Problem Description

The purpose of this project is to consider the integration between a flexible and a fixed route public transportation system, and then the benefits and drawbacks by making it possible to transfer between the modes of transport. This is done by formulating a mathematical model of the Integrated Dial-a-Ride Problem which considers both the costs of operating the vehicle fleet and the quality of the service. The problem will be solved by the use of an exact method, and the focus will be on strengthening the formulation with help of valid inequalities.

Main contents:

1. Literature review of related routing problems
2. Description of the problem
3. Development of the mathematical model
4. Describe ways to strengthen the model and possible a more advanced exact solution approach
5. Implementation of the model using Xpress-mosel
6. Computational study and discussion of the option of transfers between services

Preface

This thesis is the result of my thesis work during spring 2015 at the Norwegian University of Science and Technology, Department of Industrial Economics and Technology Management.

The project presents a routing problem called the integrated dial-a-ride problem. The problem consists of a homogeneous vehicle fleet and a fixed route network. Given characteristics of the vehicle fleet and the network specifying transfer locations and links between the locations, a set of requests are to be scheduled. The requests specify a load (of people), pickup and drop off locations and time windows. A simplified model of the problem is proposed, implemented in Xpress IVE optimization suite using Xpress-Mosel programming language, and solved on several test instances. Furthermore, the effects of among others incorporating the option of transferring between vehicles and fixed network on the problem complexity the operational fleet costs and the user convenience is considered.

The project have been done in collaboration with my academic supervisor Henrik Andersson, who has also been in contact with Carl-Henrik Häll at Linköping University. I am grateful for all help and guidance they have provided throughout this semester, and the results provided in this thesis would not have been possible without their support.

Trondheim –.06.2015

Ida Nedregård

Abstract

The demand for public transport network is increasing, both to provide a more environmentally friendly travel option and as a service for those who do not have access to private ways of travel. In this thesis an integrated dial-a-ride problem (IDARP) is presented. A dial-a-ride problem (DARP) is similar to a pickup and delivery problem where the load to be picked up and delivered represents people. The integrated option of a DARP includes the possibility for the request to transfer between vehicles and travel some part of the route by another mode of transport. In addition, the IDAR formulation in this thesis allow for the requests to walk between locations. In detail, the problem has to fulfill a set of requests through scheduling a demand responsive vehicle fleet. In addition, some part of the request can be performed by a fixed route service consisting of one connected network that is assumed cost free. Furthermore, if the distance between two locations is less than the maximum distance a request states is acceptable to walk, the request can walk this distance if it is beneficial for the demand responsive vehicle fleet. The goal is to find a schedule that minimizes fleet operation costs and user inconvenience, while at the same time considers several constraints. The constraints can be grouped into restrictions set by the requests, restrictions for the vehicle fleet and for the fixed network. Examples of constraints can be time windows of service and capacities of load and usage of the vehicles.

In this thesis some previous studied solution methods have been discussed, and a thorough discussion of different aspects of the problem, in addition to modeling issues, is included. An arc-based IDAR formulation is given, and by using an exact solution method several test cases are solved. Several groups of valid inequalities and reduction techniques have been incorporated into the model so as to represent the convex hull better. These options are tested to find which options should be included in the model on a regular basis. Through the use of these cases the benefits and drawbacks of including three problem features are discussed. These problem features are (1) the possibility of transfer, (2) the possibility to transport multiple requests at the same time, and (3) the possibility to allow a request to walk some part of the trip. On average, it seems that the problem features lower the cost of operating the fleet, while not considerably increasing the user inconvenience. However, the problem size and complexity increases when including the problem features, which makes it more difficult to find good solutions. Furthermore, a few modeling issues and some characteristics of the vehicle fleet and the requests are discussed both from a technical perspective and a economical perspective. On average, if the setting and characteristics are set so economical benefits are possible, the resulting problem is usually more complicates to solve. The implementation of the IDARP manages to solve small instances with up to five transfer locations, one node visit and nine request, or two node visits and seven requests.

Sammendrag

Etterspørselen etter offentlige transportmidler og kollektivtransport øker, både for å tilrettelegge for mer miljøvennlig transportmidler og for de som ikke har et annet transportmiddel tilgjengelig. Denne masteroppgaven løser et ruteproblemet hvor en fleksibel transportflåte med biler og en mindre fleksibel kollektivtransportflåte er integrert for å skape et helhetlig transportmiddel, referert til som *integrated dial-a-ride problem* (IDARP). I dette ruteproblemet skal folk transporteres fra et hentepunkt til et sluttpunkt ved hjelp av flåtene tilgjengelige. De tilgjengelige flåtene består av en bil flåte, hvor ruteplanen bestemmes helt fleksibelt så lenge noen spesifikke kvalifikasjoner holdes. I tillegg finnes det et kollektivnett hvor flåten har spesifikke ruter de må følge men hvor det er antatt at frekvensen på avganger er så høy at dersom en passasjer er satt av på stedet kan vedkommende reise med kollektivflåten med en gang. Passasjerene kan derfor bli transportert med den fleksible flåten enten hele veien eller de kan benytte seg av kollektivnettet hele eller deler av veien. I tillegg tillater modellen at folk kan gå mellom steder dersom avstanden er kort nok. Målet med problemet er å finne de billigste rutene for den fleksible flåten, med hensyn til antall biler, total avstand reist og tidsbruken av bilene. Kostnader for å holde kollektivnettet operativt og for å benytte kollektivnettet er ikke inkludert i modellen. For å forsikre seg om at kvaliteten på tjenesten ikke faller for mye er det lagt ved bøter dersom folk må gå, bytte mellom biler eller kollektivnett, samt dersom total reisetiden øker utover direkte reisetid (bil kjører direkte fra hentested til leveringssted). Det er også lagt ved andre restriksjoner i problemet, for eksempel at passasjerene skal hentes innen et visst tidsvindu, eller at biler kun kan ha et visst antall passangerer i bilen samtidig eller kan kun brukes i en viss tid sammenhengende.

