MASTER

Crowdsourced delivery
the traditional delivery method reinvented

van Cooten, C.

Award date:
2016

Link to publication
Crowdsourced Delivery - The Traditional Delivery Method Reinvented

by
Cas van Cooten

BSc Industrial Engineering — Eindhoven University of Technology 2014
Student identity number 0782288

in partial fulfilment of the requirements for the degree of

Master of Science
in Operations Management and Logistics

Supervisors:
dr. L.P. Veelenturf, TU/e, OPAC
prof.dr. T. van Woensel, TU/e, OPAC
D. van Zanten, CTO, Trunkrs
Subject headings: crowdsourced shipping; pickup and delivery problem; routing and matching problem; ridesharing; city logistics; transportation; distribution; private drivers
Acknowledgement

This thesis is the result of a graduation project that has been conducted at Trunkrs in the completion of the Master Operations Management & Logistics at Eindhoven University of Technology. The research was conducted in the DATAS project of the NWO Vital Logistics program. It was time in which I faced many challenges, and therefore I would like to thank all people that supported me during this project.

First of all, I would like to thank Dimitri Balai and Jan Wijn for giving me the opportunity of graduating within Trunkrs. I thank them for their interest, support, feedback, and the great lunches we had together. Special thanks go out to Dennis van Zanten for supervising my project during these six months. We had some great discussions together and helped each other with the challenges we both faced. Furthermore, I would like to thank all my fellow interns at Trunkrs. Although we could not always help each other, I enjoyed the small talks we had. I wish you all the best of luck for the future.

I could not complete this thesis without the support of two supervisors at the university. Many thanks go out to Luuk Veelenturf, who is my first supervisor. Luuk, I remember that I contacted you at the right moment to ask you to become my mentor. An hour later and it would not be possible. After one and a half year it is safe to say that we have been a perfect match. I would like to thank you for all the support and feedback during this time. Especially thanks for having faith in me to take that extra step in my thesis. Although we met each other scarcely, I would like to thank Tom van Woensel. I appreciate it that you gave me the freedom and full responsibility of my project, and supported if necessary.

Next, I would like to thanks all my friends I met during my study. I remember some great times we had together and I know we will have them in the future as well.

A special thanks goes out to Bruno, Gerrit, Gerwin and Vivette. Without you proofreading my thesis it would not have the same result.

Furthermore, I would like to thank my family, especially my mom and dad. You did not only support me during this project, but you have always been there for me. Sometimes you need a safe home to come back to and let everything go. I appreciated it and I always will.

Last and most certainly not least, I would like to thank my girlfriend, Madelon. You always listened to my struggles and kept me motivated to accomplish the challenges I faced. I know you supported me during this period, and I know you will continue that in the future.

Cas van Cooten
Abstract

Currently, consumer sophistication and expectations place intense pressure on the urban supply chain, including demands for quicker options such as same-day deliveries, narrower time windows, and time guaranteed delivery. However, due to this intense pressure, internet retailers face many logistical challenges. To surmount the pressure consumers put on the logistics service providers, new opportunities need to be investigated. One of these possibilities is crowdsourced delivery. However, crowdsourced delivery is relatively new and not well studied. In this paper, we explore the potential of using excess capacity along existing traffic flows of the crowd to deliver online ordered goods. We define the routing and matching problem and provide an arc-based MILP formulation and a path-based heuristic approach to solve the problem. Both the solution methods incorporate the private drivers’ maximum additional travel time, stop willingness, and the option of transferring goods between drivers. The matching of transportation requests and drivers is a new variant of the pick-up and delivery problem. Numerical results show that crowdsourced shipments result in significant economic and social benefits, depending on the characteristics of the crowdsourced delivery network. For logistics service providers it is potentially more cost-efficient. For the society, it can reduce traffic congestions and carbon dioxide emissions. Moreover, the benefits increase with the increasing number of participating private drivers and parcels.
Management Summary

Due to the growing pervasiveness of technology and the subsequent rise in online retail, the volume of parcel shipments has doubled over the past eight years. Besides, consumers put intense pressure on the logistics service suppliers by demanding “quicker options such as same-day deliveries, narrower time windows, time guaranteed delivery, and smaller individual volumes at no added cost to the consignee” (Goh, 2015, p. 4). Due to the enormous growth in goods ordered online, especially, the last mile delivery faces many challenges to become more cost-efficient and reduce air pollution. Goodman (2005) notes that the last mile delivery comprises 28% of the total transportation costs. Moreover, according to Edenhofer et al. (2014) the transportation sector was accounted for 14% of the carbon dioxide, CO₂, emission worldwide in 2010. Therefore, innovative solutions are more than welcome.

Problem statement

Trunkrs, a young startup company, located in Utrecht, aims to ease the strain on the distribution network caused by the growth in transportation requests. Trunkrs wants to use excess capacity along existing traffic flows of the crowd to deliver online ordered goods. By combining private drivers and parcel shipments, Trunkrs intends to resolve the problem of increasing congestions, and higher pollution levels. Moreover, Trunkrs wants to become more beneficial for shops in the current logistics service market, by promising “same-day delivery for next day prices”. A schematic representation of the proposed Trunkrs’ distribution network can be seen in Figure 1.

Figure 1: Schematic overview of Trunkrs’ distribution network.
Distribution networks featuring private drivers for delivery processes are relatively new in the logistics research field. Hence, before realizing the theoretical idea of Trunkrs we need to create a supportive decision-making tool to answer several critical questions. These questions are: (i) which private driver takes which packages, and (ii) what route does he/she need to take? (iii) When do we use professional drivers? Based on these questions we formulated the following research question:

“How should Trunkrs’ distribution network be operationalized to minimize costs of distributing packages and guarantee same-day delivery?”

Model
To support the decision-making process, we proposed a model that is based on the routing and matching problem. The idea of this model is to decide the optimal match between drivers and parcels. Besides, it will also determine the optimal route for each private driver when taking multiple transportation requests in one trip. We provide an arc-based general MILP formulation for the routing and matching problem. The model incorporates both the professional backup option and the private drivers. In addition, the private driver’s maximum additional travel time, stop willingness, and the option to transfer parcels between drivers is considered. However, due to the high computational complexity we also provide a path-based heuristic to solve the routing and matching problem more efficiently. The distribution process is divided into three trips beforehand, which all need to be executed subsequently by different drivers. The three trips are: the first-mile delivery, the line-haul, and the last mile delivery. Since we split the transportation request route into three pieces, we cannot guarantee to find the optimal solution in some cases. Moreover, the solution approach of the heuristic can be divided into three separate parts that are subsequently solved. In the first part we find all feasible jobs for all drivers by a recursive algorithm (i); thereafter, the optimal route and associated costs savings (ii) are calculated for all jobs performed by a certain driver. Finally, both parts are used to make the (iii) match between drivers and jobs that maximize costs savings is made.

Conclusion and recommendations
The performance of the crowdsourced delivery platform was tested for realistic problem instances for Trunkrs. Numerical results show that the crowdsourced delivery platform is both cost-efficient and beneficial in environmental terms. Based on larger problem instances, roughly €0.25 is saved per parcel delivery. Besides, results have shown that up to 14 percent points on kilometers can be saved by the crowdsourced delivery platform. For larger problem instances this is up to 24 percent points. However, the reliability of the distribution network is highly dependent on the number of available private drivers. This conclusion is endorsed by the results of Archetti, Savelsbergh, and
Speranza (2016); Arslan, Agatz, Kroon, and Zuidwijk (2016); Chen, Mes, and Schutten (2016); Lee and Savelsbergh (2015). However, the quantity is not the only aspect that matters. Other aspects, such as the customer density, region size, and the number of parcels are all aspects that influence the required number of private drivers. Results show that the more drivers that are available, the more cost-efficient the crowdsource delivery platform become. However, not all private drivers that are available are used. This does not imply that less private drivers also have the same results. Instead, the more options the model has, the better the solutions are that can be generated by the routing and matching model. Moreover, the added benefits of the crowdsource delivery platform increase with the increasing number of participating private drivers and parcels.

We recommend Trunkrs to utilize the use of private drivers fully. In order to fully utilize the stop willingness of the private drivers we propose to choose a cost model that has a per parcel incentive. Results clearly show that the transportation costs are less compared to a cost model with a variable compensation per minute. A fixed fee per parcels might also lead to a higher stop willingness and number of participants. The latter is caused by the fact that the payout of a fixed price per parcel is easier to understand than a variable compensation per minute of their detour. However, Trunkrs cannot fully disregard the use of professional drivers. Trunkrs remains dependent on their backup option, the professional drivers, to maintain the high service level and guarantee same-day delivery. Moreover, the stages in the distribution chain that lead to the last mile delivery (i.e., the first-mile delivery and line-haul) are required to be executed by professional drivers, based on the results. The transportation costs during these stages are relatively small, and professional drivers are better suited for this kind of operations. Besides, parcels that remain unmatched during the last mile delivery need to be delivered by professional drivers.

In addition, results suggest it is desirable to analyze the characteristics of the distribution network before implementing the crowdsource delivery platform. For instance, the spatial distribution of the origin and destination of the private drivers and parcels affect the performance of the platform. Hence, Trunkrs might consider collaborating with more service stations or storage services (e.g., locker systems) at convenient locations. This could alleviate the computational pressure on the model, and time synchronization restrictions for parcels and drivers. Besides, more service stations also provide more matching options for the system.
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>List of Figures</td>
<td>vii</td>
</tr>
<tr>
<td></td>
<td>List of Tables</td>
<td>vii</td>
</tr>
<tr>
<td></td>
<td>List of Abbreviations</td>
<td>viii</td>
</tr>
<tr>
<td></td>
<td>List of Nomenclature</td>
<td>ix</td>
</tr>
<tr>
<td>1</td>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1</td>
<td>Problem statement</td>
<td>2</td>
</tr>
<tr>
<td>1.2</td>
<td>Research questions</td>
<td>5</td>
</tr>
<tr>
<td>1.3</td>
<td>Delineation and scope</td>
<td>6</td>
</tr>
<tr>
<td>1.4</td>
<td>Terminology</td>
<td>8</td>
</tr>
<tr>
<td>1.5</td>
<td>Thesis outline</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>Related literature</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>Problem definition</td>
<td>14</td>
</tr>
<tr>
<td>3.1</td>
<td>Formal description</td>
<td>14</td>
</tr>
<tr>
<td>3.2</td>
<td>Mathematical model</td>
<td>17</td>
</tr>
<tr>
<td>3.3</td>
<td>Solution method</td>
<td>21</td>
</tr>
<tr>
<td>3.3.1</td>
<td>The matching problem</td>
<td>21</td>
</tr>
<tr>
<td>3.3.2</td>
<td>The recursive algorithm</td>
<td>23</td>
</tr>
<tr>
<td>3.3.3</td>
<td>The Traveling Salesman Problem with Time Windows and Precedence Constraints</td>
<td>26</td>
</tr>
<tr>
<td>3.3.4</td>
<td>Pre-processing</td>
<td>28</td>
</tr>
<tr>
<td>3.4</td>
<td>Conclusion</td>
<td>28</td>
</tr>
<tr>
<td>4</td>
<td>Instance design</td>
<td>29</td>
</tr>
<tr>
<td>4.1</td>
<td>Parcel generation</td>
<td>30</td>
</tr>
<tr>
<td>4.2</td>
<td>Driver generation</td>
<td>32</td>
</tr>
<tr>
<td>4.3</td>
<td>Base parameters</td>
<td>33</td>
</tr>
<tr>
<td>5</td>
<td>Numerical results</td>
<td>35</td>
</tr>
<tr>
<td>5.1</td>
<td>The first-mile delivery and the line-haul</td>
<td>35</td>
</tr>
<tr>
<td>5.2</td>
<td>Base result and impact of varying number of drivers</td>
<td>36</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Impact of stop willingness and additional travel time</td>
<td>38</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Analysis of various cost models</td>
<td>40</td>
</tr>
<tr>
<td>5.3</td>
<td>Comparison with the traditional delivery method</td>
<td>41</td>
</tr>
<tr>
<td>5.4</td>
<td>Improve computational time</td>
<td>43</td>
</tr>
<tr>
<td>5.4.1</td>
<td>Results of pre-processing on a larger problem instance</td>
<td>43</td>
</tr>
<tr>
<td>6</td>
<td>Conclusion and recommendations</td>
<td>46</td>
</tr>
<tr>
<td>6.1</td>
<td>Research questions</td>
<td>46</td>
</tr>
<tr>
<td>6.2</td>
<td>Recommendations</td>
<td>49</td>
</tr>
<tr>
<td>6.2.1</td>
<td>Recommendations for Trunkrs</td>
<td>49</td>
</tr>
<tr>
<td>6.2.2</td>
<td>Future research</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>References</td>
<td>51</td>
</tr>
</tbody>
</table>
List of Figures

1 Schematic overview of Trunkrs’ distribution network .......................................................... iii
1.1 PostNL’s development of parcel volumes (PostNL annual report 2014) ............................... 1
1.2 Schematic overview of a hub-and-spoke distribution network ........................................... 2
1.3 Timeline of events of Trunkrs’ distribution network ......................................................... 3
1.4 Schematic overview of Trunkrs’ distribution network ......................................................... 4
3.1 A driver travels from his origin(circle) to his destination(square) while fulfilling a task ........... 15
3.2 A driver travels from his origin(circle) to his destination(square) while fulfilling two tasks .................................................................................................................. 16
3.3 Schematic representation of the solution method ............................................................... 21
3.4 Bipartite graph with two drivers (k1 and k2) and three jobs (p1, p2 and (p1, p2)) .................... 22
3.5 Single-task job detour example .............................................................................................. 25
3.6 Combination of two single-task jobs ..................................................................................... 26
4.1 Overview of the Randstad ..................................................................................................... 31
4.2 Overview of customer density in the Randstad ...................................................................... 32
5.1 Problem instance with 100:200 drivers and tasks ................................................................. 37
5.2 Total CPU time in seconds for four base cases ...................................................................... 38
5.3 The effect of maximum additional travel time and stop willingness ..................................... 39

List of Tables

4.1 Base parameter values ........................................................................................................... 34
5.1 Base results for varying number of drivers ........................................................................... 36
5.2 CPU time in seconds ............................................................................................................. 38
5.3 Results for varying maximum additional travel time and stop willingness ............................ 39
5.4 Results for various cost models ............................................................................................ 40
5.5 Payment per private driver difference for various cost models in euros ............................... 41
5.6 Kilometers saved varying number of drivers .................................................................... 42
5.7 Kilometers saved varying maximum additional travel time and stop willingness ................ 42
5.8 Base scenario results with and without pre-processing ......................................................... 43
5.9 CPU time in seconds with and without pre-processing ......................................................... 44
5.10 Results and CPU time for large problem instance ......................................................... 45
## Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DARP</td>
<td>Dial-a-Ride Problem.</td>
</tr>
<tr>
<td>KPI</td>
<td>Key Performance Indicator.</td>
</tr>
<tr>
<td>LNS</td>
<td>Large Neighborhood Search.</td>
</tr>
<tr>
<td>MILP</td>
<td>Mixed Integer Linear Program.</td>
</tr>
<tr>
<td>PDP</td>
<td>Pickup and Delivery Problem.</td>
</tr>
<tr>
<td>REFs</td>
<td>Resource Extension Functions.</td>
</tr>
<tr>
<td>SPA</td>
<td>Shortest Path Algorithm.</td>
</tr>
<tr>
<td>SPP</td>
<td>Shortest Path Problem.</td>
</tr>
<tr>
<td>SPPRC</td>
<td>Shortest Path Problem with Resource Constraints.</td>
</tr>
<tr>
<td>TSP-TWPC</td>
<td>Traveling Salesman Problem with Time Windows and Precedence Constraints.</td>
</tr>
<tr>
<td>VRP</td>
<td>Vehicle Routing Problem.</td>
</tr>
<tr>
<td>VRPPD</td>
<td>Vehicle Routing Problem with Pickups and Deliveries.</td>
</tr>
<tr>
<td>VRPPDTW</td>
<td>Vehicle Routing Problem with Pickups and Deliveries and Time Windows.</td>
</tr>
</tbody>
</table>
Nomenclature

