.2%</td>
<td>136.609</td>
<td>11.8%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>55.814</td>
<td>4.8%</td>
<td>132.796</td>
<td>11.5%</td>
</tr>
<tr>
<td>((0.2; 0.25; 0.5,0))</td>
<td>1</td>
<td>43.633</td>
<td>3.8%</td>
<td>79.878</td>
<td>6.9%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>63.457</td>
<td>5.5%</td>
<td>99.702</td>
<td>8.6%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>49.405</td>
<td>4.3%</td>
<td>85.650</td>
<td>7.4%</td>
</tr>
</tbody>
</table>

The matching savings are higher for situations with fewer combinations, but the differences are not big. In the current situation, the savings are 4.9% when only bookings with equal times are matched. As mentioned before, the savings are based on the distance in which each individual booking is executed by its most optimal carrier. With more flexible settings, the total savings can reach a level of almost 10% which is quite large. Comparing To rides with Back rides departing 0.5 hours later (scenario 2) leads to a large increase of the savings (about 15,000 kilometers).

The third scenario has less impact because there are less possibilities since this scenario results in additional waiting times for customers. When the parameters \((f,m,\xi)\) are low, in many cases these matches are not allowed. When the \((1; 0.5; 0.5)\) setting is used, the results are better for the third scenario compared to the other solutions as a result of the parameter values.

Another point to take into account is to check how bookings which are combined are located. The most important aspect here is the distance. In most combinations, at least one of the combined bookings is located close to the destination (<20 km). A limited amount of combinations combines bookings which are both located more than 50 kilometers away from the airport. This holds for both the original \((0.2; 0.25; 0)\) and a more flexible \((1; 0.5; 0.5)\) setting, the patterns are similar for these settings.

Combination reduces mainly the empty kilometers, but also customer distances. Matching only reduces empty kilometers. The average savings for customer distances in the current situation are 20 kilometers and 10 for the flexible setting. Vehicle distances are around 45 for both settings in Combination and around 38 for Matching. In Appendix 10 Tables (22-24) are given in which the findings are summarized and discussed.

As mentioned before, the results of the simple Excel model are not optimal. Based on these results, it can be concluded that combining and matching rides might be attractive. In the next section, a Linear Programming model will be used to improve the results.
6.4. Linear Programming model results

In this section the results of the Linear Programming Model are presented. In Appendix 9, the model description of 5.4.1 and 5.4.2. is worked out for the settings of Tinker. AIMMS is used to implement and test the model, using data of Tinker. This section compares the simple Excel model results with the results of the LP model. The advantage of the Linear Programming model is that it can compare all bookings which are entered in the model and the model is able to deal with all possible parameter values. The simple Excel model described in the previous section cannot do that. Because all bookings must be compared to each other, the LP model runs very slow. To compare 1.000 bookings, a standard computer requires about 5 minutes loading the data and calculating the combinations and matches. When 2.000 bookings are loaded, over 20 minutes are required. So an expensive computer is required to use the model if the number of bookings to compare is over 1.000. Because of the large calculation times for high quantities of data, instead of calculating the savings for all months between December 2013 and May 2014, only the data of December (970 To rides and 578 Back rides, resulting in a total of 1.448 bookings) will be used. From these months, December was selected, since it was (based on the number of bookings) an average month. The results of both the LP and simple Excel model for December will be compared for \( \eta = 0 \). Since calculations for \( \eta = 0.5 \) use different data as the LP model, these values cannot be used to compare the models.

The LP model is executed for three situations: the original scenario \((0.2; 0.25; 0)\), the mixed \((0.2; 0.25; 0.5; 0.5)\) setting and the flexible \((1; 0.5; 0.5)\) setting. The effects are measured for \( \eta = 0, 0.5 \) and 1. The total distance without combination/matching is 171.862 in December. Results are given in Table 11.

The absolute savings (the number of kilometers saved compared to the situation without combination and matching) are given for combination To rides, combination Back rides and Matching. The sum of these values is the Total savings. The fraction can be found by dividing the total savings with 171.862.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Model</th>
<th>Value of ( \eta )</th>
<th>Combination To Rides</th>
<th>Combination Back Rides</th>
<th>Matching</th>
<th>Total Savings</th>
<th>Fraction</th>
</tr>
</thead>
<tbody>
<tr>
<td>((0.2; 0.25; 0))</td>
<td>Excel</td>
<td>0</td>
<td>1.683</td>
<td>584</td>
<td>4.551</td>
<td>6.818</td>
<td>4.0%</td>
</tr>
<tr>
<td></td>
<td>LP</td>
<td>0</td>
<td>1.683</td>
<td>584</td>
<td>6.751</td>
<td>9.018</td>
<td>5.2%</td>
</tr>
<tr>
<td></td>
<td>LP</td>
<td>0.5</td>
<td>1.683</td>
<td>584</td>
<td>9.440</td>
<td>11.707</td>
<td>6.8%</td>
</tr>
<tr>
<td></td>
<td>LP</td>
<td>1</td>
<td>1.683</td>
<td>584</td>
<td>12.436</td>
<td>14.703</td>
<td>8.6%</td>
</tr>
<tr>
<td>((0.2; 0.25; 0.5; 0))</td>
<td>Excel</td>
<td>0</td>
<td>4.325</td>
<td>1.260</td>
<td>4.384</td>
<td>9.969</td>
<td>5.8%</td>
</tr>
<tr>
<td></td>
<td>LP</td>
<td>0</td>
<td>4.523</td>
<td>1.260</td>
<td>6.604</td>
<td>12.387</td>
<td>7.2%</td>
</tr>
<tr>
<td></td>
<td>LP</td>
<td>0.5</td>
<td>4.523</td>
<td>1.260</td>
<td>9.615</td>
<td>15.398</td>
<td>9.0%</td>
</tr>
<tr>
<td></td>
<td>LP</td>
<td>1</td>
<td>4.523</td>
<td>1.260</td>
<td>12.416</td>
<td>18.199</td>
<td>10.6%</td>
</tr>
<tr>
<td>((1; 0.5; 0.5))</td>
<td>Excel</td>
<td>0</td>
<td>8.578</td>
<td>2.956</td>
<td>3.805</td>
<td>15.339</td>
<td>8.9%</td>
</tr>
<tr>
<td></td>
<td>LP</td>
<td>0</td>
<td>10.017</td>
<td>2.984</td>
<td>6.406</td>
<td>19.407</td>
<td>11.3%</td>
</tr>
<tr>
<td></td>
<td>LP</td>
<td>0.5</td>
<td>10.017</td>
<td>2.984</td>
<td>10.954</td>
<td>23.955</td>
<td>13.9%</td>
</tr>
<tr>
<td></td>
<td>LP</td>
<td>1</td>
<td>10.017</td>
<td>2.984</td>
<td>12.601</td>
<td>25.602</td>
<td>14.9%</td>
</tr>
</tbody>
</table>

Table 11: Savings in the Linear Programming Model compared to the Simple Excel Model
Based on these results, it can be found that the differences between the Excel and LP model in combination are small under similar parameter settings. For the current situation \((m = 0.25, f = 0.2)\) there are even no differences (both models result in 1.683 and 584). When the values of \(m, f\) and \(\xi\) are higher, the differences increase, which is a result of the higher possible number of combinations (as given in Table 7). Since the number of possible combinations is limited in the current situations, it is likely that there are only a few bookings that could be combined with each other, meaning that the simple Excel model with \(z = 3\) compares all possible combinations. If the parameters are more flexible, it is likely that more bookings could be combined. In that case, Excel does not compare all possibilities, resulting in differences.

For matching, the differences between the models are higher. Since the Excel model is limited in the calculations, the LP model results in higher savings. The only situations which can be compared are the situations with \(\eta = 0\), since Excel does not consider all possibilities when \(\eta = 0.5\) and the model even cannot handle with \(\eta = 1\). The LP model scores at matching 1.48 times better in the current situation, 1.51 times in the mixed setting and 1.68 times in the flexible setting, considering matching only. As a result, the total savings are 1.32, 1.24 and 1.27 times higher for LP compared to Excel.

The total savings are higher for matching as for combination for the current setting. The higher values of matching are a result of the fact that for matching there are less constraints as for combination; for matching only the arrival and departing times must be close to each other, while combination also has constraints related to capacity and travel times. The difference between matching and combination is smaller in the mixed setting, and in the flexible setting combination savings are even higher as matching savings for \(\eta = 0\).

The average kilometer savings per combination are higher for combination (63 for To rides, 59 for Back rides) as for matching (49). This is a result of the fact that combination saves both Customer distances (with customers in the taxi) and Vehicle distances (without customers in the taxi), while matching only saves Vehicle distances. The explanation of these values is given in Appendix 11 (Table 23). These values are comparable to the results of the simple Excel model, in which was concluded that Combinations saves on average about 20 kilometers on customer distances and 45 kilometers on vehicle distances, while the average savings for matching are around 38.

It can be concluded that the difference between the models is not big for combination, but are large for matching. Higher waiting times for the vehicle drivers, which cannot be entered in the Excel model, improve the model a lot, so if these are allowed, using the LP model is recommended.

In chapter 7 is described how the model should be implemented. In chapter 8, conclusion and recommendations are given. Here, suggestions are given about what model should be used under which circumstances.
7. Implementation

To use the model, the current database must be used as input. The location of the customers, the destinations of the customers and the carriers, maximum capacity, values for $\sigma$, $\tau$, and $\eta$, load and unload times and (if distance coordinates are used) a formula for the speed calculation should be set. Booking specific information as the arrival time, maximum waiting times, maximum additional travel time and fraction, luggage, number of passengers and class type is also required as input. Each time when a customer books the database should be updated with that booking.

The described models can be implemented step by step. Tinker could for example first implement the Back Rides combination model. If additional travel/waiting/load and unload times are longer as expected, this will not result in customers missing a flight, therefore the risk for the customers is lower if the approach is implemented first. If it works, it can be implemented for To rides as well. Matching can be added afterwards. Alternatively, and recommended, matching could be implemented first, starting with matching only rides with a similar time ($\eta = 0$), since this does not have a big effect on the customers. The calculated total savings are higher for matching as for combination in the current situation, so Matching should be implemented first. To keep the progress visible, it is recommended to implement it step by step.