\(A\) Set of all feasible arcs.
\(A\_j\) Arrival time of job \(j\).
\(A_{ki}\) Arrival time of driver \(k\) at node \(i\).
\(A_k\) The additional travel time driver \(k\) is willing to make.
\(A_{pi}\) Arrival time of parcel \(p\) at node \(i\).
\(B_i\) Continuous variable that represents the arrival time of the driver in node \(i\).
\(D\) Set of drivers.
\(D\_j\) Departure time of job \(j\).
\(D_{ki}\) Departure time of driver \(k\) from node \(i\).
\(D_{pi}\) Departure time of parcel \(p\) from node \(i\).
\(J\) Collection of all feasible jobs.
\(J_k\) Collection of all feasible jobs served by driver \(k\).
\(J^U_p\) Collection of jobs that contains task \(p\) during phase \(U\).
\(L_p\) Delivery lead-time of delivery task \(p\).
\(M\) Large number.
\(N\) Set of all origins and destinations.
\(N^P\) Set of origin and destination nodes of all tasks \(P\).
\(P\) Set of delivery tasks (parcels).
\(Q_k\) Stop willingness of driver \(k\).
\(R_k\) Set of all feasible routes for driver \(k\).
\(S\) Set of all service stations.
\(T_k\) Maximum travel time of driver’s trip \(k\).
\(\delta\) The constant vehicle speed in km/h.
\(\gamma\) Compensation per minute for the private driver’s additional travel time due to the detour.
\(c_p\) Delivery costs of parcel \(p\) done by a professional driver.
\(c_{ij}\) Costs of using arc \((i, j)\) for \(i, j \in N\).
\(d_k\) Destination location of driver’s trip \(k\).
\(d_p\) Drop-off location of delivery task \(p\).
\(d_{ij}\) Travel distance between locations \(i\) and \(j\).
\(e_k\) Earliest departure time of driver \(k\).
\(e_p\) Earliest pickup time of delivery task \(p\).
\(l_k\) Latest arrival time of driver \(k\).
\(l_p\) Latest delivery time of delivery task \(p\).
\(o_k\) Origin location of driver’s trip \(k\).
\(o_p\) Pickup location of delivery task \(p\).
\(s_{kj}\) Costs savings of driver \(k\) and job \(j\) compared to the backup option.
\(t_{ij}\) Travel time between locations \(i\) and \(j\).
\(x_{ij}\) Binary decision variable for using arc \((i, j)\) for \(i, j \in N\).
\(x_{kj}\) Binary decision variable for driver \(k\) and job \(j\).
$S_{pki}$  Binary variable if parcel $p$ is picked up by driver $k$ at node $i$.

$W_p$  Binary decision variable if parcel $p$ is delivered by a professional driver.

$Y_{pki}$  Binary decision variable for driver $k$ carrying parcel $p$ and traveling from node $i$ to node $j$.

$Z_{ki}$  Binary decision variable for driver $k$ for traveling from node $i$ to node $j$.

$x_{kij}$  Binary parameter if edge $(i, j) \in N$ belongs to the set of paths of driver $k$. 
1 Introduction

In 2014 PostNL, the biggest distributor of mail and parcels of the Netherlands, shipped 142 million packages and 2,705 million addressed mails across the Netherlands. Figure 1.1 presents the development of parcel volumes of PostNL. In eight years the volume of parcel shipments doubled. The cause of the enormous number of parcels that has been shipped can be related to the spectacular growth of online sales. However, internet retailers still face many logistical challenges to fulfill goods ordered online successfully. At the same time, “the growing pervasiveness of technology and the subsequent rise in online retail, coupled with consumer sophistication and expectations will place intense pressure on the urban supply chain, including demands for quicker options such as same-day deliveries, narrower time windows, time guaranteed delivery, and smaller individual volumes at no added cost to the consignee” (Goh, 2015, p. 4). In order to satisfy the intense pressure consumers put on the logistics service providers, new opportunities need to be investigated. One of these opportunities is crowdsourced delivery. This concept aims to use excess capacity on journeys, that nonetheless takes place, to support delivery operations. This principle is a good example of a bigger trend, called the “sharing economy”. This allows people to enhance the use of resources through redistribution, sharing and reusing excess capacity of goods and services (Arslan et al., 2016). In line with the sharing economy, the delivery by crowd reduces adverse environmental impacts, such as emissions and additional traffic when dedicated delivery vehicles are used (i.e., logistics service providers).

Figure 1.1: PostNL’s development of parcel volumes (PostNL annual report 2014).

---

1.1 Problem statement

Nowadays, most big logistics service providers, such as FedEx, UPS, PostNL and DHL, use a traditional delivery method, called the hub-and-spoke network (Bowen, 2012). In this network, parcels are picked up at the depot of a store and delivered to a hub. This hub is located in the center of a pre-specified region. From this hub, a line-haul delivers the shipments to a centrally located depot. This depot sorts the shipments and afterward, trucks depart to a hub again. At this hub, the shipments are sorted again and assigned to couriers who will take care of the last mile delivery to the customer. Figure 1.2 represents a schematic overview of a hub-and-spoke network. It should not come as a surprise that most relative impact on costs, in both financial and environmental terms, are made during the last mile delivery. Naturally, transporting bulks of packages is always relatively cheaper per package than carrying a few packages. Hence, most costs are made in the last mile delivery. Goodman (2005) notes that the last mile delivery comprises 28% of the total transportation costs.

Moreover, according to Edenhofer et al. (2014) the transportation sector was accounted for 14% of the carbon dioxide, CO₂, emission worldwide in 2010. Therefore, innovative solutions are
more than welcome. Besides, in November 2015, 195 countries signed the Paris Agreement on greenhouse gasses mitigation, adaption, and finance from 2020. In this agreement, the members promised to reduce their carbon dioxide emission “as soon as possible” and do their best to keep global warming “to well below 2 degrees Celsius”\(^2\). So, it should not come as a surprise that improvements in planning techniques could help easing the strain on the environment caused by transportation.

A young startup company, Trunkrs, located in Utrecht, the Netherlands, considers the idea of combining commuters and parcel shipments. Trunkrs wants to resolve the problem of increasing congestions, higher pollution levels and, moreover, wants to become more beneficial for shops in the current logistics service market, by promising “same-day delivery for next day prices”. The concept is that commuters who travel the same route every day can also pick up and deliver customer packages. Ideally, these commuters will be used for the whole distribution network from shop to the recipient. Commuters who travel every day from their hometown to their work office can pick up orders from stores and deliver these to a service station (e.g., depot, hub, gas station, hardware store, etc.) near the highway exit. Subsequently, the line-haul from a service station to service station can also be performed by commuters following the same procedure. Finally, for the last mile delivery, the same method can be used. A commuter pickups the parcels at the service station near his work office or home and deliver these packages to the customers along the way home. The general timeline of a parcel delivery by Trunkrs can be found in Figure 1.3.

![Figure 1.3: Timeline of events of Trunkrs' distribution network.](image)

A schematic overview of Trunkrs’ distribution network is depicted in Figure 1.4. As can be seen in this figure, commuters are used for several processes in the distribution network. For example, the blue and green car both take care of the shipments from the service station to customers (i.e., the last mile delivery), whereas the red car delivers ordered goods from the shop to the service station (i.e., the first-mile delivery). Finally, the yellow car functions as a line-haul between the two service stations, in this case, two gas stations.

It might occur that a customer is not at home to receive a package. Normally these undelivered packages return to the depot and will be delivered the next day. However, this is not an option for commuters. Trunkrs cannot ask commuters to bring back undelivered packages back to the service stations since this is not on their way to home. Trunkrs prevail this problem by introducing a so-called “neighborhood man”. Every neighborhood has a pre-specified person who is willing to

receive undelivered packages. Customers are then informed that they can pick up their package at their assigned neighborhood man when it suits them most. The advantage of this neighborhood man is that these undelivered packages do not return to the service station, the private driver does not have to drive back to this service station, and customers can pick up their packages when it suits them.

![Schematic overview of Trunkrs’ distribution network.](image)

Distribution networks featuring private drivers for delivery processes are relatively new in the logistics research field. Amazon recently started a similar idea in the United States of America, called Amazon Flex. Amazon offers people to sign up as a private driver. In this case, you can choose any available 2, 4, and 8-hour blocks of time to work the same day, or set availability for up to 12 hours per day for the future. Blocks are allocated based on expected volume and availability of delivery partners. Packages can then be picked up at a location near your home. You will receive packages to deliver in a local radius, based on the length of the delivery block you signed up for. Amazon’s idea is very similar to the use of professional drivers, but now only

---

small regions are covered by private drivers. Hence, new traffic is still generated in their concept. Trunkrs differs from Amazon’s approach since they use existing traffic streams. Furthermore, an approach similar to that of Trunkrs is the pilot Walmart started in 2013. Walmart investigated the use of its in-store customers to deliver goods to its online customers on their way home from the store (Morphy, 2013).

As mentioned, Trunkrs’ theoretical concept of crowdsourced shipping is relatively new and not well studied. Hence, Trunkrs desires a model that will support their decision-making process. Currently, several aspects are still open for all kinds of decisions that should be made based on a model and results. These aspects are: (i) which private driver takes which packages, and (ii) what route does he/she need to take? (iii) When do we use professional drivers? To support the decision-making process, we proposed a model that is based on the routing and matching problem. The idea of this model is to decide the optimal match between drivers and parcels. Besides, it will also determine the optimal route for each private driver. However, more in-depth research is required to establish such distribution network, mainly operated by private drivers and professional drivers as a backup option.

1.2 Research questions

Based on the problem statement, we concluded that Trunkrs desires a model that will support their decision-making process and define how the distribution network should be formed. Therefore, the following research question is defined:

“How should Trunkrs’ distribution network be operationalized to minimize costs of distributing packages and guarantee same-day delivery?”

Self-evident, the main goal of Trunkrs is to minimize costs to become more beneficial for shops compared to other logistics service providers. Trunkrs believes this can be accomplished by their proposed idea described above. Another Key Performance Indicator (KPI) that is of interest is the CO\textsubscript{2} emission reduction. As this is a logical consequence of using existing traffic streams instead of generating “new” traffic exclusively for deliveries. Hence, the performance of the crowdsourced delivery platform is analyzed from three different viewpoints: Trunkrs, the private drivers, and the society/environment. The following underlying research questions are supportive to answer the main research question:

1. How many private drivers are necessary to create a reliable distribution network?

Lee and Savelsbergh (2015) investigated that a crowdsource distribution network is highly dependent on the number of available private drivers. Naturally, the more private drivers that are available, the better the system will perform regarding costs. In this case, for each delivery, there
1.3 Delineation and scope

are multiple drivers available. Therefore, for Trunkrs it is necessary to know what the minimal number of private drivers needs to be, to create a reliable distribution network. This depends on several aspects, such as the number of packages that needs to be delivered, the size of the region that needs to be covered, the drop density, etc.

2. What cost model should Trunkrs use for the private drivers?

Since the main research question is to minimize costs it is necessary to know which costs are associated with deliveries done by private and professional drivers. As already briefly mentioned in the introduction, generally most costs are made in the last mile delivery. Private drivers are paid differently than professionals. To improve the tactical decision making of Trunkrs, we should analyze various cost models according to which private drivers are paid. For example, private drivers could receive only a compensation per minute of their detour, or in combination with a fixed price per parcel. Results need to show which cost model is most beneficial for Trunkrs.

3. How should the optimal routes for package deliveries be determined?

Trunkrs’ concept of delivering differs from the traditional method. In the traditional method, the problem is to route vehicles. In the proposed method the problem is to match packages with drivers on already existing traffic streams. Therefore, the problem shifts from routing vehicles to routing packages. Routing problems are researched thoroughly, and several of these problems apply to this specific case (i.e., The Pickup and Delivery Problem and the Shortest Path Problem). Besides, a reliable and optimal matching procedure is key to minimize costs.

4. What is the trade-off for using professional drivers?

Trunkrs’ distribution network mostly relies on private drivers. However, in some cases, it might occur that professional drivers are the better option (i.e., costs are less than when using private drivers). This could be related to several factors, for instance, drop density in a specific region or requests that cannot be matched with private drivers still needs to be delivered to achieve a certain service level. Trunkrs desires to know all these trade-offs.

1.3 Delineation and scope

The above described details on the research consist of routing and matching packages with private drivers. These packages all have their pickup and delivery locations. The pickup location will in most cases be a service station (only in the first-mile delivery this pickup location will be a shop or warehouse). For this step, it is necessary to find the optimal match and routes for a given set of transportation requests, which minimizes the costs. Hence, this forms the delineation of the research. The research shall only focus on the Randstad area since this is the region Trunkrs is
willing to expand to in the near future (before expanding to all of the Netherlands). Furthermore, we will discuss the entire distribution network from a shop (warehouse) to customers performed by private drivers or professional drivers. This distribution process will consist of three parts. The (i) first-mile delivery, the (ii) line-haul and, finally, the (iii) last mile delivery. The first-mile delivery consists of collecting packages from shops and deliver them to service stations (i.e., a many-to-one problem). The line-haul delivers packages from a service station to service station. Finally, the last mile delivery is the first-mile delivery the other way around. In this case, the packages are picked up at service stations and delivered to the customers (i.e., a one-to-many problem). Since the distinction between these three parts is made, we will also examine them individually.

Finally, we consider a deterministic problem instance. Only packages ordered before a predetermined time (i.e., 15:00 hour) will be served the same day as ordered. This is necessary since the algorithm needs computational time to find the optimal match between drivers and packages and all private drivers need to be informed. Hence, all information is known beforehand and therefore the problem is static.

The research has some clear benefits in financial and environmental terms. We assume that is its beneficial for companies because costs will be lower than contracts with professional logistics service providers. Furthermore, Trunkrs exemplifies the sharing economy and operates in a relatively carbon dioxide neutral way.

Finally, some assumptions are made for this research. Currently, there are no private drivers used in the distribution network. We assume that a sufficient number of private drivers are available to perform deliveries. Furthermore, some crucial information on these drivers is necessary for the routing and matching problem. In the first place, we need to know what the origin and destination locations of the private drivers are. Moreover, since the distribution network is restricted with time windows we also want to know the earliest departure time and the latest arrival time of the private drivers. Furthermore, the private drivers need to specify the maximum amount of additional time they are willing to travel. In this case, we know the time windows in which a private driver is available for deliveries. The final constraint a private driver needs to provide is the willingness to stop (i.e., the number of stops a private driver wants to make).

Currently, Trunkrs’ private driver network is not big enough to handle any requests. However, people are signing up to join the network. Hence, the necessary information on private drivers will be generated based on the information we already have at our disposal. Another point of interest are the potential service stations. From these service stations, it is required to know their location. Trunkrs will examine several of these potential service stations before they contact them. So, for our research, the locations of the service stations are based on the potential locations examined beforehand internally by Trunkrs. For the parcel deliveries, we assume that all packages are of relatively small sizes, and all can be delivered by anyone. Therefore, no additional requirements
on vehicle size are necessary, and there are no capacity restrictions of the delivery vehicles. One additional factor that is interesting to investigate is the CO₂ reduction private drivers create compared to professional drivers. The average emission for a private car was 107 grams of CO₂ per kilometer (g/km) in the Netherlands, in 2014 (Algemene Nederlandsche Wielrijders-Bond [ANWB], 2015). For delivery vans, the average is around 165 g/km CO₂. Therefore, we assume that a route done by a professional driver produces on average 54% more CO₂ emission, based on the kilometers saved. A final assumption is made for the neighborhood men. Currently, no one is signed up as neighborhood man. So, again assumptions are made on this. We assume that a neighborhood man is located at the center of a neighborhood. If during the execution of the research people sign up as private driver or neighborhood man this data will be used in the data set. Below, we provide a list of all assumptions that are made for this research.

- Focus on the entire distribution network limited to the Randstad area.
- A sufficient number of professional and private drivers are available to perform deliveries.
- The problem instance is deterministic and static.
- Information on origin and destination locations of the private driver is simulated.
- The earliest departure time and the latest arrival time of the private driver is simulated.
- The stop willingness is based on the stop willingness of the small sample size.
- There are no capacity restrictions on the vehicles.
- Potential service stations examined internally by Trunkrs will be used.
- Professional drivers produce on average 54% grams of CO₂ per kilometer more than private drivers.
- Neighborhood men will be located at the center of a neighborhood.

1.4 Terminology

The discussion about routing and matching problems consists of terminology, which we explain here. There exist four different types of routing and matching problems. These problems can either be static or dynamic, and it can be deterministic or stochastic. All four possible combinations exist in these problems and by type of problem we mean one of these combinations:

- SD (static and deterministic)
- SS (static and stochastic)
- DD (dynamic and deterministic)
- DS (dynamic and stochastic)

Critical in this classification is the definition of both static and dynamic. In this paper, we use the definition by Toth and Vigo (2014) and Psaraftis, Wen, and Kontovas (2016), according
to whom a routing problem is characterized as dynamic if the input on the problem is received and updated during the execution of the tours. If all problem inputs are received before the tour is executed and do not change thereafter, the routing problem is static. "As a general rule, if the problem calls for the determination of a set of preplanned routes that are not reoptimized and are computed from inputs that do not evolve in real time, the problem is static. Conversely, if the routes are reoptimized or if the output is a policy that prescribes how the routes should evolve as a function of those inputs that evolve in real time, then the problem is dynamic" (Psaraftis et al., 2016, p. 7). Furthermore, there can be either deterministic or stochastic information inputs. Stochastic information is input information that has a certain probability or probability distribution. Examples of stochastic inputs are variable demand size of the request or the probability that the customer is present during delivery. Contrarily, a routing and matching problem is deterministic if there are no stochastic inputs. In this case, all information is fixed beforehand and will not change over time.

Besides the above-described classifications, time-dependency plays a significant role in routing problems. In most dynamic routing problem it is assumed that the travel times between pairs of transportation requests are fixed and based on forecasts or historical information. However, travel times might change over the course of the day. This information only reveals in real time as the vehicle travels. This classification is known as time-dependent (Taş, Dellaert, van Woensel, & de Kok, 2014).

Given that the definition of the word “problem” in this paper refers to the abstract problem and is not directly associated with the real-world problem, it is conceivable that a routing problem may be static whereas its associated real-world problem is dynamic. Given an SD problem, it is conceivable that in the real-world problem we may see all kind of dynamic inputs, which may force the driver to alter the predetermined route. Examples of these dynamic inputs are for instance traffic congestion, a road closed due to an accident, or others (Psaraftis et al., 2016).

1.5 Thesis outline

The remainder of this report is organized as follows: In Chapter 2, we discuss the relevant literature. Furthermore, in Chapter 3 the routing and matching problem is well defined. Moreover, we provide a mathematical model and a heuristic approach to solve the problem. In Chapter 4, we define the generation of the experimental setting to test the performance of the crowdsource delivery platform. Next, in Chapter 5, we present our results from our numerical experiments. Finally, the conclusion and recommendations for Trunkrs and future research are given in Chapter 6.
2 Related literature

This chapter addresses the related literature on a more general analysis of distribution networks and the routing problems. For a more detailed literature review on routing problems we refer to Van Cooten (2016).

Generally, two kinds of distribution network analyses can be distinguished (Azzi, Battini, Persona, & Sgarbossa, 2012). First, (i) network flows optimization: in this case, a pre-designed or existing distribution network is considered, and the goal is to optimize the flows of goods through the network. Second, (ii) network design or re-design: in this case, the goal is to choose the best solution/configuration of the facilities within the network to satisfy the goals of the company. The latter is also related to the main problem of Trunkrs. Designing a distribution network involves strategic decision making which influences tactical and operational decisions (Crainic & Laporte, 1997). For example, service station locations, transportation, and inventory decisions, which affect the costs of the distribution network and the quality of service (Ambrosino & Scutella, 2005). Moreover, for the design or re-design of a distribution network, and especially for Trunkrs’ case, these decisions are critical to becoming beneficial. In general, the design of a distribution network consists of six major points (Battini, 2008):

1. Facility location and demand allocation problems: Where to locate facilities (e.g., depots, hubs, service stations, etc.), the number of each type of facilities and how to allocate the product demands to these facilities.

2. Routing Problems: In this case, we speak of the routing problems in its broadest sense. This problem consists of designing the optimal set of routes for a fleet of vehicles to serve a given set of customers.

3. Inventory management decisions: Determine the inventory level stored at each warehouse to satisfy customer demands. This includes safety buffers, replenishment policies, etc.

4. Goods delivery strategies and optimization: Delivery quantity optimization and transportation modalities between facilities of the distribution network, including transportation cost rate computation and transportation outsourcing decisions.

5. Network complexity analysis: Industrial networks grow, both in size and complexity and flexibility enables quick adaptation to their changing needs while ensuring the security of the entire enterprise, but new complexity measures are necessary to study networks growth and development and to understand their real level of competitiveness and performance.

6. Network performances measurement: Identify and measure key dimensions and indexes along which to evaluate the performance of a distribution network.
Above classifications show that designing new distribution networks involve a lot of strategic decisions. However, for the case of Trunkrs, some of these classifications may be relaxed. Since Trunkrs is a logistics service provider it has no control over inventory management decisions of clients. Therefore, the main focus for Trunkrs are the facility locations, routing problems and the goods delivery strategies and optimization. The latter two overlap each other since transportation costs, and outsourcing decisions are necessary to evaluate and optimize the routing problems. Within this research we will only focus on the routing problems and the goods delivery strategies and optimization, because the service station locations are researched by another colleague and are therefore out of scope. However, results on potential locations of this research will be used in our research. Results of our research will help to analyze the complexity and performance of the Trunkrs distribution network. With certain “What-if” scenarios and simulations the distribution network can be tested.

As already mentioned in Chapter 1 the core of Trunkrs’ concept is a Pickup and Delivery Problem (PDP). A PDP constitute an important class of the Vehicle Routing Problem (VRP) in which objects or people have to be transported between origins and destinations. In the traditional setting of the PDP, a dedicated fleet of vehicles is available to satisfy all transportation requests. This is not the case for Trunkrs. Nevertheless, Trunkrs can relate their problem to a significant number of research on PDPs. An extensive overview of most common PDPs can be found in Berbeglia, Cordeau, Gribkovskaia, and Laporte (2007) and Berbeglia, Cordeau, and Laporte (2010). The authors discuss both static and dynamic cases of the PDPs.

Probably the most interesting variant of the PDP, in general, is the Vehicle Routing Problem with Pickups and Deliveries (VRPPD) and the Vehicle Routing Problem with Pickups and Deliveries and Time Windows (VRPPDTW) (Toth & Vigo, 2014). The VRPPD consists of routing a fleet of vehicles to serve all customer requests. The customer requests specify the load, pickup, and delivery locations. Moreover, in the VRPPD vehicles can serve more than one request at the same time. An example where the VRPPD is widely used is in the local area courier services. The problem is \textit{NP-hard}, since it contains the TSP as a special case (Ropke & Pisinger, 2006). Besides the VRPPD, the VRPPDTW has been well studied. In addition to the general VRPPD now time windows are included. These can be either soft or hard constraints. According to Toth and Vigo (2014) the two best performing heuristics for the VRPPDTW are those of Bent and Van Hentenryck (2006) and of Ropke and Pisinger (2006). The objective function of Bent and Van Hentenryck (2006) is first to minimize the number of used vehicles and then the travel costs. The proposed heuristic consists of two stages. The first stage is a simulating annealing algorithm which minimizes the number of routes. Then, a Large Neighborhood Search (LNS) algorithm is applied in the second stage. Experimental results show that the effectiveness of the approach produced many new best solutions on instances with 100, 200, and 600 customers with variable
vehicle fleet sizes. All results were found in much less than the allowed computational time (i.e., 60 minutes for 100- and 200-customer benchmarks, and 90 minutes for 600-customer benchmarks). Ropke and Pisinger (2006) proposed an improvement on this heuristic. The heuristic has generated better solution than those of Bent and Van Hentenryck (2006).

The proposed idea of Trunkrs is closely related to the literature on carpooling. Baldacci, Maniezzo, and Mingozzi (2004) examined a static carpooling problem that aims to match drivers and riders together. Similar to our research, drivers can perform multiple pickups along with their route. However, the authors assumed that all riders and drivers have the same destination. The authors propose both an exact and a heuristic method for the carpooling problem, based on two integer programming formulations of the problem and column generation. They use several constraints (i.e., a capacity constraint, maximum ride time constraint and finally a departure/arrival time constraint) to determine the feasibility of the solution and guarantee passenger convenience.

Most similar to our proposed research is the work done by Arslan et al. (2016). The authors consider a peer-to-peer platform, that similar to us, creates matches between delivery tasks and private drivers. However, they proposed a dynamic variant of this problem and only investigated a network that only exists of private drivers, whereas we want to investigate a mix between private and professional drivers. Arslan et al. propose a rolling horizon framework and develop an exact solution approach to solve the various subproblems. Results of the research suggest that the use of private drivers has the potential to make the last-mile more cost-efficient and environmentally friendly.

In the same vein, Ghilas, Demir, and Van Woensel (2016) investigated the opportunity to use excess capacity in public transportation services, such as bus, train, metro and other rail systems. These public services operate according to predetermined routes and schedules. The authors integrated these public transport services as a part of the freight journey of logistics service providers. The proposed an arc-based mixed integer programming formulation for the integrated transportation system. Experimental results show a significant reduction in operating costs and carbon dioxide emission. Furthermore, results show that the idea of integrated transport systems is also promising in reduce dense traffic in urban areas.

Related to the use of public transportation services is the research by Li, Krushinsky, Reijers, and Van Woensel (2014). Li et al. (2014) studied the idea to integrate parcel distribution with a taxi service. This problem is defined as a Share-a-Ride Problem, an extension of the Dial-a-Ride Problem (DARP). The authors proposed two multi-commodity sharing models. Due to the high computational complexity of the model, the authors chose to develop a model to optimize the insertion of parcels into existing taxi routes. The numerical results show that there exists a trade-off between the profit of a taxi company and the acceptance rate of parcels. Most results were “based on the proved optimal solutions, only in a few cases the solver reached the time
Results show that the performance of the taxi-sharing system is highest in areas resembling cities. Therefore, the proposed idea is quite promising for urban areas. However, results also show that logistics service providers need to be available to ensure that all requests are served.

As already briefly mentioned in the introduction the idea of Trunkrs highly depends on the number of available private drivers. Several authors endorse the importance of a sufficient number of private drivers (Archetti et al., 2016; Arslan et al., 2016; Lee & Savelsbergh, 2015). Hence, this is one of the research questions that is investigated. Lee and Savelsbergh (2015) researched if, within a distribution network of private drivers, there is a role for professional drivers. The authors investigated the benefits, complexity, and costs of employing a small number of dedicated drivers (i.e., professionals) to serve transportation requests that otherwise remain unmatched. They formulated the problem as an integer program to solve real-size problem instances. The results show that these groups of dedicated drivers are necessary to guarantee a high service level.

Finally, a crucial role in optimizing the routes for private drivers is the distance traveled by a driver. Minimizing distance most-likely will also minimize costs. In this case, both the Shortest Path Problem (SPP) and the Shortest Path Problem with Resource Constraints (SPPRC) can be applied. A Shortest Path Algorithm (SPA) that is of interest is Dijkstra’s algorithm (Dijkstra, 1959). At this point, the algorithm of Dijkstra remains one of the most robust approaches to optimally solving the SPP where all arcs are non-negative (Cherkassky, Goldberg, & Radzik, 1996). Originally Dijkstra’s algorithm (Dijkstra, 1959) was developed to solve the single-pair shortest path problem (i.e., find the shortest path between two vertices). However, it can also be used to find the shortest path from a single source vertex to all other vertices in the graph, i.e., the single-source shortest path problem. The single-source shortest path problem was discussed and formalized in his paper. However, the single-pair shortest path problem can still be solved by simply stopping the algorithm once the shortest path from the single source vertex to the single destination vertex has been determined. Moreover, Dijkstra’s algorithm is applicable in both directed and undirected graphs.

A typical extension of the SPP is the SPPRC. In some cases finding the shortest path from a single origin to a single destination is not enough, i.e., some resources need to be taken into account. A resource corresponds to quantity, such as time, the load picked-up by a vehicle, or the duration of a break in a work shift, that varies along a path according to functions, called Resource Extension Functions (REFs) (Irnich & Desaulniers, 2005, p. 34). The SPPRC is related to our research due to the fact that private drivers provide constraints that Trunkrs needs to consider and take into account while optimizing routes. These constraints were stop willingness and maximum additional travel times, as mentioned in Chapter 1.
3 Problem definition

In this chapter, the problem definition of the routing and matching problem is given. In Section 3.1, we formally describe the problem and provide all details with regards to the notations. In Section 3.2, we present a mixed integer linear program for the routing and matching problem. However, due to its computational complexity, we also present a heuristic approach in Section 3.3. This consists of explaining the path-based matching problem, the recursive algorithm, the determination of the optimal route and costs savings, and a pre-processing heuristic for larger problem instances. Finally, in Section 3.4 we conclude this chapter.

3.1 Formal description

In this section, we formally describe the routing and matching problem. We consider a static and deterministic crowdsource delivery platform. This means that all required information (i.e., delivery tasks and driver announcements) is known beforehand and does not change over time. This is different from the online problem Arslan et al. (2016) investigated, in which the platform continuously received new delivery tasks and driver trip announcements over time. Let $N$ denote the set of all origins and destinations for both parcels and drivers. Furthermore, let $d_{ij}$ denote the travel distance, and $t_{ij}$ the travel time between locations $i, j \in N$.

Let $P$ be the set of all delivery tasks. Each parcel $p \in P$ has a pickup location, $o_p$, which can be a warehouse, retail store, or dedicated pickup point (i.e., service station). In most cases, the delivery location, $d_p$, will be the home of the online buyer. It might occur that the delivery location is another dedicated service station, which then functions as a transfer point. This is because the distribution route from the shop to the customer is split up into three trips, i.e., the first-mile delivery, the line-haul, and the last mile delivery. Each parcel has an earliest pickup time, $e_p$, and a latest delivery time, $l_p$, that represent the time that it needs to be delivered. Hence, the parcel has to be delivered within a certain lead-time, $L_p$, where $L_p = l_p - e_p, L_p \geq t_{o_p,d_p}$. Therefore, for each delivery task we can calculate the implicit latest departure time by $l_p = l_p - t_{o_p,d_p}$.

Let $S$ denote the set of all service stations. As already mentioned above a service station can function as a dedicated pickup or delivery location. The latter applies when multiple service stations are included in the whole delivery chain from the retail store or warehouse to the home of the online buyer.

Let $D$ be the set of driver announcements. Each driver’s trip announcement $k \in D$ has an origin, $o_k$, and destination, $d_k$, location. Besides, the driver $k \in D$ has an earliest departure time, $e_k$, and a latest arrival time, $l_k$. The driver also specifies a maximum travel time, $T_k$, where $t_{o_k,d_k} \leq T_k$. Hence, the maximum additional travel time, $A_k$, driver $k$ is willing to make is equal to $A_k = T_k - t_{o_k,d_k}$. Furthermore, the driver specifies his stop willingness, $Q_k$. This implies the
maximum number of additional stops the driver is willing to make, excluding the visit of a service station. The exclusion means three things: (i) during the first-mile delivery the visit of the service station (i.e., the delivery location of the packages) does not count as a stop. Hence, if driver $k$ specifies a stop willingness of $Q_k = 3$, then the driver can pick up at three locations plus the additional stop at the service station. (ii) For the line-haul the first origin of the packages at a service station is not taken into account. Accordingly, the driver can deliver to at most $Q_k$ service stations. The latter also implies for the third part of the distribution network, i.e., the last mile delivery (iii). Again, the pickup of the parcels at the service station is not included in the stop willingness of the driver. When multiple pickups or deliveries are made at the same address, this counts as a single stop. As such, the stop willingness is a hard constraint and reflects the number of different locations the driver is willing to visit. A single task will consist of at most two stops, one at the pickup locations and another one at the delivery location. As mentioned in Section 1.1, Walmart started testing a pilot where in-store customers were asked to deliver online ordered goods to customers. Hence, the pickup location is the same as the origin location of the driver. This situation is schematically represented in Figure 3.1a. The dotted line represents the original trip of the driver, whereas the solid line is the crowdsourced trip. In Trunkrs’ case, this situation will occur sporadically. In most cases the pickup location is different from the driver’s origin, see Figure 3.1b. Therefore, two additional stops need to be made (i.e., one pickup and one delivery).