To and Back rides are considered differently, there is a dataset for the To rides and one for the Back rides. The combination problem will be solved for both To and Back rides. The problem must be solved when all data is available, since (depending on the values of the parameters) the results of several bookings can change if one booking is added to the list, especially for the bookings with a travel time close to that booking.

The model must be run on daily basis and the schedule must be known at least 1 day in advance in order to communicate the schedule on time to the carriers. This means that at least 24 hours before a booking should be executed, the planning for that booking must be known by the carriers. All known bookings which are not scheduled must be taken into account to also be able to combine bookings at the end of day $i$ with bookings of the beginning of day $i + 1$. So the steps to take when using the model are as follows:

- Continuously: update dataset with new bookings;
- Each day ($i$), 0:00: schedule all bookings of day $i + 1$ which are not scheduled before, considering all bookings in the dataset;
- Send schedule to carriers;
- Remove used bookings from dataset (all bookings of day $i$ and possible bookings from other days which are combined with a booking from day $i$).

Instead of considering all bookings, it is enough to consider all bookings of day $i$ plus the bookings of day for which the arrival or depart time is at most $\xi_i$ hours later as day $i$ (for the schedule of 1 May and a value of $\xi_i = \xi = 2$, bookings between 1 May 0:00 and 2 May 2:00 must be considered).

If a schedule is created with this model, a back-up plan must always be available, it is needed for example when flights are delayed. This is known hours in advance, therefore the carriers must have a back-up taxi if a combined ride cannot be combined anymore as a result.

In the next and final chapter, conclusions and recommendations are given.
8. Conclusions and recommendations

In this report, models are presented which can be used to combine bookings in one ride and match rides to each other efficiently. Two types of combining rides are considered; **Combination** was defined as using one vehicle to transport multiple bookings in one vehicle to their destination and **Matching** was defined as planning rides in such a way that after the execution of a ride To the destination, the vehicle can execute a ride Back starting from a location at (or close to) that destination almost directly afterwards.

These models were applied to Tinker, a company using an online booking system to transport customers from their location to airports like Schiphol by taxi. Tinker lets customers choose to allow combination or not. A dataset around 10,000 bookings has been used to test the models.

In the first section of this chapter, the results of the project are linked to the research questions and assignment as described in section 3.3. Section 8.2 continues with general conclusions.

8.1. Conclusions Research Questions / Assignment

RQ1: To what extend does the maximum additional travel/waiting time affect the abilities to combine taxi rides, respecting customer satisfaction.

In the process to combine two bookings in one ride, several constraints required attention, the maximum number of passengers and luggage may not be exceeded, arrival times (for rides To the airport/destination) and departing times (for rides Back) may not differ too much and the total additional travel time (time that customers travel longer) and waiting time (time that customers arrive to early, or depart later as planned from the airport) may not exceed a maximum value, both for the customers and vehicle drivers. In addition both customers must allow combination.

These variables had a strong effect on the solution. Allowing a customer waiting time of 30 minutes ($\xi = 0.5$) almost doubles the number of possible combinations. Increasing the allowed fraction ($f$) to 0.5 even triples the number of possible combinations. Allowing more additional travel time ($m$) only does not have much impact, however if the fraction ($f$) is increased as well it will improve the combinability. This effect is even stronger for high values of $f$. Allowing a waiting time for the vehicle driver in order to match rides also has a big effect, allowing a 30 minutes waiting time increases the number of possible matches by 50%.

RQ2: What kind of cost reduction can be achieved by combining multiple To and or Back bookings and scheduling rides in such a way that multiple rides are executed with the same vehicle by doing a Back ride (almost) directly after a To ride?

The reductions depend on the parameter settings of RQ1. In the current settings, only 2% of the bookings can be combined, resulting in a savings of only 1% for combination. 4% is saved with matching. Savings are the reduction of kilometers to execute the ride, which are calculated by taking the difference of the distances of the current situation and the distances in the combined / matched setting. For higher values, the savings can increase to a higher values, depending on the parameter settings even values above 10% can be reached.
Implement a Transportation Heuristic dealing with extensions to improve the allocation of combined / matched rides.

Two type of models are developed; an Excel model which is easy to use, but only can compare a limited amount of bookings (not resulting in an optimal solution) and a Linear Programming model which is harder to implement and use, but has better solutions. The models are used to compare the original situation by situations with different values for the maximum allowed additional travel time, fraction (compared to the original situation in which the bookings are not combined) and waiting times and to calculate the reduction of kilometers for each setting. Using the Linear Programming models shows solutions which are better, resulting in higher savings (about 30%), the difference is not high for combination, but for matching it does have a strong effect. The most important advantage of the Linear Programming model is that it can handle all possible combinations of parameter settings, while the Excel models is limited in that.

8.2. Recommendations to Tinker

When the Excel or LP model will be implemented, it must be sure that all data is correct. Here it is assumed that the customers are always ready on time. Especially when people travel back home, this might not be the case. Customers require different times to collect luggage and get to the airport. Load and unload times must be clear, the model is able to add load and unload times, but the correct values must be known to be able to use them. The effect of load and unload times is not tested in this project, but is an interesting topic for further research. Tinker currently uses SQL to make the schedules for carriers, so in the most optimal case the combination and matching must be done within the SQL server. Instead of delivering a list of customers which must be picked up by the carrier Tinker must deliver a list of rides in which combination and matching already has taken place.

Considering changes of the values of \( m, f, \xi \) and \( \eta \) increase the number of possibilities. The model of section 5.2 can be used to find the best fit. In addition, market research must be done to learn how customers and vehicle drivers react on these changes. If customers tend to not allow combinations if these values are increased, it is better not to consider that. An interesting topic for further research is to find the optimal balance between an increase of (one of) the parameters and the customers allowing combinations (by selecting Economic Class). This also can be related to the offered discount to the customer for selecting Economic Class. Also the savings can be a topic for further research. In this report, the savings are expressed in kilometers, but in reality more factors are involved. Costs of vehicle drivers and maintenance of vehicles are examples of costs which also affect the savings / costs.

Tinker can choose between the Excel model of section 5.3 and the LP model of section 5.4. If Tinker wants the optimal solution and can implement the LP problem without many difficulties, they should do that. In the SQL language, it is possible to enter the Linear Programming formulation, however the values must be calculated in a reasonable amount of time. If the solution does not have to be optimal and Tinker accepts a less optimal solution, the Excel model can be used to allocate bookings and rides, although the differences between the models cannot be neglected, especially for the matching procedure.
References


Appendices
Appendix 1: Cost structure analysis
Here the runs of price calculations (Tinker: Smarter in Travel, 2013), related to section 4.1 are presented in Table 12. This example gives an indication how distance, number of passengers and luggage, direction and booking days in advance influence the price difference between Comfort Class and Economic Class. The formula used to calculate the prices is given in section 4.1. The number of passengers, distance, luggage (type A, B and C), Booking Time in Advance, Direction (To, Back or both), price for Economic Class, price for Comfort Class and the difference of the price between the two classes are given.

<table>
<thead>
<tr>
<th>Persons</th>
<th>Distance</th>
<th>Luggage A</th>
<th>Luggage B</th>
<th>Luggage C</th>
<th>Time in advance</th>
<th>To</th>
<th>Back</th>
<th>Economic Price</th>
<th>Comfort Price</th>
<th>Price difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>106</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>1</td>
<td>1</td>
<td>120.85</td>
<td>137.8</td>
<td>16.95</td>
</tr>
<tr>
<td>1</td>
<td>106</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>1</td>
<td>0</td>
<td>61</td>
<td>69.55</td>
<td>8.55</td>
</tr>
<tr>
<td>1</td>
<td>106</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>1</td>
<td>59.85</td>
<td>68.25</td>
<td>8.4</td>
</tr>
<tr>
<td>1</td>
<td>106</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>1</td>
<td>58.15</td>
<td>67.4</td>
<td>9.25</td>
</tr>
<tr>
<td>1</td>
<td>106</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>1</td>
<td>59</td>
<td>68.25</td>
<td>9.25</td>
</tr>
<tr>
<td>1</td>
<td>106</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>30</td>
<td>0</td>
<td>1</td>
<td>59</td>
<td>67.4</td>
<td>8.4</td>
</tr>
<tr>
<td>1</td>
<td>106</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>30</td>
<td>0</td>
<td>1</td>
<td>64.05</td>
<td>73.25</td>
<td>9.2</td>
</tr>
<tr>
<td>1</td>
<td>106</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>1</td>
<td>63.2</td>
<td>72.45</td>
<td>9.25</td>
</tr>
<tr>
<td>1</td>
<td>106</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>1</td>
<td>68.25</td>
<td>77.45</td>
<td>9.2</td>
</tr>
<tr>
<td>1</td>
<td>106</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>1</td>
<td>75.8</td>
<td>85.85</td>
<td>10.05</td>
</tr>
<tr>
<td>1</td>
<td>106</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>1</td>
<td>80.85</td>
<td>91.75</td>
<td>10.9</td>
</tr>
<tr>
<td>1</td>
<td>106</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>1</td>
<td>92.6</td>
<td>104.35</td>
<td>11.75</td>
</tr>
<tr>
<td>1</td>
<td>106</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>1</td>
<td>106.85</td>
<td>119.45</td>
<td>12.6</td>
</tr>
<tr>
<td>1</td>
<td>106</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>1</td>
<td>122.8</td>
<td>122.8</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>106</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>1</td>
<td>N/A</td>
<td>N/A</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>106</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>1</td>
<td>59.85</td>
<td>68.25</td>
<td>8.4</td>
</tr>
<tr>
<td>2</td>
<td>106</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>1</td>
<td>66.9</td>
<td>76.15</td>
<td>9.25</td>
</tr>
<tr>
<td>2</td>
<td>106</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>1</td>
<td>74.45</td>
<td>83.7</td>
<td>9.25</td>
</tr>
<tr>
<td>3</td>
<td>106</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>1</td>
<td>77</td>
<td>87.05</td>
<td>10.05</td>
</tr>
<tr>
<td>3</td>
<td>106</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>1</td>
<td>79.5</td>
<td>89.6</td>
<td>10.1</td>
</tr>
<tr>
<td>3</td>
<td>106</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>1</td>
<td>83.7</td>
<td>93.8</td>
<td>10.1</td>
</tr>
<tr>
<td>4</td>
<td>106</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>1</td>
<td>87.9</td>
<td>97.15</td>
<td>9.25</td>
</tr>
<tr>
<td>5</td>
<td>106</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>1</td>
<td>98</td>
<td>107.2</td>
<td>9.2</td>
</tr>
<tr>
<td>6</td>
<td>106</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>1</td>
<td>108.9</td>
<td>118.15</td>
<td>9.25</td>
</tr>
<tr>
<td>1</td>
<td>18</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>1</td>
<td>10.6</td>
<td>12.15</td>
<td>1.55</td>
</tr>
<tr>
<td>1</td>
<td>18</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>1</td>
<td>10.9</td>
<td>12.25</td>
<td>1.35</td>
</tr>
<tr>
<td>1</td>
<td>18</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>1</td>
<td>11.45</td>
<td>12.95</td>
<td>1.5</td>
</tr>
<tr>
<td>1</td>
<td>18</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>1</td>
<td>12.25</td>
<td>13.8</td>
<td>1.55</td>
</tr>
<tr>
<td>2</td>
<td>18</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>1</td>
<td>12.8</td>
<td>14.3</td>
<td>1.5</td>
</tr>
</tbody>
</table>