As already mentioned in Section 1.3, we assume there are no capacity restrictions. Therefore, the restrictions on both time and stop willingness are more restrictive than the capacity restriction. This is because in Trunkrs’ case all packages are of a small size and easily fit in the trunk of a car. If eventually a restriction based on capacity is necessary (e.g., when Trunkrs also wants to transport larger consumer goods, such as furniture), than it is effortless to include additional restrictions based on volume.

In line with the notation used by Arslan et al. (2016), we define a job $j$ as a set of tasks, where
a job can consist of a single task or multiple tasks. An example of a job consisting of two tasks in depicted in Figure 3.2. In this case, the driver starts in his origin, circle $k_1$, and travels to the pickup location where he picks up two parcels, $p_1$ and $p_2$, at, for example, a service station. Then he delivers the two parcels and finally travels to his destination, square $k_1$. The set $J$ denotes the collection of all jobs that are in at least one feasible match. A match between $j$ and driver $k$ is considered feasible if there exists at least one route $r$ in which the driver starts in his origin $o_k$, covers all tasks in job $j$ (i.e., first the pickup location $o_p$, then the delivery location(s) $d_p$) and, finally, end at this destination $d_k$. The route $r$ is feasible when it satisfies the following constraints:

- **Travel time constraint.** The total maximum additional travel time of $r_{k,j}$ is less than or equal to $A_k$.
- **Stop willingness constraint.** The number of unique locations that are visited by driver $k$ in route $r_{k,j}$ is less than or equal to the stop willingness $Q_k$.
- **Time schedule constraints.** Driver $k$ cannot depart before his earliest departure $e_k$ or arrive after his latest arrival time $l_k$. Moreover, every task $p \in P$ cannot be picked up before its earliest pickup time $e_p$ or arrive after its latest arrival time $l_p$.
- **Precedence constraints.** For each task $p \in P$, a driver must pick up the parcel before delivering it. This indicates that the difference in time between the delivery time and the pickup time of task $p \in P$ is greater than or equal to $t_{o_p,d_p}$.
- **Transfer precedence constraints.** For each task $p \in P$, it is possible to transfer them between drivers at service stations. This implies that the drop-off time of task $p \in P$ at the transfer point is less than or equal to the pickup time of task $p \in P$ at the transfer point.

Let $R_k$ denote the set of all feasible routes for driver $k$ and job $j$, i.e., $r_{k,j}$.

Let $R_k$ denote the set of all feasible routes for driver $k$ and job $j$, i.e., $r_{k,j}$.

![Figure 3.2: A driver travels from his origin(circle) to his destination(square) while fulfilling two tasks. Adapted from Arslan et al. (2016).](image)

Within this research, a distinction between private drivers and professional drivers is made, and both are paid differently. Private drivers are commuters. Ideally, the whole distribution network is served by private drivers, but this will not always be the case. Therefore, professional
drivers (i.e., subcontractors) are used as a “backup” solution in case regions are not covered yet by private drivers or for line-haul deliveries between service points. The latter is necessary to ensure a reliable service level and supply service points with deliveries which subsequently are delivered by commuters. Moreover, professional drivers may also be used in regions which have a high drop density or when it is not possible to serve a certain task before its deadline. The drop density is defined as the number of stops (i.e., deliveries) a courier can make within an hour.

Private drivers are paid according to a per-minute fee for the detour. This implies that when a private driver needs an additional 15 minutes, compared to his usual travel time, to deliver consumer packages, he will receive a 15-minute fee. The emergency backup option is more expensive than using private drivers: the per-minute fee the subcontractors is $\alpha \geq 1$ times the costs of the private drivers.

This distribution network aims to minimize the total system delivery costs made by private and professional drivers. Since the platform accepts all jobs, this minimization is equal to maximizing the total profit.

3.2 Mathematical model

In this section, we present an arc-based Mixed Integer Linear Program (MILP) for the routing and matching problem. This model is derived from the mathematical models developed by Chen et al. (2016) and Ghilas et al. (2016). Chen et al. (2016) developed a multi-driver multi-parcel matching problem and proposes a general integer linear programming formulation, which incorporates drivers maximum detour, capacity limits, and the option of transferring parcels between drivers. This problem is very similar to our case. Ghilas et al. (2016) created an MILP which included a scheduled passenger transportation service. In this case, some request continued their journey on a scheduled public transportation system, such as a train, bus or taxi.

The goal of this model is to deliver the parcels on time with minimum overall costs. These costs consist of (i) the shipping costs, in case it is done by a professional subcontractor, and (ii) the compensation for the private drivers additional traveling costs. This model will provide the optimal match between drivers and parcels, the optimal route of each driver and each parcel, and the schedule for the drivers and the parcels delivered by private drivers. The model works in an environment where all information (demands, travel times, etc.) is assumed to be known in advance. Hence, a plan for the whole planning horizon (e.g., one day) can be generated.

The model is defined as a directed graph $G = (N, A)$, where $N$ denote the set of nodes representing all origins, destinations or possible transfer points (i.e., service stations). Let $A$ be the set of arcs that connects two locations. Each arc $(i, j) \in A$ is associated with a distance $d_{ij}$.
and a travel time $t_{ij}$. Furthermore, the following decision variables are included:

$$Z_{kij} = \begin{cases} 
1, & \text{if driver } k \text{ is traveling from node } i \text{ to node } j, \\
0, & \text{otherwise.}
\end{cases}$$

$$Y_{pkij} = \begin{cases} 
1, & \text{if driver } k \text{ carries parcel } p \text{ from node } i \text{ to node } j, \\
0, & \text{otherwise.}
\end{cases}$$

$$W_p = \begin{cases} 
1, & \text{if parcel } p \text{ is delivered by a professional driver,} \\
0, & \text{otherwise.}
\end{cases}$$

$$S_{pki} = \begin{cases} 
1, & \text{if parcel } p \text{ is picked up by driver } k \text{ at node } i \in N, \\
0, & \text{otherwise.}
\end{cases}$$

$$x_{kij} = \begin{cases} 
1, & \text{if the edge } (i,j) \text{ belongs to the set of paths of driver } k, \\
& \text{and the total length of the path is no longer than } T_k, \\
0, & \text{otherwise.}
\end{cases}$$

Furthermore, two decision variables included in this model are $D_{ki}$ and $D_{pi}$, which denote the departure time of driver $k$ or parcel $p$ at node $i \in N$, respectively. Moreover, let $c_p$ denote the costs of delivering parcel $p$ by a professional driver, and $\gamma$ be the fixed compensation per minute for the private driver’s additional travel time due to the detour. Let $A_{ki}$ and $A_{pi}$ be the dependent variable that represent the arrival time of driver $k$ or parcel $p$ at node $i \in N$, respectively. Finally, the parameter $M$ denotes a large number.

$$\min \sum_{p \in P} c_p W_p + \gamma \sum_{k \in D} \left( \sum_{(i,j) \in N} t_{ij} Z_{kij} - t_{o_k,d_k} \right)$$  \hspace{1cm} (1)$$

Subject to

$$\sum_{j \in N} Z_{kij} = 1 \quad \forall k \in D, i = o_k$$ \hspace{1cm} (2)$$

$$\sum_{i \in N} Z_{kij} - \sum_{i \in N} Z_{kji} = 0 \quad \forall k \in D, \forall j \in N \setminus \{o_k, d_k\}$$ \hspace{1cm} (3)$$

$$\sum_{i \in N} Z_{kij} = 0 \quad \forall k \in D, j = o_k$$ \hspace{1cm} (4)$$

$$Z_{kij} \leq x_{kij} \quad \forall k \in D, \forall (i,j) \in N$$ \hspace{1cm} (5)$$
3.2 Mathematical model

\[ \sum_{(i,j) \in N} t_{ij} z_{kij} \leq T_k \quad \forall k \in D \]  
(6)

\[ \sum_{k \in D} \sum_{j \in N} y_{pkij} + w_p = 1 \quad \forall p \in P, i = o_p \]  
(7)

\[ \sum_{i \in N} \sum_{k \in D} y_{pkij} - \sum_{k \in D} \sum_{i \in N} y_{pkji} = 0 \quad \forall p \in P, \forall j \in N \setminus \{o_p, d_p\} \]  
(8)

\[ \sum_{k \in D} \sum_{i \in N} y_{pkij} = 0 \quad \forall p \in P, j = o_p \]  
(9)

\[ y_{pkij} \leq z_{kij} \quad \forall p \in P, \forall k \in D, \forall (i, j) \in N \]  
(10)

\[ s_{pj} \geq \sum_{i \in N} y_{pkji} - \sum_{i \in N} y_{pkij} \quad \forall p \in P, \forall k \in D, \forall j \in N \]  
(11)

\[ \sum_{(i,j) \in N} z_{kij} \leq q_k + 1 \quad \forall k \in D \]  
(12)

\[ a_{kij} \geq d_{ki} + t_{ij} - M(1 - z_{kij}) \quad \forall k \in D, \forall i \in N \setminus \{d_k\}, \forall j \in N \setminus \{o_k\} \]  
(13)

\[ d_{pi} \geq e_p(1 - w_p) \quad \forall p \in P, i = o_p \]  
(14)

\[ a_{pji} \geq l_p(1 - w_p) \quad \forall p \in P, j = d_p \]  
(15)

\[ d_{pi} \geq a_{pi} \quad \forall p \in P, \forall i \in N \setminus \{o_p, d_p\} \]  
(16)

\[ d_{ki} \geq c_k \quad \forall k \in D, i = o_k \]  
(17)

\[ a_{kij} \geq l_k \quad \forall k \in D, j = d_k \]  
(18)

\[ d_{ki} \geq a_{ki} \quad \forall k \in D, \forall i \in N \setminus \{o_p, d_p\} \]  
(19)

\[ d_{pi} - d_{ki} \leq M(1 - \sum_{j \in N} y_{pkij}) \quad \forall p \in P, \forall k \in D, \forall i \in N \setminus \{d_p, d_k\} \]  
(20)

\[ d_{pi} - d_{ki} \geq -M(1 - \sum_{j \in N} y_{pkij}) \quad \forall p \in P, \forall k \in D, \forall i \in N \setminus \{d_p, d_k\} \]  
(21)

\[ a_{k} - a_{pi} \leq M(1 - \sum_{j \in N} y_{pkij}) \quad \forall p \in P, \forall k \in D, \forall i \in N \setminus \{o_p, o_k\} \]  
(22)

\[ a_{k} - a_{pi} \geq -M(1 - \sum_{j \in N} y_{pkij}) \quad \forall p \in P, \forall k \in D, \forall i \in N \setminus \{o_p, o_k\} \]  
(23)

\[ z_{kij}, y_{pkij}, w_p, s_{pj} \in \{0, 1\} \quad \forall p \in P, \forall k \in D, \forall (i,j) \in N \]  
(24)

\[ d_{ki}, d_{pi}, a_{ki}, a_{pi} \geq 0 \quad \forall p \in P, \forall k \in D, \forall i \in N \]  
(25)

The objective function (1) minimizes the overall transportation costs related to the distribution network, which consists of the shipping costs if the delivery is done by a professional driver and the compensation private drivers receive for their detour. Constraint (2) ensure that the drivers take one and only one path. Constraint (3) represent the flow conversion constraints. Constraint (4) make sure that the driver does not return at this origin location. Constraint (5) is a valid...
inequality that guarantees that only edges are used that are feasible for the driver. Constraint (6) is the maximum detour constraint of the driver. By constraints (7) and (8) is ensured that the parcel will either be delivered by a private driver or a professional driver from origin to destination. Constraint (9) guarantees that the parcel will not return at his origin location, this is to eliminate subtours. Scheduled parcels cannot travel without a driver. This is ensured by Constraint (10). Constraint (11) stores the location where the parcels are picked up by drivers. Constraint (12) ensures that the stop willingness of the driver is not violated. The stop willingness is increased by one because the final destination does not count as a stop since this stop nonetheless takes place. The arrival times of drivers are calculated by Constraint (13). Constraints (14) and (15) ensure that each parcel that is delivered by private drivers departs after their corresponding earliest departure time at their origin node, and arrives before their latest arrival time at their destination node. Self-evident, the departure time cannot be earlier than the arrival time at the same node, this is ensured by Constraint (16). Complementary, Constraint (17)-(19) enforce the time constraints for the driver. Constraints (20) and (21) ensure that the departure time of the parcel is equal to the departure time of the driver that will carry it. Constraints (22) and (23) guarantee that the arrival of the parcel is equal to the arrival time of the driver that carried it. Constraints (24) and (25) are domain constraints.

In addition, Chen et al. (2016) developed five valid inequalities that significantly improved the run time. Those valid inequalities also apply to our proposed model since we adapted it from them. Note that these inequalities are not necessary to solve the routing and matching problem.

\[ \sum_j Z_{kij} = 1 \quad \forall k \in D, j = d_k \quad (26) \]
\[ D_{ki} \leq M \sum_j Z_{kij} \quad \forall k \in D, \forall i \in N \setminus \{d_k\} \quad (27) \]
\[ A_{ki} \leq M \sum_j Z_{kij} \quad \forall k \in D, \forall i \in N \setminus \{o_k\} \quad (28) \]
\[ D_{pi} \leq M \sum_k \sum_j Y_{pkij} \quad \forall p \in P, \forall i \in N \setminus \{d_p\} \quad (29) \]
\[ A_{pi} \leq M \sum_k \sum_j Y_{pkij} \quad \forall p \in P, \forall i \in N \setminus \{o_p\} \quad (30) \]

Constraint (26) ensure that the driver will only visit the destination node only once. Constraints (27)-(30) prevent the model from assigning departure and arrival times to nodes that are not visited by the drivers or parcels. Results showed that these valid inequalities reduced the actual size of the MILP and the run time improved up to 11.6\%. However, due to the $M$ in the mathematical formulation, it still requires much time till the model converges.
3.3 Solution method

The model described in Section 3.2 is an extension to the Share-a-Ride Problem, which is an NP-hard problem (Li et al., 2014). The computational complexity, as already mentioned, is very high. Therefore, it is conceivable to develop a heuristic approach to solve the routing and matching problem efficiently. Instead of the arc-based formulation we presented in Section 3.2, the heuristic, presented in Section 3.3.1, is path-based. The proposed heuristic does not provide the optimal solution since the transportation request route from the origin to destination is split into three trips beforehand. All trips need to be executed subsequently by different drivers, and are limited to the earliest pickup time and latest arrival time of each stage. The arc-based MILP does not examine the separated trips individually but considers the distribution network in its whole. Hence, this heuristic will not always provide the optimal solution. In this section, we describe the solution method we use to find the match between jobs and drivers.

The solution method can be divided into three separate parts that are subsequently solved. In the first part we find all feasible jobs for all drivers by a recursive algorithm (i); subsequently, the optimal route and associated costs savings (ii) are calculated for all jobs performed by a certain driver. Both parts provide us with the information required to create the (iii) match between drivers and jobs that maximize costs savings is made. Figure 3.3 visualises the solution method. First, we discuss the path-based matching problem in Section 3.3.1. The matching problem is dependent, as mentioned, on the input determined by the recursive algorithm and the Traveling Salesman Problem with Time Windows and Precedence Constraints (TSP-TWPC), presented in Sections 3.3.2 and 3.3.3, respectively. Finally, in Section 3.3.4 we provide a pre-processing step to reduce the actual size of the routing and matching problem.

Figure 3.3: Schematic representation of the solution method.

3.3.1 The matching problem

Probably the most important part of the solution is the determination of the “optimal” match between jobs and drivers. In line with Arslan et al. (2016); Stiglic, Agatz, Savelsbergh, and
3.3 Solution method

Gradisar (2015), we can model this as a path-based matching problem with side constraints. For each driver \( k \in D \) and each job \( j \in J \) we create a node. An arc between node \( k \) and node \( j \) serve as a feasible match between driver \( k \) and job \( j \). The weight of the arc denotes the costs savings when job \( j \) is served by driver \( k \) when compared to serving it by the backup option. This is determined by the TSP-TWPC. Figure 3.4 illustrates the bipartite graph for an example with two drivers and three jobs. The three jobs are either a single task job (i.e., \( p_1 \) or \( p_2 \)) or a multiple task job (i.e., \( p_1 \) and \( p_2 \) combined). The number above the arc represents the costs savings. In this example, the match between driver \( k_2 \) and job \( p_1 \) and the match between \( k_2 \) and job \( (p_1, p_2) \) are not feasible. The optimal solution, for this example, is to match driver \( k_1 \) with job \( p_1 \) and driver \( k_2 \) with job \( p_2 \) for a total costs saving of 16.

![Figure 3.4: Bipartite graph with two drivers (k1 and k2) and three jobs (p1, p2 and (p1, p2)).](image)

Let \( A \) be the set of all feasible arcs. Let \( J_k \) denote the collection of jobs that can be served by driver \( k \), for \( k \in D \). The set \( J_k \) is determined by the recursive algorithm presented in Section 3.3.2. Furthermore, let \( J^U_p \), \( p \in P \) be the set of jobs that contains task \( p \) during phase \( U \), where \( U = (\text{first}, \text{line}, \text{last}) \) representing the first-mile delivery, line-haul, and last mile delivery, respectively. Then, \( x_{kj} \) represents the binary decision variable that indicate whether the arc is in the optimal matching \( (x_{kj} = 1) \) or not \( (x_{kj} = 0) \). Let, \( D_j \) and \( A_j \) be the departure and arrival time of job \( j \), respectively. The departure time is assumed to be the earliest departure time of job \( j \). The arrival time is calculated by the travel time of the optimal route for job \( j \). This is determined by the TSP-TWPC presented in Section 3.3.3. Moreover, the TSP-TWPC also provides \( s_{kj} \). Let \( s_{kj} \) be the weight of the arc \( (k, j) \) which denotes the costs saving if driver \( k \) is assigned to job \( j \) as compared to the backup option. Then, the matching problem with the objective of maximizing the total costs savings can be formulated as the following integer program (Arslan et al., 2016; Stiglic et al., 2015):

\[
\max \sum_{(k,j) \in A} s_{kj}x_{kj} \quad (31)
\]

22
Subject to

\[ \sum_{j \in J_k} x_{kj} \leq 1 \quad \forall k \in D \quad (32) \]

\[ \sum_{p \in P} \sum_{u \in U} \sum_{j \in J_p^{cu}} x_{kj} \leq 1 \quad \forall p \in P, \forall u \in U \quad (33) \]

\[ \sum_{k \in D} \sum_{j \in J_p^{cu}} x_{kj} A_j \leq \sum_{k \in D} \sum_{j \in J_p^{cu}} x_{kj} D_j \quad \forall p \in P \quad (34) \]

\[ \sum_{k \in D} \sum_{j \in J_p^{cu}} x_{kj} A_j \leq \sum_{k \in D} \sum_{j \in J_p^{cu}} x_{kj} D_j \quad \forall p \in P \quad (35) \]

\[ x_{kj} \in \{0, 1\} \quad \forall (k, j) \in A \quad (36) \]

The objective function (31) aims to maximize the sum of the costs savings of the matches over the backup option. Constraints (32) and (33) assure that each driver is assigned to at most one job, and each job is assigned to at most one driver. Constraints (34) and (35) ensure the correct synchronization between the arrival time and departure time of the parcel \( p \) at a service station. Constraint (36) is a domain constraint. This path-based MILP will be solved by using Gurobi 6.5.1.

### 3.3.2 The recursive algorithm

In this section, we describe the recursive algorithm that determines the feasible jobs to be included in the matching problem. As mentioned above, we do not only need to determine the feasible jobs, but also the order in which we serve each task within a job. Arslan et al. (2016) noted that, in the worst-case scenario, the number of feasible matches for \( p \) tasks and \( k \) drivers is \( O(k^2p) \). In this case, each driver can serve all tasks in one trip. However, every driver specifies his stop willingness, and we have time restrictions. Hence, the number of feasible routes is likely to be far less in practice.

The recursive algorithm is principally based on an observation also used in the papers of Arslan et al. (2016); Stiglic et al. (2015):

**Observation 1:** A job \( j \in J \) does not have a feasible route if there is a set \( j' \) which is a subset of \( j \), denoted by \( j' \subset j \), that has no feasible route.

This observation implies that a job containing multiple tasks is only considered feasible if all these tasks are feasible independently as well. A match between one driver and two tasks is only feasible if both tasks are individually feasible with this driver. Furthermore, this observation also implies that if two tasks are feasible independently, but cannot form a feasible job together (i.e., no delivery sequence is feasible for this driver), all unions that include these two tasks are infeasible for that driver. This observation reduces the number of jobs to be searched by the algorithm.
Based on Observation 1 the feasible jobs for driver \( k \in D \) can be determined by the recursive algorithm given in Algorithm 1. For the algorithm, it is necessary first to determine the single tasks jobs that are feasible for driver \( k \). Subsequently, single-tasks are combined to make jobs of two, three tasks and so forth. Let \( J_k^w \) be the set of jobs with \( w \) tasks that are feasible for driver \( k \). Finally, let \( R_k \) denote the set of all feasible routes for driver \( k \).

**Algorithm 1 Matching Algorithm**

**Require:** The list of \( D \) drivers, \( S \) service stations and \( P \) tasks.

Initialize: Find all feasible pairs of a driver with a single-task job (i.e., one pickup and one delivery). Store these pairs for driver \( k \) in \( J_k^1 \) and construct the route set \( R_k \) for each pair of \((k, j)\).

1: for all \( k \in D \) do
2: \( w \leftarrow 2 \)
3: while \( w \leq Q_k \land J_k^{w-1} \neq \emptyset \) do
4:     for all \( j \in J_k^{w-1} \) do
5:         for all \( \{p\} \in J_k^1 \land p \notin j \) do
6:             if FEASIBLE((\( j, p, k \))) then
7:                 \( R_k' \leftarrow R_k \cup R_{k,j,p} \)
8:                 \( J_k^w \leftarrow J_k^w \cup (j \cup p) \)
9:             else
10:                 the job \((j \cup p)\) is infeasible
11:             end if
12:         end for
13:     end for
14: \( w \leftarrow w + 1 \)
15: end while
16: \( J_k \) and \( R_k \) are determined. Go to next driver
17: end for
18: Continue to determine maximum costs savings for \( \forall j \in J_k \) with the TSP-TWPC.

Algorithm 1 requires the list of drivers, service stations and tasks as an input. It starts with determining all feasible single-task jobs for all drivers. For each driver \( k \in K \) is calculated what the travel time is from his/her origin to destination, i.e., \( t_{o_k,d_k} \). This forms the baseline for the determination of feasible single-task jobs. Subsequently, a single task is included in his original route. So, the pickup location, \( o_p \) and destination location, \( d_p \), are integrated with the original route of driver \( k \). For this new route the travel time, \( r_{k,j} \), is calculated and checked if it is less than or equal to \( A_k \) (i.e., the travel time constraint). Figure 3.5 provides an example of a single-task job detour. The weight of each arc denotes the travel time in minutes. The original route of driver \( k \), from circle \( k_1 \) to square \( k_1 \), had a travel time of 60 minutes. After include task 1 the travel time increase to \( 45 + 10 + 10 = 65 \) minutes. In case the driver specified a maximum additional travel time of \( A_k \geq 5 \) this single-task job is considered feasible for driver \( k \). The same procedure is then repeated for all single-task jobs. All pairs of drivers and single tasks job are stored in the route set \( R_k \). Note that a single task job \( j \), there exist only one route, which is the origin of
the driver and the task followed by the destination of the task and the driver. This is due to the *precedence constraint*, that implies that the packages first must be picked up in order to deliver it to its destination. All feasible single-task jobs for driver $k$ are stored in $J_{1k}^i$.

![Figure 3.5: Single-task job detour example.](image)

Next, Algorithm 1 will create combinations of single task jobs and check if they are feasible for driver $k$. For a chosen driver $k$ we set $w$, so jobs are created that consists of $w$ tasks. Then, at row 3, the algorithm checks if $w$ is less than or equal to the stop willingness, $Q_k$, of the driver. Moreover, there is checked if the set containing $w - 1$ tasks is not empty. The latter is evident since it is impossible to find a job of size $w$ when there were no options for a job with size $w - 1$.

In case any of these two checks is violated then the algorithm will stop and move on to the next driver. Subsequently, at row 4 the algorithm takes one job, $j$, from the set of jobs with size $w - 1$, and includes a single-task job, $p \in J_{1k}^i$, into a new job with size $w$, at row 5. However, at row 5 is also stated that single-task job $p$ should not already be in job $j$. These two steps are merely the implication of Observation 1. For each job the algorithm calls sub-algorithm FEASIBLE, presented in Algorithm 2, at row 6 to check whether the job is feasible. FEASIBLE includes the origin and destination of driver $k$ to job $j$ and single-task job $p$, and calculate the Euclidean distance of that job. The Euclidean distance is uplifted by 30% to reflect the road network in the Netherlands. Also, the distance is converted to a travel time by factor $\delta$, which denotes the constant vehicle speed in km/h. If the additional travel time is less than or equal to $A_k$, then FEASIBLE returns a yes answer. This implies that the job $j \cup p$ is a feasible job for driver $k$.

**Algorithm 2** Feasibility check of the combination of job $j$ and task $p$ for driver $k$

1: function FEASIBLE($j, p, k$)
2: \[
    z \leftarrow \text{false}
\]
3: \[
    j' \leftarrow o_k \cup j \cup p \cup d_k
\]
4: \[
    T_{k,j \cup p} = \frac{1}{\pi} \sum_{i=1}^{w} \text{EUCLIDEAN}(i, i+1), \forall q \in j'
\]
5: \[
    A_{k,j \cup p} = T_{k,j \cup p} - t_{o_k} d_k
\]
6: if $A_{k,j \cup p} \leq A_k$ then
7: \[
    z \leftarrow \text{true}
\]
8: end if
9: return $z$
10: end function

An example of the feasibility check is provided in Figure 3.6. The weight of the arc denotes
the travel time between the two locations in minutes. The original single-task job \( j \) had a travel
time of 65 minutes and the route is represented by the solid black line. Next, Algorithm 1 in-
cludes the single-task job \( p \) to job \( j \), to create a job of size \( w = 2 \). Single-task job \( p \) has the
same pickup location as job \( j \), so no additional kilometers needs to be made for the pickup. The
sub-algorithm FEASIBLE calculates the travel time for the new job, i.e., \( 45 + 10 + 3 + 10 = 68 \)
minutes. Note that in this case the route goes from driver origin \( k1 \rightarrow \) pickup location \( p1 \), \( p2 \rightarrow \)
delivery location \( p1 \rightarrow \) delivery location \( p2 \rightarrow \) driver destination \( k1 \). However, it might be possi-
ble that the route from delivery location \( p2 \) to \( p1 \) is more beneficial than the current route. This
is a subproblem to our solution method and is solved by the TSP-TWPC explained in Section
3.3.3. The additional travel time for this route is equal to 8. If driver \( k \) specified a stop willingness
\( Q_k \geq 2 \) and the maximum additional travel time \( A_k \geq 8 \) then FEASIBLE will return a yes answer
and Algorithm 1 will continue.

Subsequently, the algorithm stores the new feasible job into the job set \( J^w \) and the feasible
route in set \( R_k \). In case the sub-algorithm FEASIBLE returns a no answer then the job \( j \cup p \) is
infeasible and cannot be executed by driver \( k \). Finally, the algorithm is repeated for each driver
\( k \in D \) and ends when either the size of the jobs being searched are larger than the stop willingness
of driver \( k \) or when set \( J^w_{k-1} \) is empty. After \( J_k \) and \( R_k \) are determined for all drivers the solution
method will continue to determine the optimal route for all jobs. How this is determined is
described in the next section.

### 3.3.3 The Traveling Salesman Problem with Time Windows and Precedence Con-
straints

As already mentioned before, for a single-task job there exist only one feasible route. However,
in Figure 3.2 and Figure 3.6, we have seen a multiple-task job. For jobs containing multiple
tasks, there might be multiple feasible routes. Hence, the determination of the optimal route with
the largest costs savings, for corresponding jobs is a subproblem of our matching problem. The
The subproblem is the well-known TSP-TWPC.

The TSP-TWPC is known to be NP-hard (Desrosiers, Dumas, & Soumis, 1986) and has applications in many sequencing and distribution problems like production planning, vehicle routing, and crew scheduling. Although the problem is NP-hard, in our case the number of tasks per job is relatively small (i.e., the stop willingness of the driver). Thus we can solve each problem very fast.

The TSP-TWPC can be formulated as a mixed integer linear program (Mingozzi, Bianco, & Ricciardelli, 1997; Fagerholt & Christiansen, 2000; Arslan et al., 2016). The route must always start at the origin of the driver and end at his destination. Set $N^P$ holds all nodes that correspond to the origins and destinations of all tasks $P$. Recall that $o_p, d_p$ and $d_k, o_k$ represent the origin and destination nodes for task $p$ and driver $k$, respectively. Furthermore, let $N$ be the set of all nodes including the origin and destination of the driver $k$. Finally, let $N^P_+$ and $N^P_-$ denote the nodes corresponding with the origins and the destinations, respectively. Let $x_{ij}$ be the binary decision variable that indicate whether the arc is in the optimal route ($x_{ij} = 1$) or not ($x_{ij} = 0$), for $i, j \in N$. Moreover, let $c_{ij}$ be the costs of using arc $(i, j)$. Finally, the continuous variable $B_i$, for $i \in N$ show the arrival time of the driver in node $i$. The mixed integer program provide us with the costs savings for a job $j$, the optimal route, and the arrival time of the job, $A_j$, at a given end node. The costs savings and the arrival time of the job are both input parameters for the matching problem in Section 3.3.1. The MILP can be formulated as follows:

$$\min \sum_{i,j \in N} c_{ij}x_{ij} - c_{o_k,d_k}$$ (37)

Subject to

$$\sum_{j \in N} x_{ij} = 1 \quad \forall i \in N^P$$ (38)

$$\sum_{j \in N^P_+} x_{o_k,j} = 1$$ (39)

$$\sum_{i \in N^P} x_{i,d_k} = 1$$ (40)

$$\sum_{j \in N} x_{ij} - \sum_{j \in N} x_{ji} = 0 \quad \forall i \in N^P$$ (41)

$$B_{i+n} \geq t_{i,i+n} + B_i \quad \forall i \in N^P_+$$ (42)

$$B_j \geq B_i + t_{ij} - M(1 - x_{ij}) \quad \forall (i, j) \in N$$ (43)

$$e_i \leq B_i \leq L_i \quad \forall i \in N$$ (44)

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in N$$ (45)
3.4 Conclusion

The objective function (37) aims to minimize the total additional travel costs to serve all delivery tasks by the driver. This implies, that the original travel costs of the driver, \( c_{ok,dk} \), are subtracted from the total travel costs. Constraint (38) ensures that all tasks are served exactly once. Constraint (39) make sure that the first location the driver visits is the pickup location of all tasks. This pickup location is the same for all tasks. Constraint (40) ensures that driver ends at his destination. Constraint (41) represents the flow conversion constraint. Constraint (42) make sure that the precedence constraints are not violated. Constraints (43) and (44) are the time window constraints. Constrains (45) and (46) are domain constraints. The MILP will be solved by using Gurobi 6.5.1.

3.3.4 Pre-processing

When problem instances grow, the actual size of the routing and matching problem can become very large. Hence, by introducing a pre-processing heuristic, we improve computational time and reduce the actual size of the routing and matching problem.

It is conceivable when problem instances grow to divide a region into smaller areas that each is solved separately from each other to reduce computational time. In this case, each parcel’s end location is located in a specific area that is associated with a service station. To further reduce the computational time, each driver is assigned to one service station closest to their destination location. Hence, not all feasible service station for each driver needs to be checked for feasible jobs. Naturally, the smaller the region become, the smaller the problem instance are. However, this might negatively affect the solution significantly. The performance of the pre-processing heuristic will be researched.