53
The price depends on the number of passengers, when more passengers travel, the total price increases, but the price per passenger decreases. The absolute price difference between Comfort and Economic class increases slightly, the relative price difference between the two classes decreases a bit. In Figure 8 this effect is illustrated.
The number of luggage units also have an effect. The effect on the discount for allowing combinations is similar as in the situation with passengers. However, when the maximum luggage units is reached (and the booking cannot be combined with other bookings taking luggage), the difference between the classes is zero, since it is hardly possible to make a combination. In Figure 9 steps of 4 Luggage Units are taken and the prices for Comfort class (upper price) and Economic Class (bottom price) are given.

The booking days in advance also influence the price. People who book early get a cheaper price. This is done to stimulate customers to book early. The discount for allowing combination is higher when customers book earlier, reducing to almost zero if people book only a few days in advance. This is done because it is much more difficult to combine bookings which are done late because of the fact that schedules must be known by carriers in advance. Prices for last minute bookings is around 2.5 times higher as the price for early bookings (>61 days in advance). In Figure 10 the effect is illustrated.
The largest difference between the two is 16 euro for this situation. The cheapest possible costs for this setting are 59,95 euro. Public transport for this route (106 kilometer) is €28,9 for first class and €18,3 for second class transport. Fuel costs are 17,09 euro (9292ov.nl, 2014).

Finally, distance also has an effect. When bookings are located close to the airport the discount for allowing combinations is small. This is done because these customers probably will not face any additional travel time, and the savings are smaller for these bookings. When the distance is increased, the offered compensation is much higher, growing to a value above €15 for distances over 200 kilometers. Figure 11 shows the effect.

Again, the red line present comfort class and the blue economic class. The distance is mentioned in the graph as well.
Appendix 2: Data characteristics of Tinker

Here Figures related to booking characteristics are given. As mentioned in section 4.1, the selected data is the first 10,000 bookings of Tinker (November 2012 – September 2013).

Most bookings have two passengers travelling. Only about 900 bookings have more than 4 passengers involved, these bookings require a taxi bus. In Figure 12, the totals per category are given:

![Figure 12: Number of people travelling per booking](image)

In Figure 13, a similar graph is presented for the luggage units (LU). Most people take 12 LU with them, the total bookings per number of LU is given:

![Figure 13: Luggage quantities per booking](image)

Figure 14 gives information about the booking days in advance. It can be found that most people book short before they travel, 45% of them book within 10 days in advance. In Figure 15, the results for people booking at most ten days in advance is presented in more detail, it can be found that most people book 2 days in advance (the minimum possible quantity using the online booking system of Tinker), and there is a decreasing pattern. The 118 bookings booked one day in advance are not done with the online booking system, but these people called Tinker to make a reservation.
The fraction of people booking over 50 days in advance is small (about 800 bookings, 8%).

![Figure 14: Bookings per day in advance over long term](image)

![Figure 15: Bookings per day in advance (0-10 days)](image)

When looking at the distance to the destination (airport) of the customers it can be concluded that most people live close to the airport (10-30 kilometers). However, there are also bookings from customers who have to travel over 100 kilometers (about 750). In Figure 16, the number of bookings are grouped in windows of 10 kilometers.

![Figure 16: Bookings grouped by distance, given in kilometers](image)
To find information about how many customers allow combination a different dataset is used. Bookings between November 2013 and March 2013 are used for that analysis, as described in section 4.2. Looking at the left Figure of Figure 17, it can be found that when customers travel with more passengers, the fraction that allows combination is higher. For the luggage (which is presented in the right graph of Figure 17) a similar effect is found.

Customers who have to travel longer are less likely to allow their rides to be combined with other customers. There is a decreasing pattern for the fraction of customers choosing Economic class, which is showed in Figure 18. Because the number of bookings with large distances is small, there are small increases around 140 kilometers and 190 kilometers. About 90% of the customers who live close to the airport select Economic Class, this fraction decreases to about 40% for customers living over 200 kilometers away from the airport.

Time does not have a large effect on the percentage of customers selecting Economic Class. Figure 19 shows the percentage in clusters of hours for all rides, in Figure 20 the bookings are separated in To and Back rides. In the night (0:30-5:30) the fraction is a bit lower and during the morning (6:00-9:00) a bit higher, but the differences are not very large. The difference between To and Back rides is also not large.
The red line indicates Back rides and the blue line indicates To rides. Some of these time windows only have a few data points, resulting in values of 1 and zero. In Table 13, the number of rides in each window and the fractions allowing combination are given.

Table 13: Fractions Class grouped per time and direction

<table>
<thead>
<tr>
<th>Time</th>
<th>To Rides</th>
<th>Fraction Economic</th>
<th>Back Rides</th>
<th>Fraction Economic</th>
</tr>
</thead>
<tbody>
<tr>
<td>0:00:00</td>
<td>84</td>
<td>0.68</td>
<td>5</td>
<td>0.60</td>
</tr>
<tr>
<td>0:30:00</td>
<td>178</td>
<td>0.61</td>
<td>1</td>
<td>1.00</td>
</tr>
<tr>
<td>1:00:00</td>
<td>127</td>
<td>0.49</td>
<td>5</td>
<td>0.80</td>
</tr>
<tr>
<td>1:30:00</td>
<td>108</td>
<td>0.51</td>
<td>7</td>
<td>0.57</td>
</tr>
<tr>
<td>2:30:00</td>
<td>194</td>
<td>0.58</td>
<td>83</td>
<td>0.52</td>
</tr>
<tr>
<td>3:00:00</td>
<td>64</td>
<td>0.64</td>
<td>132</td>
<td>0.52</td>
</tr>
<tr>
<td>4:00:00</td>
<td>136</td>
<td>0.63</td>
<td>424</td>
<td>0.59</td>
</tr>
<tr>
<td>4:30:00</td>
<td>38</td>
<td>0.50</td>
<td>328</td>
<td>0.58</td>
</tr>
<tr>
<td>Time</td>
<td>Value</td>
<td>Speed</td>
<td>Distance</td>
<td>Average</td>
</tr>
<tr>
<td>--------</td>
<td>-------</td>
<td>-------</td>
<td>----------</td>
<td>---------</td>
</tr>
<tr>
<td>5:30:00</td>
<td>38</td>
<td>0.76</td>
<td>936</td>
<td>0.67</td>
</tr>
<tr>
<td>6:00:00</td>
<td>19</td>
<td>0.89</td>
<td>302</td>
<td>0.70</td>
</tr>
<tr>
<td>7:00:00</td>
<td>110</td>
<td>0.52</td>
<td>389</td>
<td>0.68</td>
</tr>
<tr>
<td>7:30:00</td>
<td>69</td>
<td>0.67</td>
<td>233</td>
<td>0.75</td>
</tr>
<tr>
<td>8:30:00</td>
<td>183</td>
<td>0.69</td>
<td>365</td>
<td>0.70</td>
</tr>
<tr>
<td>9:00:00</td>
<td>83</td>
<td>0.64</td>
<td>134</td>
<td>0.72</td>
</tr>
<tr>
<td>10:00:00</td>
<td>160</td>
<td>0.58</td>
<td>256</td>
<td>0.61</td>
</tr>
<tr>
<td>10:30:00</td>
<td>60</td>
<td>0.62</td>
<td>109</td>
<td>0.71</td>
</tr>
<tr>
<td>11:30:00</td>
<td>179</td>
<td>0.67</td>
<td>223</td>
<td>0.64</td>
</tr>
<tr>
<td>12:00:00</td>
<td>103</td>
<td>0.67</td>
<td>134</td>
<td>0.63</td>
</tr>
<tr>
<td>13:00:00</td>
<td>165</td>
<td>0.61</td>
<td>162</td>
<td>0.63</td>
</tr>
<tr>
<td>13:30:00</td>
<td>113</td>
<td>0.59</td>
<td>61</td>
<td>0.67</td>
</tr>
<tr>
<td>14:30:00</td>
<td>249</td>
<td>0.67</td>
<td>149</td>
<td>0.59</td>
</tr>
<tr>
<td>15:00:00</td>
<td>104</td>
<td>0.68</td>
<td>81</td>
<td>0.72</td>
</tr>
<tr>
<td>16:00:00</td>
<td>158</td>
<td>0.63</td>
<td>113</td>
<td>0.58</td>
</tr>
<tr>
<td>16:30:00</td>
<td>101</td>
<td>0.57</td>
<td>50</td>
<td>0.64</td>
</tr>
<tr>
<td>17:30:00</td>
<td>175</td>
<td>0.73</td>
<td>78</td>
<td>0.58</td>
</tr>
<tr>
<td>18:00:00</td>
<td>84</td>
<td>0.57</td>
<td>48</td>
<td>0.71</td>
</tr>
<tr>
<td>19:00:00</td>
<td>146</td>
<td>0.67</td>
<td>102</td>
<td>0.72</td>
</tr>
<tr>
<td>19:30:00</td>
<td>83</td>
<td>0.66</td>
<td>38</td>
<td>0.63</td>
</tr>
<tr>
<td>20:00:00</td>
<td>205</td>
<td>0.66</td>
<td>15</td>
<td>0.73</td>
</tr>
<tr>
<td>21:00:00</td>
<td>83</td>
<td>0.67</td>
<td>5</td>
<td>0.40</td>
</tr>
<tr>
<td>22:00:00</td>
<td>124</td>
<td>0.69</td>
<td>7</td>
<td>0.57</td>
</tr>
<tr>
<td>22:30:00</td>
<td>71</td>
<td>0.73</td>
<td>2</td>
<td>1.00</td>
</tr>
<tr>
<td>23:30:00</td>
<td>362</td>
<td>0.62</td>
<td>3</td>
<td>1.00</td>
</tr>
</tbody>
</table>

The average speed is used to translate distances in times. When distances are longer, the average speed follows an increasing pattern. Bookings close to the airport have an average speed of about 50 kilometers per hour, increasing to over 100 kilometers per hour for distances over 200 kilometers. This is illustrated in Figure 21.
The effect of time on the average speed is not very large. When using the same hour slots as in the analysis of customers selecting Economic Class it can be found that the average speed does not heavily depend on the time in which the ride is executed. In Figure 22 the average speed (blue line) and distance in km (red line) are given.

Figure 22: Speed and distance related to time.
Appendix 3: Schiphol flights fraction per time

In this section flight patterns of Schiphol are given. In Figure 23, the fraction of landings and departures (take-offs) is given in clusters of hours. It can be found that most flights are executed in the morning, and in the night a smaller fraction of flights is executed.

Figure 23: Flight pattern Schiphol in hours in percentages

The pattern of Tinker depends on the flight pattern since their customers take a flight. However, it can be found that a relatively higher amount of people use Tinker during the night and early morning. This probably is the result of fewer availability of public transport and family / friends bringing people to/back from the airport. The pattern of Tinker is related to the flight pattern of Schiphol, but the pattern is shifted since To rides of Tinker are executed about 3 hours before departure and Back rides are executed 1 hour after the landing. This effect is visualized in Figure 24 for To rides and Figure 25 for Back rides. The value are given in clusters of one hour. It can be found that the Back rides are spread more over the days as the To rides:
The presented histograms show the flight times, meaning that customers will depart approximately 1 hour after this flight time, for arriving flight it holds that customers must have arrived at the airport 2 or 3 hours before this time. This Figure shows the fraction of demand per half hour.
Appendix 4: Allocation Example

This chapter gives a small example related to section 5.1. The booking information of Table 14 will be used, assuming that direction and classes are the same. BIA represents the booking time in advance:

<table>
<thead>
<tr>
<th>I</th>
<th>D</th>
<th>Postcode</th>
<th>Customer</th>
<th>Postcode</th>
<th>Location</th>
<th>Time</th>
<th>Pax</th>
<th>LU</th>
<th>BIA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>1017</td>
<td>1118</td>
<td>4.900</td>
<td>52.350</td>
<td>1-12-2013 3:00</td>
<td>1</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>1394</td>
<td>1118</td>
<td>5.050</td>
<td>52.250</td>
<td>1-12-2013 3:30</td>
<td>2</td>
<td>14</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>1511</td>
<td>1118</td>
<td>4.875</td>
<td>52.450</td>
<td>1-12-2013 4:00</td>
<td>2</td>
<td>8</td>
<td>29</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>3136</td>
<td>1118</td>
<td>4.350</td>
<td>51.925</td>
<td>1-12-2013 4:30</td>
<td>2</td>
<td>16</td>
<td>51</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>7203</td>
<td>1118</td>
<td>6.200</td>
<td>52.150</td>
<td>1-12-2013 5:00</td>
<td>2</td>
<td>12</td>
<td>82</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>1055</td>
<td>1118</td>
<td>4.850</td>
<td>52.375</td>
<td>1-12-2013 5:30</td>
<td>2</td>
<td>7</td>
<td>28</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>1058</td>
<td>1118</td>
<td>4.850</td>
<td>52.350</td>
<td>1-12-2013 6:00</td>
<td>1</td>
<td>4</td>
<td>30</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>1067</td>
<td>1118</td>
<td>4.800</td>
<td>52.375</td>
<td>1-12-2013 7:30</td>
<td>1</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>5211</td>
<td>1118</td>
<td>5.300</td>
<td>51.700</td>
<td>1-12-2013 8:00</td>
<td>1</td>
<td>6</td>
<td>8</td>
</tr>
</tbody>
</table>

In this example two fleets are considered, for fleet 1 the maximum LU is 32 in this example and maximum passengers is 8, for fleet 2 these values are 18 and 4 (this is similar to the situation of Tinker). It is assumed that at most 2 customers can be combined. First the allocation procedure will be described, after that the solutions are compared.

Appendix 4a: Location/destination method

The distance matrix is given in Table 15 (rounded on km). The distance formula of Pythagoras presented in paragraph 5.2.1 is used to calculate the distances:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>21</td>
<td>12</td>
<td>82</td>
<td>155</td>
<td>7</td>
<td>6</td>
<td>12</td>
<td>90</td>
<td>19</td>
</tr>
<tr>
<td>2</td>
<td>21</td>
<td>0</td>
<td>31</td>
<td>91</td>
<td>136</td>
<td>28</td>
<td>26</td>
<td>33</td>
<td>71</td>
<td>36</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
<td>31</td>
<td>0</td>
<td>88</td>
<td>160</td>
<td>9</td>
<td>12</td>
<td>13</td>
<td>102</td>
<td>23</td>
</tr>
<tr>
<td>4</td>
<td>82</td>
<td>91</td>
<td>88</td>
<td>0</td>
<td>220</td>
<td>79</td>
<td>77</td>
<td>75</td>
<td>115</td>
<td>65</td>
</tr>
<tr>
<td>5</td>
<td>155</td>
<td>136</td>
<td>160</td>
<td>220</td>
<td>0</td>
<td>161</td>
<td>161</td>
<td>167</td>
<td>119</td>
<td>172</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>28</td>
<td>9</td>
<td>79</td>
<td>161</td>
<td>0</td>
<td>3</td>
<td>6</td>
<td>96</td>
<td>15</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>26</td>
<td>12</td>
<td>77</td>
<td>161</td>
<td>3</td>
<td>0</td>
<td>7</td>
<td>93</td>
<td>13</td>
</tr>
<tr>
<td>8</td>
<td>12</td>
<td>33</td>
<td>13</td>
<td>75</td>
<td>167</td>
<td>6</td>
<td>7</td>
<td>0</td>
<td>99</td>
<td>11</td>
</tr>
<tr>
<td>9</td>
<td>90</td>
<td>71</td>
<td>102</td>
<td>115</td>
<td>119</td>
<td>96</td>
<td>93</td>
<td>99</td>
<td>0</td>
<td>96</td>
</tr>
</tbody>
</table>

First, customer 1 is selected. Customer 1 is allocated to customer the most close to that customer (customer 7). The total passengers of these two bookings together is 2 with LU=10, which is below the maximum. Similarly, customer 2 is allocated to customer 3, customer 4 to 8 and customer 5 to 9. In this location/destination based allocation, the customer who lives the closest to the destination is picked up last. This allocation can be improved using linear programming.
Appendix 4b: Travel date and time
Based on travel date and time, customers 1-2, 3-4, 5-6 and 7-8 are allocated to each other. In this approach, the customer who must arrive at the airport first is picked up first.

Appendix 4c: Number of luggage and passengers
The First Fit Decreasing algorithm is used to allocate the rides. In this algorithm, bookings are sorted on not increasing order. Passengers are considered as a more important weight. This results in the order given in Table 16:

<table>
<thead>
<tr>
<th>Customer</th>
<th>Pax + LU</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>2,16</td>
</tr>
<tr>
<td>2</td>
<td>2,14</td>
</tr>
<tr>
<td>5</td>
<td>2,12</td>
</tr>
<tr>
<td>3</td>
<td>2,08</td>
</tr>
<tr>
<td>6</td>
<td>2,07</td>
</tr>
<tr>
<td>1</td>
<td>1,06</td>
</tr>
<tr>
<td>8</td>
<td>1,06</td>
</tr>
<tr>
<td>9</td>
<td>1,06</td>
</tr>
<tr>
<td>7</td>
<td>1,04</td>
</tr>
</tbody>
</table>

The item with the highest quantity is allocated to the alternative with the highest quantity which is allowed. This continues till no alternatives fit anymore. After this, the process starts over with the highest not yet allocated booking. Obviously, bookings can only be used once. Here, at most two bookings are combined. First customers are allocated in such a way that they fit in fleet 2. Starting with customer 4, the remaining number of passengers is 2 and remaining LU is 2 as well. It can be found that this booking cannot be combined in small fleet with another booking. Booking 2 can only be combined with booking 7. Booking 5 will be combined to booking 1, booking 3 with booking 6 and booking 8 with 9. The customer with the largest capacity quantity is picked up first in this heuristic.

Appendix 4d: Moment of booking
Here bookings are sorted on booking on advance time. This results in the values of Table 17:

<table>
<thead>
<tr>
<th>Customer</th>
<th>BIA</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>28</td>
</tr>
<tr>
<td>3</td>
<td>29</td>
</tr>
<tr>
<td>7</td>
<td>30</td>
</tr>
<tr>
<td>4</td>
<td>51</td>
</tr>
<tr>
<td>5</td>
<td>82</td>
</tr>
</tbody>
</table>
Since the numbers of bookings are 9, one booking will not be combined. Allocation is done in such a way that customer who booked last is allocated first to give customers who booked last an advantage. This results in combinations 8-2, 9-1, 3-6 and 4-7.