3.4 Conclusion

In this chapter, we provided the problem definition for the routing and matching problem. Although the problem is solvable by an MILP, the computational complexity is very high. Therefore, we choose to develop a heuristic to solve the routing and matching problem efficiently. The solution method consists of three parts. First, the recursive algorithm determines all feasible jobs for each driver. Next, for all these jobs the optimal route is determined which minimizes the transportation costs. Finally, the optimal match between driver and job is made to deliver all parcels. In case a parcel cannot be delivery by a private driver there exists a “backup” option. This option is taken into account during the executing of the heuristic. Hence, the outcome of the solution method provides us with the maximum costs savings, the match between private/professional driver and parcel, and finally, the optimal route for each driver.
4 Instance design

In this chapter, we describe how we generate the experimental settings for the Trunkrs case. The goal of our numerical experiment is two-fold. First, we would like to answer all relevant research questions presented in Section 1.2. Second, to support answering the main research question, we test the efficiency of the crowdsource delivery platform and the benefits of integrating private drivers into the distribution network.

Note that Trunkrs is still in the startup phase of their business. Therefore, the experimental design is aiming for a realistic case for Trunkrs, one that is likely to happen in the near future. This implies that the problem instances have only very limited service stations, clients, private drivers, and parcels. Therefore, one problem instance will represent an average day for Trunkrs. However, we will also examine a larger problem instance to see how the crowdsource delivery platform and the heuristic performs.

The complexity of the problem is mainly affected by four factors: (i) the number of private drivers, (ii) the number of parcels that need to be delivered, (iii) the stop willingness of the driver, and (iv) the maximum detour of the driver. Naturally, if the number of private drivers and/or the number of parcels increases the more options there are that need to be evaluated. This can have a significant impact on the performance of the algorithm of the routing and matching problem. Along the same line, if the number of stops the driver is willing to make increase, the more options this driver has. The same applies for the maximum detour of the driver. Other factors that also affect the performance of the algorithm are the planning horizon and the spatial distribution of the network. The results are analyzed from three different viewpoints: Trunkrs, the private drivers, and the society/environment. The most important performance indicator of Trunkrs is the total transportation costs spend on delivery by either private or professional drivers. Besides, the matched parcels and matched drivers are a good performance indicator of the model. From the driver’s perspective, we monitor the additional stops, the amount of options, the extra travel time, and the number of parcels the private driver delivers. From the societal point of view, we record the reduction of traffic congestion and the reduction in CO$_2$ emission. This is based on the *kilometers saved* by the crowd deliveries in comparison to distribution network operated fully by professionals.

Hence, the evaluation of the model will be based on these performance indicators:

- **Total costs**: the sum of the compensations of the matched private drivers and the expenses of the use of the backup option.
- **Parcels-matched**: the fraction of tasks that are served by private drivers; the complement represents the percentage of tasks that are served by the backup option.
- **Drivers-matched**: the fraction of private drivers that are used.
4.1 Parcel generation

To test the viability of the crowdsource delivery platform, and monitor the performance of the algorithm, we generate the parcel location based on real world data. This data comes from one of the clients of Trunkrs and consists of roughly 173,000 sales records during one year (i.e., from September 2014 till September 2015) in the Netherlands. These records consist of 6-digit zip codes, customer name, and date of ordering. Since our research focusses on the Randstad, we only consider the sales records in this area, resulting in an area of 100 x 100 kilometers. A visual representation of the Randstad is provided in Figure 4.1a. We divided the Randstad into four areas of interest which each an own service station/hub represented by a red square. The blue, green, purple, and yellow colors represent the regions Amsterdam, Rotterdam, The Hague, and Utrecht, respectively. We generate our instances based on the customer density from the data by clustering the addresses of the customers by their 4-digit zip codes. Figure 4.1b presents a heat map of the customer density within the Randstad. As can be seen, most sales are made in high populated areas, such as Amsterdam, Rotterdam, and The Hague.

We generate the delivery tasks as follows. Our research is limited to only one client who provides us with parcels. This client is the same as the client who provides us with the sales records. This makes the experimental design very relevant for this client. Since the parcels originated from only one client the origin location of the parcel for the first-mile delivery is always the same, i.e., Houten (represented by the black square in Figure 4.1a). Trunkrs chose to collect, consolidate and distribute the parcel at one main hub, located near Utrecht. Therefore the origin location of the parcel for the line-haul is the hub located near Utrecht. From this hub, the
4.1 Parcel generation

(a) Areas in the Randstad
(b) Heat map of order distribution

Figure 4.1: Overview of the Randstad.

Parcels need to be distributed to the service stations/hubs near the end location of the parcel (i.e., the delivery address). Hence, the destination location of the parcel during the line-haul is the service station/hub located geographically closest to the delivery address. Intuitively, if the delivery address is located nearest to the hub in Utrecht, then the line-haul phase is not required. Obviously, this destination location forms the origin location of the final step in the transportation network, the *last mile delivery*. Self-evident, the destination location during this phase is the delivery address of the parcel. Concluding, the parcel will travel over at most four nodes (i.e., the store → the hub near Utrecht → the hub nearest to the delivery address of the parcel → the delivery address of the parcel).

Although, the origin of the parcel may be fixed the destination (i.e., the delivery address) certainly is not. The destination is one of the 1,032 4-digit zip codes based on the customer density, where the exact location is a random point within this region. Of the 173,000 sales records 76,498 (i.e., 44.45% of the total number of orders) parcels were ordered within the Randstad. Therefore, the base value of the number of parcels in the Randstad will be roughly 200 per day. In Figure 4.2a is the distribution per area presented along with the population in these areas. The population in the Randstad account for roughly 40% of the population of the Netherlands, based on the numbers retrieved from the Centraal Bureau voor de Statistiek [CBS] (2013). Figure 4.2b represents the customer distribution per 4-digit zip codes. We see that the 4-digit zip code with the most orders only produces 0.44% of the total orders in the Randstad. This can be related to the high number of 4-digit zip code regions that are included in the Randstad.
4.2 Driver generation

Besides the parcels, we also need to generate data for the drivers. As a baseline for the problem instance, we will consider a 1:1 ratio between parcels and drivers. Therefore, the base value for the number of drivers is also 200. The drivers’ data is dependent on several aspects. Some of the data, such as, the stop willingness, the earliest departure time and latest arrival time will be based on the small sample size of participants that signed up. According to the Kennisinstituut voor Mobiliteitsmanagement commuters travel on average 22 kilometers from home to work office (Kennisinstituut voor Mobiliteitsbeleid [KiM], 2015). This will be used during the simulation of driver routes. However, besides the travel distance, the origin and destination locations of the driver are still unknown. Most companies are located near the four biggest cities in the Randstad, i.e., Amsterdam, Rotterdam, The Hague, and Utrecht. Figure 4.3 is a visual representation of the density of company locations in the Randstad. The darker the color, the more companies are located in this region. Therefore, the origin of the driver (i.e., the work office) is determined the same way as the parcel locations. The destination of the driver is then determined by a random point within a 44-kilometer radius, with an average of 22 kilometers, around the center of the origin location.

The earliest departure time and latest arrival time will also be determined based on the small sample size. We consider a deterministic and static routing and matching problem, and all delivery tasks need to be served during a given period. In Trunkrs’ case, all information about the drivers and parcels is known at 15:00 hour. Trunkrs desire to serve all orders between 18:00 and 22:00. Note that this is not a strict constraint, but it is a guideline. Therefore, all delivery tasks have the same delivery lead-time of four hours. During this time window, most people are at home to receive their ordered goods. To ensure that Trunkrs can serve all delivery tasks, pickup locations (i.e., stores and service station) are accessible by drivers between 15:00 and 21:00. Since the earliest delivery time is 18:00 drivers have a time window of three hours to perform the first two phases of the distribution process, i.e., the first-mile delivery and line-haul. These two phases

![Customer density per area](image1.png)

(a) Customer density per area

![Customer density per 4-digit zip code](image2.png)

(b) Customer density per 4-digit zip code

Figure 4.2: Overview of customer density in the Randstad.
4.3 Base parameters

In the previous two sections, we described the generation of the parcels and drivers for the problem instances. In this section, we will provide the base values for the remaining parameters.

In Section 3.3.2, we mentioned that we use the Euclidean distance formula to calculate the distance between two locations, and we uplift the value with 30% to reflect an urban road network. To related this distance to the travel time we assume a constant vehicle speed, $\delta$, of 60 km/h. This is in the same vain as the parameters used by Chen et al. (2016). Furthermore, the stop willingness, $Q_k$, of the drivers is set to 3. The maximum additional travel time, $A_k$, is set to 15 minutes, and private drivers receive a €25 per hour incentive for their detour (i.e., €0.42 per minute). If a parcel is sent by a professional driver, the delivery costs are 1.1 times higher (based on the hourly wage of the professional drivers) than sending it with a private driver. This is due to that professional drivers can deliver more parcels in one trip than private drivers. A professional driver will not deliver only three parcels on his trip, but this trip will have multiple deliveries. Therefore we assume that transportation costs, for example for these three deliveries, are only 1.1 times higher when using professional drivers. Finally, we assume a handling time of one minute per parcel. This handling time is taken into account during the determination of the feasible jobs for each driver.
It might occur that the delivery attempt fails, e.g., the recipient of the order is not at home. In this case, the drivers, both private and professional, are allowed to deliver it to the neighbors. However, in rare cases, it might occur that even the neighbor delivery attempt fails. In Figure 4.4, we provide the probabilities on the delivery attempts performed by professional drivers. This data is received from Trunkrs during a three month period (i.e., from May 2016 till August 2016). We see that only 1% of the first delivery attempt is not successful and is also not delivered at the neighbors. Hence, 1% of the parcels that is delivered by the crowdsourse delivery platform needs to be delivery to the neighborhood man. So, after the last delivery is made, or during the route, the neighborhood man needs to be visited. This is an unforeseen additional stop and is taken into account by the heuristic. The additional travel time and/or the stop willingness of the driver might be exceeded due to this problem, but the private driver will receive an incentive for his additional travel time.

Table 4.1 provides an overview of base values of our parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
<th>Base value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P$</td>
<td>Number of parcels</td>
<td>200</td>
</tr>
<tr>
<td>$D$</td>
<td>Number of drivers</td>
<td>200</td>
</tr>
<tr>
<td>$L$</td>
<td>Delivery lead-time</td>
<td>240 minutes</td>
</tr>
<tr>
<td>$A_k$</td>
<td>Additional travel time</td>
<td>15 minutes</td>
</tr>
<tr>
<td>$Q_k$</td>
<td>Stop willingness</td>
<td>3</td>
</tr>
<tr>
<td>$\delta$</td>
<td>Constant vehicle speed</td>
<td>60 km/h</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>Compensation for the private drivers’ detour</td>
<td>€0.42 per minute</td>
</tr>
</tbody>
</table>
5 Numerical results

The problem instances are implemented in Python and conducted on an Intel Core i7-2630QM 2.00 GHz, CPU 8GB RAM computer. Both the MILP for the TSP-TWPC and the matching problem are solved by using Gurobi 6.5.1 as an IP solver. All results for each case are based on an average of ten problem instances for that case. We evaluate the performance of the model on the performance indicators listed in Chapter 4.

This chapter is structured as follows: First, the first-mile delivery and the line-haul are examined and discussed in Section 5.1. In Section 5.2, we will review the impact of the number of private drivers on the performance. Furthermore, in Section 5.2.1, we examine the effect of drivers’ maximum additional travel constraint and the stop willingness on the performance of the model. Moreover, we investigate the effect of various cost models, by which private drivers are paid, on the performance, in Section 5.2.2. In Section 5.3, we compare the performance of the crowdsourcing delivery platform to a traditional delivery platform that is fully operated by professional drivers. Finally, in Section 5.4, we test the pre-processing heuristic to improve computational time and give the results of a larger problem instance solved with the heuristic.

5.1 The first-mile delivery and the line-haul

In the ideal situation, the private drivers outweigh the professional drivers. However, in a realistic case this might not be feasible at all. Especially the synchronization of parcel transfers are tough to manage. We want to minimize the waiting time of the private driver and guarantee a high service level. Although we assume there is no capacity restriction for private drivers we cannot assume that they can carry a batch of parcels during the line-haul phase. Besides, as mentioned in Section 4.2, commuters travel an average of 22 kilometers between work and home, and most line-haul connections are much further from each other in our case.

However, the algorithm does take private drivers into account during these two phases. Results have shown that only very few drivers are available during these stages, and when available we limited their capacity to a maximum of 10 parcels at a time. Since in our Trunkrs test case, only one shop supplies the distribution network, all parcel from this shop needed to be shipped to the service stations. If only one or two private drivers are used for a small part of the parcels, it is, of course, cheaper to perform all these shipments with professional drivers. Furthermore, deliveries made in large batches are cheaper than smaller batches executed by more (private) drivers.

Therefore, we choose to perform the first two stages, the first-mile delivery, and the line-haul, with professional drivers. This has several benefits over the use of private drivers: (i) professional drivers can guarantee a higher service level than private drivers, (ii) there are ample professional drivers available, and they have no capacity restrictions (for this number of parcels), and (iii)
professional drivers are cheaper when shipping batches of parcels.

In our experimental design, we have only one shop and four service stations. We assume that from 15:00 the parcels can be picked up at the shop, and the driver supplies all service stations. The optimal route for the line-haul is determined with the traditional TSP, and results show that all service stations are supplied before 17:30. Because the route is performed by professional drivers and never changes, the delivery costs are the same for all problem instances. Hence, we will only evaluate the delivery costs during the last mile delivery.

5.2 Base result and impact of varying number of drivers

In this section, we illustrate the performance of the crowdsource delivery model with a different number of drivers. The results are compared to a baseline result which consists of 200 drivers and 200 parcels. This will be an average number of orders that Trunkrs need to handle in the near future. The stop willingness, $Q_k = 3$, and the maximum additional travel time, $A_k = 15$. Table 5.1 presents the results of the problem instances. We compare the solution for the base scenario with three additional system densities: 50:200, 100:200, and 150:200 drivers and tasks. The 50, 100 or 150 drivers are a randomly chosen subset of the set of drivers used in the base scenario of 200 drivers. For clarification, we decide to set the costs of the base scenario (i.e., 200:200) as a baseline for the cost benchmark, and its cost is normalized to 100.

Table 5.1: Base results for varying number of drivers.

<table>
<thead>
<tr>
<th>Drivers:tasks</th>
<th>50:200</th>
<th>100:200</th>
<th>150:200</th>
<th>200:200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total costs</td>
<td>160.52</td>
<td>125.85</td>
<td>108.92</td>
<td>100.00</td>
</tr>
<tr>
<td>Number of options</td>
<td>611.52</td>
<td>604.08</td>
<td>538.47</td>
<td>504.08</td>
</tr>
<tr>
<td>Parcels-matched (%)</td>
<td>44.12</td>
<td>77.45</td>
<td>90.69</td>
<td>96.08</td>
</tr>
<tr>
<td>Drivers-matched (%)</td>
<td>57.93</td>
<td>65.21</td>
<td>64.79</td>
<td>61.23</td>
</tr>
<tr>
<td>Number of additional stops</td>
<td>2.94</td>
<td>2.43</td>
<td>1.93</td>
<td>1.59</td>
</tr>
<tr>
<td>Detour (%)</td>
<td>36.97</td>
<td>41.35</td>
<td>49.81</td>
<td>48.97</td>
</tr>
</tbody>
</table>

Naturally, we see that the model becomes more cost-efficient when the number of available private drivers increase. This improvement is mainly due to a lower number of delivery tasks served by professional drivers. The decrease in Number of options per driver is related to the fact that a subset is used for the selection of available drivers. Since the set is chosen randomly, it could happen that a “more beneficial” set of drivers is selected. Moreover, the number of options per driver is very dependent on locations of both the parcels, and drivers’ origin and destination. The percentage of Parcels-matched increase from 44.12% in the 50:200 case to 96.08% for the 200:200 case. However, we see that the percentage of Drivers-matched in the first case increases and then decreases. This can also be related to the fact that the subset of drivers is selected randomly.
The decrease is also caused to the increase in available private drivers. Since more drivers are available for the same amount delivery tasks fewer drivers are necessary. This is confirmed by the fact that the Number of stops per driver decrease when the number of available drivers increases. Furthermore, Table 5.1 also show that the Detour increase along with the number of available drivers. Except for the last two cases, here is stays roughly equal. This increase is mainly due to increase in available drivers and the fact that professional drivers are assumed to be 1.1 times more expensive. Since more private drivers are available for “longer” routes, the detour becomes larger. The routing and matching model seeks for the maximum costs savings and fully utilize the benefits of the private driver.