Appendix 4e: Conclusions
Now, the solutions will be compared based on waiting time, additional travel time and occupation. Distances to get from the carrier to the customer and from the carrier to the destination are ignored. The distance equals the distance from the first listed customer to the second listed customer plus the distance from the second listed customer to the destination. For not combined bookings, this is only the distance to the destination. The waiting time is the difference between the times of the bookings. The number of fleets is the total number of fleets used per type. The results are in Table 18.

<table>
<thead>
<tr>
<th>Method</th>
<th>Combinations</th>
<th>Total distance (in hours)</th>
<th>Total waiting time (in hours)</th>
<th># fleet 1, # fleet 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location/destination</td>
<td>1-7, 2-6, 3-8, 4-9, 5</td>
<td>19+43+24+180+172 = 438</td>
<td>4+2+3,5+3,5+0 = 13</td>
<td>2,3</td>
</tr>
<tr>
<td>Travel time</td>
<td>1-2, 3-4, 5-6, 7-8, 9</td>
<td>57+153+176+18+96 = 500</td>
<td>0,5+0,5+0,5+0,5+0 = 2</td>
<td>3,2</td>
</tr>
<tr>
<td>Capacity</td>
<td>2-7, 5-1, 3-6, 8-9, 4</td>
<td>39+174+24+195+65 = 497</td>
<td>4+2+1,5+0,5+0 = 8</td>
<td>0,5</td>
</tr>
<tr>
<td>Booking in Advance</td>
<td>8-2, 9-1, 3-6, 4-7, 5</td>
<td>69+109+24+90+172 = 464</td>
<td>4+5+1,5+2,5+0 = 13</td>
<td>2,3</td>
</tr>
</tbody>
</table>
Appendix 5: Formulas Maximum Additional Travel Time

The constraints related to the maximum additional travel time and the maximum additional travel fraction will be provided in Table 19. Since the own load and unload times of the customer exists at both sides, these terms are not presented in the absolute maximum additional travel time column. For the maximum additional travel fraction column, these should be taken into account since they do influence the ratio.

Table 19: Additional travel time constraints per setting

<table>
<thead>
<tr>
<th>Setting</th>
<th>Absolute Maximum Additional Travel Time</th>
<th>Maximum Additional Travel Fraction</th>
</tr>
</thead>
<tbody>
<tr>
<td>AB12</td>
<td>$t_{A,B} + t_{B,1} + l_B + u_B \leq t_{A,1} + m_A$</td>
<td>$1: \frac{t_{A,B}+t_{B,1}+l_B+l_B+u_A+u_B}{t_{A,1}+l_A+u_A} &lt; 1 + f_i$</td>
</tr>
<tr>
<td></td>
<td>$t_{B,1} + t_{1,2} \leq t_{B,2} + m_B$</td>
<td>$2: \frac{t_{B,1}+t_{1,2}+l_B+u_B}{t_{B,2}+l_B+u_B} \leq 1 + f_j$</td>
</tr>
<tr>
<td>AB21</td>
<td>$t_{A,B} + t_{B,2} + t_{2,1} + l_B + u_B \leq t_{A,1} + m_A$</td>
<td>$1: \frac{t_{A,B}+t_{B,2}+t_{2,1}+l_B+l_B+u_A+u_B}{t_{A,1}+l_A+u_A} \leq 1 + f_i$</td>
</tr>
<tr>
<td></td>
<td>$t_{B,2} \leq t_{B,2} + m_B$ (this is always true)</td>
<td>$2: \frac{t_{B,2}+l_B+u_B}{t_{B,2}+l_B+u_B} \leq 1 + f_j$ (this is always true)</td>
</tr>
<tr>
<td>BA12</td>
<td>$t_{A,1} \leq t_{A,1} + m_A$ (this is always true)</td>
<td>$1: \frac{t_{A,1}+u_A}{t_{A,1}+l_A+u_A} \leq 1 + f_i$ (this is always true)</td>
</tr>
<tr>
<td></td>
<td>$t_{B,A} + t_{A,1} + t_{1,2} + l_A + u_A \leq t_{B,2} + m_B$</td>
<td>$2: \frac{t_{B,A}+t_{A,1}+t_{1,2}+l_B+l_B+u_A+u_B}{t_{B,2}+l_B+u_B} \leq 1 + f_j$</td>
</tr>
<tr>
<td>BA21</td>
<td>$t_{A,2} + t_{2,1} \leq t_{A,1} + m_A$</td>
<td>$1: \frac{t_{A,2}+t_{2,1}+l_A+u_A}{t_{A,1}+l_A+u_A} \leq 1 + f_i$</td>
</tr>
<tr>
<td></td>
<td>$t_{B,A} + t_{A,2} + l_A + u_A \leq t_{B,2} + m_B$</td>
<td>$2: \frac{t_{B,A}+t_{A,2}+l_A+u_A+l_B+u_B}{t_{B,2}+l_B+u_B} \leq 1 + f_j$</td>
</tr>
</tbody>
</table>

These formulas are worked out based on the To scenario, but since symmetry is assumed, the formulas for travelling back are similar. Here times are considered, but instead of times, distances could have been considered as well.

The total travel time for the combined setting of booking $i$ (which is the left side of the absolute maximum additional travel time constraint, including the load and unload times) will be defined as $t_{comb,i,i,j}$. 

68
Appendix 6: Savings

Appendix 6a: Combination Savings for each setting
Savings are split up in customer savings (in which customers are in the taxi) and vehicle savings (the empty kilometers) in Table 20. The total savings are the sum of these values:

Table 20: Combination Savings per setting

<table>
<thead>
<tr>
<th>Setting</th>
<th>Customer Savings</th>
<th>Vehicle Savings</th>
</tr>
</thead>
<tbody>
<tr>
<td>AB12</td>
<td>( d_{A,1} + d_{B,2} - d_{A,B} - d_{B,1} - d_{1,2} )</td>
<td>( \text{MIN} \sum_k (d_{k,A} + d_{1,k}) + \text{MIN} \sum_l (d_{l,B} + d_{2,l}) - \text{MIN} \sum_m (d_{m,A} + d_{2,m}) )</td>
</tr>
<tr>
<td>AB21</td>
<td>( d_{A,1} - d_{A,B} - d_{2,1} )</td>
<td>( \text{MIN} \sum_k (d_{k,A} + d_{1,k}) )</td>
</tr>
<tr>
<td>BA12</td>
<td>( d_{B,2} - d_{B,A} - d_{1,2} )</td>
<td>( \text{MIN} \sum_l (d_{l,B} + d_{2,l}) )</td>
</tr>
<tr>
<td>BA21</td>
<td>( d_{A,1} + d_{B,2} - d_{B,A} - d_{A,2} - d_{2,1} )</td>
<td>( \text{MIN} \sum_k (d_{k,A} + d_{1,k}) + \text{MIN} \sum_l (d_{l,B} + d_{2,l}) - \text{MIN} \sum_m (d_{m,B} + d_{1,m}) )</td>
</tr>
</tbody>
</table>

The savings are presented in distances, the time can be found by dividing the distances with the corresponding average speed. When all possible savings are calculated, the most optimal combinations must be selected.

Appendix 6b: Matching Savings for each setting
In the matching procedure, only vehicle savings are saved. When ride \( I \) and \( J \) are matched, \( d_{k,I} \) is defined as the distance of depot \( k \) to the customer (the customer picked up first in a to ride, the customer brought back home last in a back ride). \( d_{a_{i,k}} \) is the distance between depot \( k \) and the ending location of a To ride or the starting location of a Back ride. Moreover, For the various settings, the distances are in Table 21:

Table 21: empty kilometers per setting

<table>
<thead>
<tr>
<th>Setting</th>
<th>To the first customer / Back from the last customer</th>
<th>Back from the last destination / To the first destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>AB12</td>
<td>( d_{k,A} )</td>
<td>( d_{a_{2,k}} )</td>
</tr>
<tr>
<td>AB21</td>
<td>( d_{k,A} )</td>
<td>( d_{a_{2,k}} )</td>
</tr>
<tr>
<td>BA12</td>
<td>( d_{k,B} )</td>
<td>( d_{a_{2,k}} )</td>
</tr>
<tr>
<td>BA21</td>
<td>( d_{k,B} )</td>
<td>( d_{a_{2,k}} )</td>
</tr>
</tbody>
</table>

In the matched scenario, two rides are matched and the empty kilometers to and back from the destination are spared. The vehicle distance in the matched setting equal \( \text{MIN}_m (d_{m,I} + d_{m,J}) \), in which \( m \) is the most optimal carrier to execute that drive. The savings equal \( d_{k,I} + d_{a_{i,k}} + d_{l,a_j} + d_{j,I} - (d_{m,I} + d_{m,J}) - d_{a_{i,a_j}} \). Since transitivity exists \( (d_{A,B} \leq d_{A,C} + d_{B,C}) \), it holds that \( d_{k,I} + d_{a_{i,k}} + d_{l,a_j} + d_{j,I} - (d_{m,I} + d_{m,J}) \geq 0 \), meaning that a match will result in a not negative saving if the destinations are the same.
Appendix 7: Excel models

Appendix 7a: Model to find how many bookings can be combined
The model takes the following actions for a dataset of $n$ bookings, starting at the first booking, $i = 1$:

1. Take booking $i$ from data list;
2. Take booking $i + 1$ from data list;
   a. Compare date and time;
   b. Check capacity (luggage and pax) constraint;
   c. Check distance/ time;
      i. Check distance / time constraints for setting AB12;
      ii. Check distance / time constraints for setting AB21;
      iii. Check distance / time constraints for setting BA12;
      iv. Check distance / time constraints for setting BA21;
   d. IF A=B=C (=MAX(i,ii,iii,iv)) = 1: RETURN 1, ELSE RETURN 0;
3. Set $i = i + 1$;
4. WHILE $i < n$ Return to step 1.

If the $n$ is reached, the process stops. The fraction of combined rides equals the sum of the output of step 2d divided by the number of bookings ($n$) minus 1 since the last booking cannot be compared with the next booking. Another way is by considering only a limited part of the dataset, in that case the fraction of combined rides equals the sum of the output of step 2d divided by the number of bookings ($n$). As output, the model shows the combine results and the fraction and the results per constraint type (a, b, c) the number of combinations (d) and the fractions.

Appendix 7b: Model to find how many rides can be matched
The model takes the following actions for a dataset of $n_{to}$ rides To the destination and $n_{back}$ rides back. This is a list of rides, which means that bookings which are combined in the combination step are entered as one ride. The procedure is as follows, starting at $l = 1, j = 1$:

1. Calculate the earliest arrival time $(at_l - \rho_l)$ of all bookings in the To rides list;
2. Calculate the latest depart time $(at_l + \eta)$ of all bookings in the To rides list.
3. Calculate the latest depart time $(dt_j + \rho_j)$ of all bookings in the Back rides list (the earliest depart time is equal to $dt_j$);
4. Take booking $l$ from data list of To rides;
5. Check for each booking in Back rides list whether $(dt_j \leq at_l + \eta)$ and $(dt_j + \rho_j) \geq (at_l - \rho_l)$. If this holds, RETURN 1, ELSE RETURN 0.
6. Take the sum of these values;
7. Set $l = l + 1$;
8. WHILE $l \leq n_{to}$ Return to step 4, ELSE Go to step 9
9. Take booking $j$ from data list of Back rides;
10. Check for each booking in To rides list whether $(dt_j < at_l + \eta)$ and $(dt_j + \rho_j) > (at_l - \rho_l)$. If this holds, RETURN 1, ELSE RETURN 0.
11. Take the sum of these values;
12. Set $j = j + 1$;
13. WHILE $j \leq n_{back}$ Return to step 9, ELSE STOP.

The sum of the output at 6 or 10 (these values should be equal) indicates how many possibilities there are to match rides. An additional output variable is counting the number of rides with positive values as output at 6 and 10 instead of taking the sum.

Appendix 7c: Combination model to calculate savings

The selection procedure of combinations when 6 bookings are used is now described. The model starts at booking $i = 1$. For each booking $i$, the booking is compared with bookings $i + 1, i + 2$ and $i + 3$. For both the list of bookings To the destination and the list of bookings Back this should be done. The formulas for the maximum additional travel time and maximum travel fraction (step 1b and 1c) are presented in Appendix 5. 1-2 denotes the savings which are related to booking 1 and 2, taking into account the constraints discussed in paragraph 5.2.2.1. The formulas for the savings (step 1e) are presented in Appendix 6 and are discussed in section 5.3.2. After calculating the distances, the model takes the following steps.

1. Check constraints for bookings $i + 1, i + 2$ and $i + 3$:
   a. Check times (check if $at_j - at_i \leq \xi_j$ for To rides and $at_j - at_i \leq \xi_i$ for Back rides). IF OK, then return 1, ELSE return 0;
   b. Check maximum additional absolute travel time constraint for each setting. IF OK, then return 1, ELSE return 0;
   c. Check maximum additional travel fraction constraint for each setting. IF OK, return 1 ELSE return zero;
   d. Take the maximum of the multiplication of step 2 and 3 for each setting;
   e. Calculate the savings for each allowed setting which is not locked (otherwise, return zero). Select the maximum of these;
   f. Check capacity constraint ($LU_i + LU_j \leq LU_{max}, pax_i + pax_j \leq pax_{max}$). IF OK, then return 1, ELSE return 0;
   g. Check whether combination is allowed (by multiplying 1, 4, 5 and the classes of the bookings (if a booking allows combination: class = 1 ELSE 0);
2. SELECT MAX 1-2, 1-3, 1-4. IF $>0$: Lock selected bookings;
3. Set $i = i + 1$;
4. IF $i$ is locked, Go to Step 3 ELSE Go to step 1

Taking the sum of the output in step 2 results in the total savings. More available output is the number of possible combinations, savings in vehicle and customer savings and the corresponding fractions. How these values are calculated will be described in paragraph 5.3.3.

Appendix 7d: Matching model to calculate savings

The following steps are taken, starting at ride $l = 1$. The starting value of Number = 0. $n_{to}$ is the total number of To rides (of both Classes).

1. Take booking $l$ from data list of To rides;
2. Look in Back Rides List Class 1
a. FIND booking \( J \) with similar time as booking \( I \).
   IF Found: Calculate Distance between destinations.
   IF \(<\text{MIN(Slack times Back Ride Bookings)}\)*. Calculate Savings,
   ELSE Return 0.
   Go To Step 2b. Number = Number + 1
   ELSE Go to Step 3a,
   b. SELECT booking \( J + 1 \).
   IF time is OK, Calculate Distance between destinations.
   IF \(<\text{Min(Slack time Back Ride Bookings)}\)*. Calculate Savings,
   ELSE Return 0
   Go to Step 2c. Number = Number + 1
   ELSE Go to Step 3b.
   c. SELECT booking \( J + 2 \).
   IF time is OK, Calculate Distance between destinations.
   IF \(<\text{Min(Slack time Back Ride Bookings)}\)*. Calculate Savings,
   ELSE Return 0
   Go to Step 4. Number = Number + 1
   ELSE Go to Step 3c.

3. Look in Back Rides List Class 2
   a. FIND booking \( J \) with similar time as booking \( I \).
   IF Found: Calculate Distance between destinations.
   IF \(<\text{Min(Slack time Back Ride Bookings)}\)*. Calculate Savings,
   ELSE Return 0
   Go to Step 3b. Number = Number + 1
   ELSE Go to Step 4.
   b. SELECT booking \( J + 1 \).
   IF time is OK, Calculate Distance between destinations.
   IF \(<\text{Min(Slack time Back Ride Bookings)}\)*. Calculate Savings,
   ELSE Return 0
   Go to Step 3c. Number = Number + 1
   ELSE Go to Step 4.
   c. SELECT booking \( J + 2 \).
   IF time is OK, Calculate Distance between destinations.
   IF \(<\text{Min(Slack time Back Ride Bookings)}\)*. Calculate Savings,
   ELSE Return 0
   Go to Step 3c. Number = Number + 1
   ELSE Go to Step 4.

4. CHECK if used Back rides are not used before. SELECT largest savings;
5. While \( l \leq n_{to} \), Set \( l = l + 1 \). Go to step 1. If \( l = n_{to} \), Go To step 6.
6. Set \( l = 1 \). Repeat Step 1-5 for (Time, Time + \( \eta \)) OR (Time – \( \eta \), Time). If \( l \) reaches \( n_{to} \) again, STOP.

* For the (Time, Time + \( \eta \)) interval, a value of 0.5 hours must be subtracted from the Distance because the departure of the Back ride is planned 0.5 hours later as the arrival of the To ride, resulting in 0.5 hours additional waiting time.
Appendix 8: Dutch law for taxi drivers

Below, the most important points of the Dutch law of vehicle drivers are summarized (Ministerie van Sociale Zaken en Werkgelegenheid, 2014):

- Working over 5.5 hours results in a break time of at least 30 minutes;
- Working over 10 hours results in a break time of at least 45 minutes;
- Breaks last at least 15 minutes;
- Workers need to have a compact resting time of 36 hours each week or 72 hours in each two weeks;
- The maximum working time is 12 hours per shift with a maximum of 60 hours each week. The average working hours per week may not exceed 48 hours in a period of 16 weeks and 55 hours in a period of 4 weeks;
- The rest time between two shifts is at least 8 hours.
Appendix 9: Models applied for Tinker situation:

Situation 1: $m_i = 0.25, f_i = 0.2, \xi_i = 0, \eta = 0$:

The combination problem (parameters are as described in section 5.4):

$$\text{Maximize: } \sum_i \sum_j x_{i,j} s_{i,j}$$

Where

$$\sum_i x_{i,j} + \sum_j x_{i,j} \leq 1, \quad \forall i, j: i \neq j$$

$$x_{i,i} = 1 \text{ if } \sum_i x_{i,j} + \sum_j x_{i,j} = 0, \quad \forall i, j: i \neq j$$

$$x_{i,j} \leq \lambda_{i,j}$$

$$\lambda_{i,j} = 1 \text{ if } !$$:

\[
\begin{align*}
\begin{cases}
\text{at}_i = \text{at}_j, & \text{to rides data} \\
\text{dt}_i = \text{dt}_j, & \text{back rides data} \\
LU_i + LU_j \leq 32 \\
pax_i + pax_j \leq 8 \\
c_i = c_j = 1 \\
t_{\text{comb},i,i,j} \leq t_{i,a_i} + 0,25 \\
\frac{t_{\text{comb},i,i,j}}{t_{i,a_i}} \leq 1,2 \\
s_{i,j} = cd_{i,j} + vd_{i,j} \\
x_{i,j}, \lambda_{i,j} \text{ binary} \\
LU_i, pax_i \in \mathbb{N}
\end{cases}
\end{align*}
\]

When the more comfortable (1, 0.5, 0.5) setting is used, the model is as follows:

$$\text{Maximize: } \sum_i \sum_j x_{i,j} s_{i,j}$$

Where

$$\sum_i x_{i,j} + \sum_j x_{i,j} \leq 1, \quad \forall i, j: i \neq j$$

$$x_{i,i} = 1 \text{ if } \sum_i x_{i,j} + \sum_j x_{i,j} = 0, \quad \forall i, j: i \neq j$$

$$x_{i,j} \leq \lambda_{i,j}$$

$$\lambda_{i,j} = 1 \text{ if } !$$:

\[
\begin{align*}
\begin{cases}
\text{at}_i - \text{at}_j \leq 0,5, & \text{at}_j - \text{at}_i \leq 0,5, & \text{to rides data} \\
\text{dt}_j - \text{dt}_i \leq 0,5, & \text{dt}_i - \text{dt}_j \leq 0,5, & \text{back rides data} \\
LU_i + LU_j \leq 32 \\
pax_i + pax_j \leq 8 \\
c_i = c_j = 1 \\
t_{\text{comb},i,i,j} \leq t_{i,a_i} + 0,5
\end{cases}
\end{align*}
\]

74
Finally, the mixed setting $(0.2; 0.25; 0.5; 0)$ has the following model:

\[
\text{Maximize: } \sum_i \sum_j x_{i,j} s_{i,j}
\]