In Figure 5.1, we see an example of a problem instance with consists of 100 drivers and 200 tasks. In this case, 50 parcels were not matched and needed to be delivered by professional drivers. The orange squares represent the matched parcels, and purple squares are the unmatched parcels. The driver’s origin and destination locations are represented by a yellow and green square, respectively. We see that most parcels that are not matched are located in areas where drivers do not live. Especially around Haarlem we see a cluster of unmatched parcels. However, because most of these unmatched parcels are clustered, delivery costs are minimal since the travel distances between deliveries is relatively small. Again, we see that most work offices are located near the four biggest cities and that the driver’s destinations are equally dispersed over the Randstad.

Table 5.2 presents the time that is required to solve the routing and matching problem for the four cases. As expected the determination of the feasible routes and optimal routes increase with the number of drivers. The computational time grows slightly linear when the number of drivers
increases linearly as well.

Table 5.2: CPU time in seconds.

<table>
<thead>
<tr>
<th>Tasks:driver</th>
<th>50:200</th>
<th>100:200</th>
<th>150:200</th>
<th>200:200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recursive algorithm</td>
<td>39.44</td>
<td>75.87</td>
<td>120.60</td>
<td>152.92</td>
</tr>
<tr>
<td>TSP-TWPC</td>
<td>202.78</td>
<td>404.10</td>
<td>557.31</td>
<td>694.94</td>
</tr>
<tr>
<td>Matching problem</td>
<td>45.03</td>
<td>145.95</td>
<td>286.54</td>
<td>408.72</td>
</tr>
<tr>
<td>Total time</td>
<td>287.24</td>
<td>625.92</td>
<td>964.45</td>
<td>1256.58</td>
</tr>
</tbody>
</table>

This effect is visually presented in Figure 5.2. Although the computational times are manageable for these cases, it suggests that some pre-processing can improve the computational time significantly. The main reason for this is that during the matching problem all feasible jobs of each driver are included, which are all investigated by the TSP-TWPC. An example of an improvement is discussed in Section 3.3.4. The result of this pre-processing is discussed in Section 5.4.

Besides introducing a new pre-processing heuristic it might also suffice to use a rolling horizon approach similar to the proposed idea of Arslan et al. (2016). When considering a rolling horizon, the algorithm does not need to re-construct each feasible match from the beginning. For each new rolling iteration, i.e., a new delivery request arrives, the algorithm updates the set of feasible jobs and re-calculate the optimal match between drivers and parcels.

5.2.1 Impact of stop willingness and additional travel time

We examine the effect of the driver’s maximum additional travel time and the stop willingness on the performance of the routing and matching problem. We vary the maximum additional travel time, $A_k$, from 10 up to 20 minutes. Besides, we vary the stop willingness, $Q_k$, from 1 to 3. For every varying parameter value, the set of drivers and parcels remained the same. Results are based
5.2 Base result and impact of varying number of drivers

NUMERICAL RESULTS

on ten simulations and shown in Table 5.3. We can conclude that the performance of the system improves with the increase in additional travel time of the drivers. Furthermore, we see that the effect is even bigger when the stop willingness of the driver also increases. Again, the baseline results of the 200:200 drivers and tasks is considered as a guideline. With a stop willingness of 1, we see that the total costs decrease with 85.10% from 166.92 to 81.82. For a stop willingness of 3, we see that the effect is even larger, it decreases with 105.41% from 153.26 to 47.85. Figure 5.3a represent the effect on the total costs. We see that the marginal benefits increase with the stop willingness. The main reason for this is that the stop willingness of the driver becomes a less strict constraint on the maximum additional travel time. This phenomenon is also justified if we see the results on Number of additional stops. The additional stops do not increase as much when the stop willingness increases from 2 to 3, compared to the increase from 1 to 2. Note that the additional stops for a stop willingness of 1 are greater than one. This is due to that some parcels cannot be delivered to the recipient, but are delivered at the neighborhood man.

Table 5.3: Results for varying maximum additional travel time and stop willingness.

<table>
<thead>
<tr>
<th>Maximum additional travel time ($A_k$)</th>
<th>Stop willingness ($Q_k$)</th>
<th>10 min.</th>
<th>15 min.</th>
<th>20 min.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Total costs</td>
<td>166.92</td>
<td>156.33</td>
<td>153.26</td>
<td>128.51</td>
</tr>
<tr>
<td>Number of options</td>
<td>7.69</td>
<td>41.39</td>
<td>123.38</td>
<td>13.60</td>
</tr>
<tr>
<td>Parcels-matched (%)</td>
<td>49.51</td>
<td>72.06</td>
<td>80.88</td>
<td>65.20</td>
</tr>
<tr>
<td>Drivers-matched (%)</td>
<td>47.47</td>
<td>46.23</td>
<td>45.07</td>
<td>63.53</td>
</tr>
<tr>
<td>Number of additional stops</td>
<td>1.06</td>
<td>1.60</td>
<td>1.83</td>
<td>1.05</td>
</tr>
<tr>
<td>Detour (%)</td>
<td>26.40</td>
<td>24.43</td>
<td>24.40</td>
<td>49.60</td>
</tr>
</tbody>
</table>

In Figure 5.3b and Table 5.3, we see the effect of the maximum additional travel time and stop willingness on the parcels that are matched with private drivers. The effect is less pronounced than compared to the effect on the total costs. The difference between a maximum additional travel time of 15 and 20 minutes are minimal. Only very few parcels remain to be delivered by professional drivers.

Figure 5.3: The effect of maximum additional travel time and stop willingness.
5.2.2 Analysis of various cost models

Private drivers are paid differently than professional drivers. However, we can also change the way in which private drivers are paid. In the base scenario, the private drivers receive a compensation for their detour of €0.42 per minute. Furthermore, we assume that the costs of using a professional driver are 1.1 times higher than the private drivers. In this section, we investigate the effect of various cost models on the performance of the crowdsource delivery model.

We compare the results of two additional cost models. In the first cost model, the private driver receives a fixed €1 for each delivery that he delivers and €0.22 per minute of his detour (i.e., €13 per hour). This implies that in the base scenario (i.e., $Q_k = 3$ and $A_k = 15$) the private drivers can receive a maximum of €6.25, which is the same maximum as the cost model used in the base scenario (i.e., €0.42 per minute of the detour). In the second cost model, we assume that the private driver only receives a fixed price per delivery of €2.

Table 5.4: Results for various cost models.

<table>
<thead>
<tr>
<th>Cost model</th>
<th>Variable</th>
<th>Fixed and variable</th>
<th>Fixed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total costs</td>
<td>100.00</td>
<td>96.68</td>
<td>83.97</td>
</tr>
<tr>
<td>Number of options</td>
<td>504.08</td>
<td>504.08</td>
<td>504.08</td>
</tr>
<tr>
<td>Parcels-matched (%)</td>
<td>96.08</td>
<td>95.10</td>
<td>94.61</td>
</tr>
<tr>
<td>Drivers-matched (%)</td>
<td>61.23</td>
<td>33.17</td>
<td>30.96</td>
</tr>
<tr>
<td>Number of additional stops</td>
<td>1.59</td>
<td>2.94</td>
<td>2.97</td>
</tr>
<tr>
<td>Detour (%)</td>
<td>48.97</td>
<td>31.76</td>
<td>25.52</td>
</tr>
</tbody>
</table>

Results are shown in Table 5.4. We see that both cost models are more cost-efficient than the base cost model. Especially, the decrease in total costs for a fixed cost model is significant. This is mainly due to the utilization of the capacity of the private drivers. For both additional cost models, we see that the Drivers-matched is a lot less compared to the base scenario (i.e., nearly 50 percent less private drivers are used). Subsequently, we see that the stop willingness of the drivers is almost fully utilized. This can be related to the fixed price per delivery, i.e., it is more beneficial to match more parcels with one driver. However, we also see that the number of parcels that remain unmatched increase, but this is not a significant increase. The decrease in detour can also be related to the fixed price per delivery. In the base case scenario, the routing and matching problem focusses more on the utilization of the maximum additional travel time. Shipments that require more travel time are more beneficial since the costs savings are higher in these cases. Hence, unmatched parcels are mostly located near a service station. With the other cost models, the focus shifts to the stop willingness of the drivers.

In Table 5.5, we see the difference between the payments per private driver for the various cost models. Although the total costs are lowest for the “fixed” cost model, we see that the average
### 5.3 Comparison with the traditional delivery method

To evaluate the performance of our crowdsource delivery platform we compare it to a traditional delivery platform. This traditional delivery platform is fully operated by professionals, and we compare the results of this platform to the results of our proposed model. Professional drivers lack the stop willingness and maximum detour constraints. However, the deliveries still need to be made before their deadline, and we assume the professional drivers are only allowed to travel four hours continuously (the same amount as the delivery lead-time). Since both the first-mile delivery and the line-haul are made by professionals, the only difference is made during the last mile delivery. Each service station covers a specified region (see Figure 4.1a). Hence, all the parcels are assigned to the service station that serves the parcels’ delivery location. The line-haul ensures that all parcels will be delivered to the assigned service station, and from there the professional drivers will deliver the parcels.

The problem then becomes a classical VRP. We assume that ample professional drivers are available at the service stations and that there are no capacity restrictions. Moreover, after the last delivery is made the drivers need to return to the service station. However, the travel time from the last location to the service station is not included in the travel time constraint of four hours. The VRP is solved according to the savings algorithm by Clarke and Wright (1964). The results are based on ten deterministic and static problem instances, each consisting of 200 parcels. Furthermore, we assume that the average vehicle speed for the professional drivers is 30 km/h, and the Euclidean distance between two locations is uplifted with 30% to reflect an urban road network. Again, a handling time of one minute per parcel is used.

### Table 5.5: Payment per private driver difference for various cost models in euros.

<table>
<thead>
<tr>
<th>Cost model</th>
<th>Variable</th>
<th>Fixed and variable</th>
<th>Fixed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>4.25</td>
<td>3.73</td>
<td>4.00</td>
</tr>
<tr>
<td>Average</td>
<td>5.22</td>
<td>5.31</td>
<td>5.94</td>
</tr>
<tr>
<td>Maximum</td>
<td>5.83</td>
<td>5.38</td>
<td>6.00</td>
</tr>
</tbody>
</table>

The payment per private driver is higher than the maximum of both other cost models. The fact that the total costs are lower can be related to two reasons. First, the number of Drivers-matched is less than the other two instances. Secondly, since we assume that professional drivers are 1.1 times more expensive than private drivers, the more we pay private drivers, the higher the costs savings are. This could be considered as a flaw in the solution method. If we compare the “variable” cost model to the “fixed and variable”, we see that both the minimum and maximum are roughly €0.50 lower for the fixed and variable cost model. Although the average is higher than the variable cost model, the reduction in total costs can be related to the choice of the cost model.
5.3 Comparison with the traditional delivery method

Results show that on average 26.6 professional drivers are necessary to deliver 200 parcels. Hence, the average number of parcels a professional driver delivers during a route is 7.52. Naturally, this is significantly more compared to the private drivers’ maximum of 1.83 parcels. Table 5.6 show the total kilometers saved by the private drivers compared to the kilometers made by the traditional delivery system. For this benchmark, the base parameter values (see Chapter 4, Table 4.1) are used. We can conclude that the kilometers saved increase with the number of available private drivers, i.e., from 2.64% to 9.11% when the drivers increase from 50 to 200 drivers.

Table 5.6: Kilometers saved varying number of drivers.

<table>
<thead>
<tr>
<th>Drivers:tasks</th>
<th>50:200</th>
<th>100:200</th>
<th>150:200</th>
<th>200:200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kilometers saved</td>
<td>2.64%</td>
<td>6.11%</td>
<td>8.07%</td>
<td>9.11%</td>
</tr>
</tbody>
</table>

In extension to the varying number of drivers, we also investigated the effect of the drivers’ constraints on kilometers saved. Table 5.7 show the results of the kilometers saved with varying maximum additional travel times and stop willingness. For these benchmarks, we use 200 parcels and 200 private drivers. We see that the kilometers saved increase with the drivers flexibility. This is mainly due to increase in the detour that private drivers make. From Table 5.3, we see that the detour significantly increase when the flexibility of the driver increases. Private drivers will deliver parcels that are further from their original route, which otherwise needs to be made by professional drivers. Therefore, more kilometers are saved by the crowdsource delivery system.

Table 5.7: Kilometers saved varying maximum additional travel time and stop willingness.

<table>
<thead>
<tr>
<th>Maximum additional travel time (A_k)</th>
<th>Minimum</th>
<th>10 min.</th>
<th>Minimum</th>
<th>15 min.</th>
<th>Minimum</th>
<th>20 min.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stop willingness (Q_k)</td>
<td>1 2 3</td>
<td>1 2 3</td>
<td>1 2 3</td>
<td>1 2 3</td>
<td>1 2 3</td>
<td></td>
</tr>
<tr>
<td>Kilometers saved</td>
<td>2.30%</td>
<td>3.49%</td>
<td>3.93%</td>
<td>5.49%</td>
<td>8.67%</td>
<td>9.11%</td>
</tr>
</tbody>
</table>

These results show that the crowdsource delivery system travel fewer kilometers than the traditional delivery method. The crowdsource delivery platform surpass the traditional delivery method on kilometers saved from 2.30% to 14.07% based on the private drivers’ constraints. This implies that the benefits for the society are greater as well. From a societal point of view, the crowdsource delivery platform result is less traffic and congestions, and it will lead to a reduction in CO₂ emission. Moreover, we have already shown that the crowdsource delivery platform is more cost-efficient than the traditional method. However, the scale of cost-efficiency is highly dependent on the number of available private drivers. The less private drivers there are and stricter their constraints are, the less economic benefits the system has.
5.4 Improve computational time

In Section 3.3.4, we introduced a pre-processing heuristic to improve computational times and reduce the actual size of the routing and matching problem. In Section 5.2, we see that computational times are manageable for a scenario that is likely to occur in the near future. However, when instances grow, Trunkrs might consider improving the computational time by introducing the pre-processing heuristic. Figure 4.1a provides us with an example of four areas in the Randstad which are covered by one service station.

We investigated the proposed pre-processing step for the base problem instance of 200 parcels and 200 private drivers. Furthermore, the base parameters values (see Chapter 4, Table 4.1) are used. The Randstad is divided into the four areas described before and each one is covered by one service station. The parcels are assigned to a service station based on their delivery location. The results are shown in Table 5.8. The results from the problem instance with pre-processing are based on the results of the four areas.

Table 5.8: Base scenario results with and without pre-processing.

<table>
<thead>
<tr>
<th></th>
<th>Without pre-processing</th>
<th>With pre-processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total costs</td>
<td>100.00</td>
<td>103.71</td>
</tr>
<tr>
<td>Number of options</td>
<td>504.08</td>
<td>423.98</td>
</tr>
<tr>
<td>Parcels-matched (%)</td>
<td>96.08</td>
<td>93.63</td>
</tr>
<tr>
<td>Drivers-matched (%)</td>
<td>61.23</td>
<td>63.47</td>
</tr>
<tr>
<td>Number of additional stops</td>
<td>1.59</td>
<td>1.50</td>
</tr>
<tr>
<td>Detour (%)</td>
<td>48.97</td>
<td>43.70</td>
</tr>
</tbody>
</table>

We observe an increase of 3.71\% in the total costs. This is mainly due to the decrease in Parcels-matched and increase in Drivers-matched. Moreover, we see a decline in the average Number of options per driver. This is related to the fact that each parcel is assigned to a specific area and combinations with other areas are not possible, which otherwise could be more beneficial. However, the reduction in computational time is significant. Results are shown in Table 5.9. The total computational time decreases with 46.36\% from 1256.58 to 674.06 seconds. Especially, the reduction in time of the MILP for the matching of jobs and drivers is enormous. There are much fewer options to be checked, so the actual size of the MILP is decreased significantly. Since the increase in total costs is minimal compared to the gain in computational time, this heuristic prove to be a solid improvement to the core routing and matching problem.

5.4.1 Results of pre-processing on a larger problem instance

In this section, we examine the performance of the pre-processing heuristic on a larger problem instance. The problem instance is based on the sales records of the client during a one-week
Table 5.9: CPU time in seconds with and without pre-processing.