Where

\[
\sum_i x_{i,j} + \sum_j x_{i,j} \leq 1, \quad \forall i, j: i \neq j
\]

\[
x_{i,i} = 1 \text{ if } \sum_i x_{i,j} + \sum_j x_{i,j} = 0, \quad \forall i, j: i \neq j
\]

\[
x_{i,j} \leq \lambda_{i,j}
\]

\[
\lambda_{i,j} = 1 \text{ if }:
\]

\[
\begin{align*}
& at_i - at_j \leq 0.5, \quad at_j - at_i \leq 0.5, \quad \text{to rides data} \\
& dt_j - dt_i \leq 0.5, \quad dt_i - dt_j \leq 0.5, \quad \text{back rides data} \\
& LU_i + LU_j \leq 32 \\
& pax_i + pax_j \leq 8 \\
& c_i = c_j = 1
\end{align*}
\]

\[
\begin{align*}
& t_{\text{comb},i,i,j} \leq t_{i,a_i} + 0.25, \quad t_{i,a_i} \leq \frac{m_1}{f} = \frac{0.25}{0.2} \\
& t_{\text{comb},i,i,j} \leq t_{i,a_i} + 0.25 \text{ AND } \frac{t_{\text{comb},i,i,j}}{t_{i,a_i}} \leq 1.2 \quad \text{otherwise}
\end{align*}
\]

\[
\begin{align*}
s_{i,j} &= cd_{i,j} + vd_{i,j} \\
x_{i,j}, \lambda_{i,j} &\text{ binary} \\
LU_i, pax_i &\in \mathbb{R}
\end{align*}
\]
The matching problem is as follows, the model itself does not depend on the input for the Combination model (and is therefore the same for all settings as described above), but the results do depend on the outcomes of the Combination model. In the setting \( \eta = 0.5 \), the model is as follows:

\[
\text{Maximize: } \sum_i \sum_j X_{i,j} r_{i,j}
\]

Where

\[
\sum_i X_{i,j} + \sum_j X_{i,j} \leq 1, \quad \forall i, j : i \neq j
\]

\[
x_{i,j} = 1 \text{ if } \sum_i X_{i,j} + \sum_j X_{i,j} = 0, \quad \forall i, j : i \neq j
\]

\[
X_{i,j} \leq \mu_{i,j}
\]

\[
\mu_{i,j} = 1 \text{ if }:
\]

\[
at_i + t_a \leq dt_j \leq at_i + t_a + 0.5
\]

\[
at_i + t_a \leq dt_j + \rho_j
\]

\[
r_{i,j} = d_i + d_j - d_{\text{match},i,j} - d_a
\]

\[
X_{i,j}, \mu_{i,j} \text{ binary}
\]

In the original situation the minimum slack time \((\rho_i, \rho_j)\) is at most \(m_i = 0.25\).

If \( \eta = 0 \), \( at_i + t_a = dt_j \). Therefore the constraint \( at_i + t_a \leq dt_j + \rho_j \) is not necessary anymore and could be removed.
Appendix 10: Savings analysis of simple Excel model

In this section will be investigated how bookings which are combined or matched are located. First the analysis is done for combination. This will be done for the current situation \((f, m, \xi) = (0.2; 0.25; 0)\) and for a situation in which many combinations are possible, \((1; 0.5; 0.5)\). First the original situation is analyzed, bookings are split in groups based on their distance to the airport. 0-10 represents the number of bookings / combinations with a value between 0 and 10 kilometers. In Table 22 is given how many bookings per group are combined, separated for To rides and Back rides. Not combined bookings are left out of scope.

Booking Combined represents the number of bookings for a given category. Minimum Distance to Airport counts the minimum of the distance to the airport of the two combined bookings (if a booking with a distance of 10 kilometer to the airport is combined with a booking with a travel distance of 70 kilometer, the booking with 10 kilometers distance counts). Customer Savings counts the customer savings per group, Vehicle Savings groups the reduction of empty kilometers.

<table>
<thead>
<tr>
<th>Combination</th>
<th>Bookings Combined To rides</th>
<th>Minimum Distance to Airport</th>
<th>Customer Savings</th>
<th>Vehicle Savings</th>
<th>Bookings Combined Back rides</th>
<th>Minimum Distance to Airport</th>
<th>Customer Savings</th>
<th>Vehicle Savings</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-10</td>
<td>23</td>
<td>21</td>
<td>29</td>
<td>0</td>
<td>14</td>
<td>12</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>10-20</td>
<td>75</td>
<td>63</td>
<td>57</td>
<td>0</td>
<td>26</td>
<td>22</td>
<td>23</td>
<td>0</td>
</tr>
<tr>
<td>20-30</td>
<td>36</td>
<td>21</td>
<td>21</td>
<td>8</td>
<td>17</td>
<td>14</td>
<td>14</td>
<td>6</td>
</tr>
<tr>
<td>30-40</td>
<td>22</td>
<td>9</td>
<td>9</td>
<td>38</td>
<td>11</td>
<td>5</td>
<td>3</td>
<td>15</td>
</tr>
<tr>
<td>40-50</td>
<td>15</td>
<td>10</td>
<td>8</td>
<td>65</td>
<td>7</td>
<td>1</td>
<td>2</td>
<td>21</td>
</tr>
<tr>
<td>50-60</td>
<td>22</td>
<td>2</td>
<td>3</td>
<td>7</td>
<td>12</td>
<td>4</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>60-70</td>
<td>17</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>9</td>
<td>2</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>70-80</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>80-90</td>
<td>8</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>90-100</td>
<td>8</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>&gt;100</td>
<td>30</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>All</td>
<td>260</td>
<td>-</td>
<td>21</td>
<td>48</td>
<td>122</td>
<td>-</td>
<td>21</td>
<td>49</td>
</tr>
</tbody>
</table>

The average savings are 21 for customer savings (both To and Back combinations), and 48 (To) and 49 (Back) for the vehicle savings, resulting in totals of savings around 70. For the flexible \((1; 0.5; 0.5)\) situation, the results are given in table 23.
Table 23: Combinations grouped in kilometer slots for the flexible situation

<table>
<thead>
<tr>
<th>Kilometers</th>
<th>Combination To rides</th>
<th>Combination Back rides</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Kilometers</td>
<td>Bookings</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Combined</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Minimum Distance</td>
</tr>
<tr>
<td>0-10</td>
<td>168</td>
<td>160</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10-20</td>
<td>791</td>
<td>567</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20-30</td>
<td>300</td>
<td>118</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30-40</td>
<td>155</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40-50</td>
<td>97</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50-60</td>
<td>146</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>60-70</td>
<td>77</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>70-80</td>
<td>41</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>80-90</td>
<td>21</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>90-100</td>
<td>23</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&gt;100</td>
<td>69</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All</td>
<td>1.888</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The number of combinations is higher, but the pattern is similar. Most combinations are with at least one booking relatively close to the airport (<30 kilometers). Customer savings are on average 11 kilometers for To rides and 12 kilometers for Back rides and Vehicle savings are on average 43 for To rides and 44 kilometers for Back rides, resulting in total savings averages of 54 (To rides) and 56 (Back rides). This is lower as in the current situation described in Table 22. This probably is the result of increased possibilities to combinations, which result in more combinations with customers living close to each other. The bookins with an equal departure / travel time were sorted on postcode, so if more combinations are possible, it is more likely that customers living close are combined.

Now this analysis for the Matching Model will be presented. The situation with \( \eta = 0 \) will be considered for again the original situation and the flexible model. Conclusions with respect to different values of \( \eta \) are similar, since this only extends the current analysis by comparing different sets of rides. For each ride, the distance from the carrier to the customer plus the distance from the destination (airport) back to the carrier are considered. The results are presented in Table 24 for both the current situation and the flexible situation. ‘Number of rides matched’ groups the rides which are matched based on this distance. This is done similarly as for combination in Tables 22 and 23. The value of ‘Minimum distance of matched rides’ takes the minimum of these distances of the two rides which are matched for each match. The vehicle savings for the matches are also grouped.
<table>
<thead>
<tr>
<th>Kilometers</th>
<th>Current (Original) Setting</th>
<th>Flexible Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Number of Rides Matched</td>
<td>Minimum distance of Matched Rides</td>
</tr>
<tr>
<td>0-10</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10-20</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>20-30</td>
<td>168</td>
<td>161</td>
</tr>
<tr>
<td>30-40</td>
<td>307</td>
<td>260</td>
</tr>
<tr>
<td>40-50</td>
<td>605</td>
<td>398</td>
</tr>
<tr>
<td>50-60</td>
<td>275</td>
<td>105</td>
</tr>
<tr>
<td>60-70</td>
<td>321</td>
<td>91</td>
</tr>
<tr>
<td>70-80</td>
<td>131</td>
<td>27</td>
</tr>
<tr>
<td>80-90</td>
<td>95</td>
<td>17</td>
</tr>
<tr>
<td>90-100</td>
<td>41</td>
<td>3</td>
</tr>
<tr>
<td>&gt;100</td>
<td>215</td>
<td>16</td>
</tr>
<tr>
<td>All</td>
<td>2.158</td>
<td>-</td>
</tr>
</tbody>
</table>

The average savings in the current situation are 39 kilometers and 37 kilometers for the flexible situation. Like for combination, this probably is the result of more possible combinations. The vehicle savings are a little lower as the vehicle savings in the combination scenario. This probably is the result of the fact that matched customers do not have to live close to each other, since they do not travel together.
Appendix 11: Savings per Combination / Match of LP model

Here, the savings of the Linear Programming model are analyzed. This analysis is used to determine whether savings are higher per combination or per match. Three situations $(f, m_1, m_2, \xi)$ are considered: the original situation $(0.2; 0.25; 0)$, the mixed situation $(0.2; 0.25; 0.5; 0)$ and the flexible situation $(1; 0.5; 0.5)$.

In Table 25, average values for Combination and Matching are calculated for the Linear Programming model of sections 5.4 and 6.4. For each of the values of the savings, number of combinations/matches and averages the first value is related to combined To rides, the second value to combined Back rides and the third value to matching. It can be concluded that the average kilometer savings are 63 for combined To rides, 59 for combined Back rides and 49 for matching. Differences between the settings are not large.