<table>
<thead>
<tr>
<th></th>
<th>Without pre-processing</th>
<th>With pre-processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recursive algorithm</td>
<td>152.92</td>
<td>66.68</td>
</tr>
<tr>
<td>TSP-TWPC</td>
<td>694.94</td>
<td>514.96</td>
</tr>
<tr>
<td>Matching problem</td>
<td>408.72</td>
<td>92.42</td>
</tr>
<tr>
<td>Total time</td>
<td>1256.58</td>
<td>674.06</td>
</tr>
</tbody>
</table>

period in August. During this period 925 orders were made in the Randstad. We assume that the one-week period is ordered in one day, to simulate a larger problem instance. Hence, 925 parcels need to be delivered in one day. These sales records are divided among the four areas which resulted in 341, 159, 213 and 212 parcels in the areas Amsterdam, Rotterdam, The Hague, and Utrecht, respectively. The stop willingness and maximum additional travel time are set to 2 and 10 minutes, respectively. The ratio between the private drivers and parcels is set to 1:1. Table 5.10 shows the performance of the crowdsource delivery platform for a larger problem instance. We see that the \textit{Total costs savings} are the largest for Amsterdam. However, proportionally to the number of parcels the most gain is made in Rotterdam. Moreover, we see that for Rotterdam all parcels are delivered by private drivers, whereas the other regions are around 70-80 percent points. If we look at the \textit{Number of options} per driver we see that Amsterdam has by far the most options per drivers. This is mainly due the dense area where parcels are located in the city of Amsterdam, which are relatively close to the driver’s origin locations. However, this also results in a much larger computational time. All other areas are solved within 10 minutes, but Amsterdam nearly takes 50 minutes to solve. Hence, the city of Amsterdam might need a further separation into smaller areas to improve computational times.

Furthermore, we can conclude that the \textit{kilometers saved} increase with the number of parcels. In Section 5.3, we found that the kilometers saved for the base scenario with $Q_k = 2$ and $A_k = 10$ was 3.49 percent points. However, in the larger problem instance, we see that the kilometers saved up are 24.33% for Amsterdam. This is due to the fact that relatively small detours are performed for many deliveries. Whereas, when professional drivers are used the total sum of kilometers traveled for these deliveries are enormous. Moreover, we conclude that the crowdsource delivery platform becomes more environmentally friendly with the increase in parcels and customer dense areas.

We can conclude that the crowdsource delivery platform is both cost-efficient and beneficial in environmental terms. Based on the larger problem instance, roughly €0.25 is saved per parcel delivery. Besides, results have also shown that the use of private drivers can save up to 14 percent points on kilometers saved over the use of professional drivers. For larger problem instances this is even 24 percent points. However, the platform is highly dependent on the number of available
5.4 Improve computational time

Table 5.10: Results and CPU time for large problem instance.

<table>
<thead>
<tr>
<th>Area</th>
<th>Amsterdam</th>
<th>Rotterdam</th>
<th>The Hague</th>
<th>Utrecht</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drivers:tasks</td>
<td>341:341</td>
<td>159:159</td>
<td>213:213</td>
<td>212:212</td>
</tr>
<tr>
<td>Total costs savings (€)</td>
<td>87.06</td>
<td>48.39</td>
<td>50.31</td>
<td>46.26</td>
</tr>
<tr>
<td>Number of options</td>
<td>942.79</td>
<td>533.11</td>
<td>318.04</td>
<td>206.57</td>
</tr>
<tr>
<td>Parcels-matched (%)</td>
<td>81.52</td>
<td>100.00</td>
<td>75.59</td>
<td>71.70</td>
</tr>
<tr>
<td>Drivers-matched (%)</td>
<td>54.25</td>
<td>59.12</td>
<td>48.80</td>
<td>46.22</td>
</tr>
<tr>
<td>Number of additional stops</td>
<td>1.50</td>
<td>1.69</td>
<td>1.55</td>
<td>1.55</td>
</tr>
<tr>
<td>Detour (%)</td>
<td>27.77</td>
<td>28.98</td>
<td>35.21</td>
<td>22.78</td>
</tr>
<tr>
<td>Kilometers saved (%)</td>
<td>24.33</td>
<td>17.47</td>
<td>16.78</td>
<td>19.32</td>
</tr>
<tr>
<td>Recursive algorithm (s)</td>
<td>290.54</td>
<td>45.89</td>
<td>52.21</td>
<td>31.29</td>
</tr>
<tr>
<td>TSP-TWPC (s)</td>
<td>2237.95</td>
<td>527.95</td>
<td>437.81</td>
<td>282.32</td>
</tr>
<tr>
<td>Matching problem (s)</td>
<td>466.04</td>
<td>70.17</td>
<td>81.65</td>
<td>70.12</td>
</tr>
<tr>
<td>Total time (s)</td>
<td>2994.53</td>
<td>644.02</td>
<td>571.66</td>
<td>383.74</td>
</tr>
</tbody>
</table>

private drivers. This conclusion is endorsed by the results of Archetti et al. (2016); Arslan et al. (2016); Chen et al. (2016); Lee and Savelsbergh (2015). Therefore, particularly in the start-up phase, the usage of professional drivers is required. Not only to deliver unmatched parcels but also to maintain the service level. However, the number of available drivers becomes less important when we use different cost models. When a fixed price per delivery is used, either in combination with a compensation per minute or not, less private drivers are used compared to only a compensation per minute. This is mainly due to the utilization of the private drivers’ stop willingness.
6 Conclusion and recommendations

In this chapter, we will present the main findings of our research. In Section 6.1, conclusions are drawn, based on the research questions that were formulated in Section 1.2. In Section 6.2, recommendations for Trunkrs and for future research are given.

In this research, we consider a distribution network that is mostly operated by private drivers and has professional drivers as a backup option. In particular, the crowdsourced delivery platform takes advantage of the spare capacity in the private vehicles from commuters along their scheduled trips. We provide an arc-based general MILP formulation for the routing and matching problem. The model incorporates both the professional backup option and the private drivers. In addition, the private driver's maximum additional travel time, stop willingness, and the option to transfer parcels between drivers is considered. However, due to the high computational complexity we also provide a path-based heuristic to solve the routing and matching problem more efficiently. This heuristic can be divided into three separate parts that are subsequently solved. In the first part we find all feasible jobs for all drivers by a recursive algorithm (i); thereafter, the optimal route and associated costs savings (ii) are calculated for all jobs performed by a certain driver. Finally, the (iii) match between drivers and jobs that maximize costs savings is made. Results show that the crowdsourced delivery platform is more cost-efficient and environmentally friendly than the traditional delivery method.

6.1 Research questions

In this section, we start with answering the supportive research questions before we answer the main research question.

1. How many private drivers are necessary to create a reliable distribution network?

The reliability of the distribution network is highly dependent on the number of available private drivers. However, the quantity is not the only aspect that matters. Other aspects, such as the customer density, region size, and the number of parcels are all aspects that influence the required number of private drivers. Results show that the more drivers that are available, the more cost-efficient the crowdsourced delivery platform becomes. However, not all private drivers that are available are used. This does not imply that less private drivers also suffice. Instead, the more options the model has, the better the solutions are that can be generated by the routing and matching model. Therefore, there is no absolute answer to this research question. Especially, with the backup option that is available a reliable distribution network can always be made. However, this is not cost-efficient. Moreover, we can conclude that the cost model has an effect on the required number of private drivers. How the cost model is related to the performance of the
routing and matching problem is explained by the next question.

2. What cost model should Trunkrs use for the private drivers?

We see that the utilization of the private drivers is dependent on the cost model. If the private drivers receive a compensation per minute of their detour, the model maximizes the driver’s additional travel time. This is mainly due to those shipments that require more travel time are more beneficial since the costs savings are higher in these cases. However, in this case, more private drivers are necessary to deliver the parcels. However, if the private drivers receive only a fixed price per delivery, or in combination with a compensation per minute, we see that the driver’s stop willingness is maximized. In both cost models, we see a decrease of roughly 50 percent points in the Drivers-matched compared to the cost model of only a compensation per minute. Besides, the total costs are also less. Therefore, we can conclude that Trunkrs should aim for a cost model that maximizes the stop willingness of the private drivers, which is a cost model that includes at least a fixed price per parcel. This is cost-efficient, and the crowdsource delivery platform becomes less dependent on the number of available private drivers.

3. How should the optimal routes for package deliveries be determined?

In this research, we provide a heuristic to solve the routing and matching problem. Results show that this is a reliable heuristic. However, we did not compare it to an optimal solution that could be obtained by solving the arc-based MILP given in Section 3.2. The heuristic does not provide us with the optimal solution because the transportation request route from the origin to destination is split into three trips beforehand. These three trips are: the first-mile delivery, the line-haul, and the last mile delivery. All trips need to be executed subsequently by different drivers. The arc-based MILP does not examine the separated trips individually but considers the distribution network in its whole. Hence, the heuristic will not always provide the optimal solution.

The heuristic works in three-fold: first, the feasible jobs are determined for each private driver based on their constraints. Second, we solve the routing for each job for all private drivers by the TSP-TWPC. The TSP-TWPC provides us with the optimal solution and can easily be solved for small problem instances. The optimal route is based on the route that maximizes the costs savings. The size of a job depends on the stop willingness of the private drivers. In our experimental design, we chose for a maximum stop willingness of 3. Therefore, the problem instances are small and easily solved. However, results also show that most computational time is required during the determination of the optimal route. This is mainly due to the high Number of options per private driver. Each option of each private driver needs to be solved in order to continue the heuristic. This may become a problem when problem instances grow, and drivers are willing to make more stops. Therefore, it is not conceivable to consider a heuristic approach for the determination of
the routes instead of an exact solution. Finally, the matching problem is solved by a path-based MILP. This MILP proves to be reliable and relatively fast in finding the matches between drivers and parcels.

Our proposed solution method provides Trunkrs with a good solution for their routing and matching problem. As suggested above, a heuristic for the routing problem could improve computational times. There is much literature on certain heuristics that will not negatively affect the outcome significantly. Hence, we can conclude that the optimal routes and matches for package deliveries can be determined by our proposed solution method.

4. What is the trade-off for using professional drivers?

Several conclusions concerning the use of professional drivers could be made. First, the results show that the professional drivers are essential in the start-up phase of Trunkrs. Especially, since not enough private drivers are available to guarantee a high service level for their customers. Second, since most private drivers are only available during the “evening hours” professional drivers are indispensable for the first two stages in the distribution chain: the first-mile delivery and line-haul. Trunkrs needs to guarantee deliveries at the service stations to maintain the high service level and ensure the synchronization with private drivers.

During the last mile delivery, we can conclude that professional drivers are required to deliver unmatched parcels. From the results (see Chapter 5, Figure 5.1), we see that most unmatched parcels are clustered and/or located near a service station. Hence, transportation costs for the use of professional drivers are relatively low. However, Trunkrs might consider using professional drivers in customer dense areas since the costs difference between the use of private and professional drivers are minimal in this case.

“How should Trunkrs’ distribution network be operationalized to minimize costs of distributing packages and guarantee same-day delivery?”

Based on the answers to the supportive research question, we can conclude that Trunkrs should aim to utilize the use of private drivers fully. However, Trunkrs cannot fully disregard the use of professional drivers. Trunkrs remains dependent on their backup option, the professional drivers, to maintain the high service level and guarantee same-day delivery. Moreover, the stages in the distribution chain that lead to the last mile delivery are required to be executed by professional drivers. The transportation costs during these stages are relatively small, and professional drivers are better suited for these kind of operations.

Assuming that private drivers need to travel anyway, the results show that an increasing number of participating private drivers is beneficial for Trunkrs. Moreover, it is also more desirable from a societal point of view, due to the reduction in CO$_2$ emissions and traffic congestions. Hence, Trunkrs should establish a crowdsource delivery platform with many private drivers.
The solution method we provide will help Trunkrs to solve the routing and matching problem they encounter during their daily operations. It supports the decision-making processes and helps them match parcels with private drivers. Moreover, it will also supply them with the optimal route for each delivery route executed by private drivers. Regarding cost models, Trunkrs should consider paying private drivers at least with a fixed price per delivery. In this case, the total costs will be less and the heuristic aims to fully utilize the capacity of the private drivers. Moreover, we expect that private drivers are more willing to participate when they receive a fixed fee per parcel. Besides, private drivers might consider increasing their stop willingness, which would become more beneficial for Trunkrs. Since results show that the total costs decrease with the increase of the stop willingness.

6.2 Recommendations

The following section is in two-fold. First, we provide recommendations especially for Trunkrs in Section 6.2.1. We will explain how the results of our research should be interpreted, and used as a supportive tool for Trunkrs’ decision-making process. Finally, we will suggest directions for future research on this topic in Section 6.2.2.

6.2.1 Recommendations for Trunkrs

Thoroughly mentioned through the research, the cost-efficiency of the crowdsource delivery platform is highly dependent on the number of private driver participants. Hence, it is highly recommended to generate a network of private drivers evenly spread across the Randstad at first. Besides, to become more environmentally friendly is it conceivable to mainly focus on areas that are customer dense, such as the four big cities in the Randstad. Results show that for Amsterdam on an average week (simulated as a day in our experiment) up to 24% kilometers can be saved by the crowdsource delivery platform. However, this can only be obtained if the delivery platform has a sufficient amount of private drivers. Hence, the focus is on generating a huge network of private drivers during the start-up phase. This network of private drivers will mainly consist of commuters. However, to increase the number of participants, we advice to also look at groups that at first sight do not look interesting (Peine, Van Cooten, & Neven, 2016). For example elderly people, homemakers, or even other logistics service providers with spare capacity. These groups are available more frequently during the day, and possibly also willing to make more deliveries.

In addition, to fully utilize the stop willingness of the private drivers we propose to choose a cost model that has a per parcel incentive. Results clearly show that the transportation costs are less compared to a variable compensation per minute. A fixed fee per parcels might also lead to a higher stop willingness and the number of participants. The latter is caused by the fact that the
payout of a fixed price per parcel is easier to understand than a variable compensation per minute of their detour.

In this research, our experimental design was relatively small. We limited our results to only one client and four service stations. If more clients participate in the crowdsource delivery platform it is conceivable to pre-process the problem instances as proposed in Section 3.3.4. Also, results suggest it is desirable to analyze the characteristics of the distribution network before implementing the crowdsource delivery platform. For instance, the spatial distribution of the origin and destination of the private drivers and parcels affect the performance of the platform. Hence, Trunkrs might consider collaborating with more service stations or storage services (e.g., locker systems) at convenient locations. This could alleviate the computational pressure on the model, and time synchronization restrictions for parcels and drivers. Besides, more service stations also provide more matching options for the system.

6.2 Future research

In this section, we will elaborate on four suggestions for future research. First, future work can be done in finding a suitable heuristic for the route optimization. Since we are using an exact method most computational effort is made during this phase of solving the routing and matching problem. Several heuristics could be investigated and see how they affect the results compared to the exact method. Secondly, although we did provide an exact solution method for the routing and matching problem, we did not solve it. Therefore, we do not know how the results compare to this exact solution in both optimality and computational time improvement.

Thirdly, as already briefly mentioned in Section 5.2, a rolling horizon approach can be introduced. Currently, everyone is connected by mobile smartphones. Hence, private drivers can participate in the crowdsource delivery platform at any time. Therefore, both delivery tasks and driver assignments arrive dynamically throughout the day. By repeatedly solving an offline problem each time that a new task or driver arrives, we solve the routing and matching problem in a dynamic setting rather than a static setting. The impact on the delivery time, and how this affect the performance of the platform should be investigated.

Finally, in line with the “sharing economy” it is conceivable to investigate a collaboration with other logistics service providers. Particularly for small and medium logistics service providers, establishing coalitions with other firms can help strengthening their market position and extend their resource portfolios (Krajewska & Kopfer, 2006). Since the crowdsource delivery platform is highly dependent on a high service level during the first-mile delivery and line-haul, it is conceivable to collaborate with established logistics service providers. From their point of view, a collaboration with a distribution network operated by private drivers is interesting to investigate for the last mile delivery.
References


Van Cooten, C. (2016). Research Models as support for a new Pickup and Delivery Problem. Literature review, School of Industrial Engineering, Eindhoven University of Technology.