<table>
<thead>
<tr>
<th>Setting $(f, m_1, (m_2, \xi))$</th>
<th>$\eta$</th>
<th>Savings</th>
<th>Number of combinations/ matches</th>
<th>Average Savings</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(0.2; 0.25; 0)$</td>
<td>0</td>
<td>1.684</td>
<td>584</td>
<td>6.751 24 10 147 70 58 46</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>1.684</td>
<td>584</td>
<td>9.440 24 10 194 70 58 49</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1.684</td>
<td>584</td>
<td>12.436 24 10 241 70 58 52</td>
</tr>
<tr>
<td>$(0.2; 0.25; 0.5; 0)$</td>
<td>0</td>
<td>4.523</td>
<td>1.260</td>
<td>6.604 84 25 144 54 50 46</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>4.523</td>
<td>1.260</td>
<td>9.615 76 20 194 60 63 50</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>4.523</td>
<td>1.260</td>
<td>12.416 76 20 240 60 63 52</td>
</tr>
<tr>
<td>$(1; 30; 0.5)$</td>
<td>0</td>
<td>10.017</td>
<td>2.984</td>
<td>6.406 167 49 140 60 61 46</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>10.017</td>
<td>2.984</td>
<td>10.954 167 49 226 60 61 48</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>10.017</td>
<td>2.984</td>
<td>12.601 167 49 248 60 61 51</td>
</tr>
</tbody>
</table>
Appendix 12: Results of savings calculations for Excel model

In Table 7 of the report, the savings are calculated for various parameter settings \((f, m, \xi)\). Here, a version with more runs is given in Table 26. Values of \(f\) of 0.2; 0.5; 1 and \(\infty\), values of \(m\) of 0.25; 1/3, 5/12 and 0.5 (15, 20, 25 and 30 minutes) and values of \(\xi\) of 0, 0.5 and 1 are considered.

<table>
<thead>
<tr>
<th>(f)</th>
<th>(m)</th>
<th>(\xi)</th>
<th>Economic Class Only</th>
<th>All Bookings</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>0.25</td>
<td>0</td>
<td>155</td>
<td>0.02</td>
</tr>
<tr>
<td>0.2</td>
<td>0.5</td>
<td>0</td>
<td>160</td>
<td>0.02</td>
</tr>
<tr>
<td>0.5</td>
<td>0.25</td>
<td>0</td>
<td>576</td>
<td>0.09</td>
</tr>
<tr>
<td>0.5</td>
<td>1/3</td>
<td>0</td>
<td>620</td>
<td>0.10</td>
</tr>
<tr>
<td>0.5</td>
<td>5/12</td>
<td>0</td>
<td>655</td>
<td>0.10</td>
</tr>
<tr>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
<td>669</td>
<td>0.10</td>
</tr>
<tr>
<td>1</td>
<td>0.25</td>
<td>0</td>
<td>652</td>
<td>0.10</td>
</tr>
<tr>
<td>1</td>
<td>1/3</td>
<td>0</td>
<td>753</td>
<td>0.12</td>
</tr>
<tr>
<td>1</td>
<td>5/12</td>
<td>0</td>
<td>858</td>
<td>0.13</td>
</tr>
<tr>
<td>1</td>
<td>0.5</td>
<td>0</td>
<td>926</td>
<td>0.14</td>
</tr>
<tr>
<td>0.2</td>
<td>0.25</td>
<td>0.5</td>
<td>301</td>
<td>0.05</td>
</tr>
<tr>
<td>0.2</td>
<td>0.5</td>
<td>0.5</td>
<td>312</td>
<td>0.05</td>
</tr>
<tr>
<td>0.5</td>
<td>0.25</td>
<td>0.5</td>
<td>1.108</td>
<td>0.17</td>
</tr>
<tr>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>1.287</td>
<td>0.20</td>
</tr>
<tr>
<td>1</td>
<td>0.25</td>
<td>0.5</td>
<td>1.253</td>
<td>0.19</td>
</tr>
<tr>
<td>1</td>
<td>1/3</td>
<td>0.5</td>
<td>1.447</td>
<td>0.22</td>
</tr>
<tr>
<td>1</td>
<td>5/12</td>
<td>0.5</td>
<td>1.644</td>
<td>0.25</td>
</tr>
<tr>
<td>1</td>
<td>0.5</td>
<td>0.5</td>
<td>1.794</td>
<td>0.28</td>
</tr>
<tr>
<td>(\infty)</td>
<td>0.25</td>
<td>0</td>
<td>655</td>
<td>0.10</td>
</tr>
<tr>
<td>(\infty)</td>
<td>1/3</td>
<td>0</td>
<td>782</td>
<td>0.12</td>
</tr>
<tr>
<td>(\infty)</td>
<td>5/12</td>
<td>0</td>
<td>948</td>
<td>0.15</td>
</tr>
<tr>
<td>(\infty)</td>
<td>0.5</td>
<td>0</td>
<td>1.050</td>
<td>0.16</td>
</tr>
<tr>
<td>(\infty)</td>
<td>0.25</td>
<td>0.5</td>
<td>1.263</td>
<td>0.19</td>
</tr>
<tr>
<td>(\infty)</td>
<td>1/3</td>
<td>0.5</td>
<td>1.495</td>
<td>0.23</td>
</tr>
<tr>
<td>(\infty)</td>
<td>5/12</td>
<td>0.5</td>
<td>1.817</td>
<td>0.28</td>
</tr>
<tr>
<td>(\infty)</td>
<td>0.5</td>
<td>0.5</td>
<td>2.040</td>
<td>0.31</td>
</tr>
<tr>
<td>(\infty)</td>
<td>0.25</td>
<td>1</td>
<td>1.553</td>
<td>0.24</td>
</tr>
<tr>
<td>(\infty)</td>
<td>1/3</td>
<td>1</td>
<td>1.849</td>
<td>0.29</td>
</tr>
<tr>
<td>(\infty)</td>
<td>5/12</td>
<td>1</td>
<td>2.253</td>
<td>0.35</td>
</tr>
<tr>
<td>(\infty)</td>
<td>0.5</td>
<td>1</td>
<td>2.530</td>
<td>0.39</td>
</tr>
</tbody>
</table>
Since bookings with a small distance are limited by $f$ and bookings with a longer distance are limited by $m$, considering only a maximum absolute additional travel time ($m = m_1$) for small distances and considering the maximum additional travel fraction ($f$) and a larger maximum absolute additional travel time ($m = m_2$) for large distances probably leads to more possible combinations. So bookings with a small distance to the destination (travel time less than $\frac{m_1}{f}$) will only be limited by a value of $m_1$ (setting $f$ to $\infty$), and large rides will be limited by $m_2 (> m_1)$ and $f$. The results for a couple of situations are given in Table 27.

<table>
<thead>
<tr>
<th>$f$</th>
<th>$m_1$</th>
<th>$m_2$</th>
<th>$\xi$</th>
<th>Economic Class Only</th>
<th>Number of combinations</th>
<th>Fraction</th>
<th>All Classes</th>
<th>Number of combinations</th>
<th>Fraction</th>
</tr>
</thead>
<tbody>
<tr>
<td>0,2</td>
<td>0,25</td>
<td>0,5</td>
<td>0</td>
<td>659</td>
<td>0,10</td>
<td>1.382</td>
<td>0,13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0,2</td>
<td>0,25</td>
<td>0,5</td>
<td>0,5</td>
<td>1.273</td>
<td>0,20</td>
<td>2.419</td>
<td>0,22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0,2</td>
<td>0,25</td>
<td>0,5</td>
<td>1</td>
<td>1.564</td>
<td>0,24</td>
<td>2.845</td>
<td>0,26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0,5</td>
<td>0,25</td>
<td>0,5</td>
<td>0</td>
<td>744</td>
<td>0,11</td>
<td>1.664</td>
<td>0,15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0,5</td>
<td>0,25</td>
<td>0,5</td>
<td>0,5</td>
<td>1.433</td>
<td>0,22</td>
<td>2.938</td>
<td>0,27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0,5</td>
<td>0,25</td>
<td>0,5</td>
<td>1</td>
<td>1.761</td>
<td>0,27</td>
<td>3.464</td>
<td>0,32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0,25</td>
<td>0,5</td>
<td>0</td>
<td>924</td>
<td>0,14</td>
<td>2.112</td>
<td>0,19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0,25</td>
<td>0,5</td>
<td>0,5</td>
<td>1.794</td>
<td>0,28</td>
<td>3.765</td>
<td>0,35</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0,25</td>
<td>0,5</td>
<td>1</td>
<td>2.229</td>
<td>0,34</td>
<td>4.440</td>
<td>0,41</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix 13: Speed formula determination

As mentioned in section 6.1, data of the distance and meters and the travel time (which are available for executed bookings in the past) are used to determine a formula for the speed. Since these values are only available for distances and times between customers and destinations, a formula is required to predict the speed for other values. There are several possibilities for the formula, five of these are considered:

- Exponential function;
- Linear function;
- Power function;
- Logarithmic function;
- Polynomial function.

In figure 26, distances of the bookings of Tinker are plotted against speed and trend lines are added. These points are given in red.

![Figure 26: Speed Formulas](image)

In reality, the average speed increases with the distance and grows to a limiting value. It can be found that the exponential function has acceptable results for low distances, but has problems with high distances, resulting in speed averages which are not allowed in the Netherlands (maximum speed at highway is 130 km/h). This also holds for the linear function. The polynomial function also does not perform well on high distances, but here, the average speed is decreasing, which is not realistic. The logarithmic formula and the power function do not have these problems. The difference between these functions is small, for small distances the average speed of the logarithmic function is a bit lower (which makes sense, since for rides below 5 kilometers, it is unlikely that some kilometers can be driven on highways. Since the $R^2$ of the logarithmic function is higher (77.7%), that formula will be used. The $R^2$ is quite high and it can be found that the difference between the real value of the speed can be either positive or negative, which is not the case for other formulas. Therefore the logarithmic formula, $y = 17.47 \ln(x) + 12.852$ in which $x$ is the distance in kilometers, will be used.