Denne masteren går gjennom et utvalg av ulike studier gjort på liknende ruteproblemer. Først er eksakte og heuristiske løsningsmetoder diskutert, deretter ulike problemkarakteristikker, og til slutt noen tema rundt modellering og implementering av problemet. En buebasert IDAR formulering er gitt og problemet er løst ved å bruke en optimal løsningsmetode. Flere grupper med restriksjoner er lagt ved basis modellen for å minke kompleksiteten av å løse IP problemet, og for å finne et løsningrom som representere det konvekse hullet bedre. De forskjellige gruppene er testet i en innledende analyse og en diskusjon rundt effekten på løsningskompleksiteten er gjennomført. Resten av analysen er fokusert rundt en teknisk og en økonomisk diskusjon av problemet når ulike problem karakteristikker og faktorer er endret. De tre problem karakteristikkene omhandler muligheten til (1) å bytte mellom biler eller å bruke kollektivnettet, (2) transportere flere passangerordre samtidig i samme bil, og (3) la folk gå mellom ulike steder. I tillegg er enkelte faktorer som omhandler bilflåten og ordrene endret, og diskutert ut fra en tekniske og økonomiske standpunkt. Enkelte faktorer som omhandler modellering og implementering av problemet er også sett på i analysen. Generelt, å inkludere problem karakteristikker og faktorer i problemformuleringen som skaper økt fleksibilitet for ruteproblemet, og gjør det mulig å finne billigere ruter, kompliserer problemet og gjør det tyngre å løse til optimalt. Implementeringen av dette problemet klarer å løse små instanser med fem kollektivsteder, en node for hvert sted og ni ordre, eller fem kollektivsteder, to noder per sted og syv ordre.

Contents

List of Figures

List of Tables

1 Introduction

Today, people travel more frequent and farther than earlier. The demand for personal mobility is increasing, thus increasing the pressure on the traffic system. In addition, consumer demand is higher, and many passengers require better convenience, for instance door-to-door transport. As a result, more people prefer to travel by private cars. The use of private cars increases congestion and the carbon dioxide emissions. Another factor to consider is that not everyone has the option of driving his or her own private car. Especially, two groups of people to consider are the elderly and disabled. They might not be able to travel by car or to take themselves to the fixed route transport services. As a consequence, it is necessary to provide services that are convenient enough to transport elderly and disabled, and convenient enough to provide an alternative for private cars. These services range from completely fixed route services to entirely demand responsive services (Errico et al., 2013). Fixed route services are services which travels on specific and regular routes at scheduled times, and are therefore cheaper to operate. The passengers have to get themselves to and from the departure and delivery points of the route. On the other side, entirely demand responsive services are services that are not limited to certain routes and can pickup and drop off passengers at the time and place of their convenience. As a consequence of this flexibility the system is expensive to operate. Several integrated systems also exist, where fixed routes are connected with more demand-responsive or semi-fixed route systems. Integrating an expensive demand responsive transportation system with a cheap and fixed route service might make it possible to lower the cost of the service while keeping the service quality at an acceptable level. As a consequence the service can be a medium cheap and semi-flexible transportation alternative and targeting a bigger part of population.

To answer the demand of increased mobility, public transport systems are evolving towards bigger and more flexible services. These services can provide a more environmentally friendly alternative to private cars and taxis while maintaining user convenience. For those groups that do not have access to private cars or are able to use fixed route services, more demand responsive services are essential. At present, these systems are quite expensive to operate and thus highly subsidized, and are only available for people with special permits (Errico et al., 2013). However, by managing to lower the cost of operating the flexible traffic system the service can be made available for a greater part of the population. In recent years, several so-called transportation network companies (TNC), for instance Über and Lyft in the US, have emerged (Transportation License Section). These companies use an online-enabled platform to connect passengers with drivers using their personal, non-commercial vehicles. Through this mechanism they have been able to supply a service that is similar to taxies, but a lot cheaper. In addition, several ride-sharing systems have emerged, connecting those who have empty seats in their car with people who need a ride.

In operation research many different forms of routing problems are discussed, and among these are the pickup and delivery problems which transports goods to

or from one (or several) depot(s) or transport goods between customers (Golden et al., 2008). The problem considering scheduling a vehicle fleet to transport a number of passengers from their pickup locations to their drop off location at specific times is referred to as a dial-a-ride problem (DARP). The simplest form of a DARP, taxis, where a vehicle from the fleet pickup the requested load, drives directly to the requests drop off location and afterwards answers the next request in line. However, the cost of operating this fleet is high since the requests are serviced on a first-come first-served strategy without regard to geographic locations of the requests. Furthermore, the time to perform the set of requests increases if the service of one request has to be finished before the next request can start. In addition, considerable time is spent driving an empty vehicle as the vehicle have to drive empty from drop off of one request to pickup of the next request.

This thesis considers a type of DARP, referred to as the integrated dial-a-ride problem (IDARP), which look at the possibility to integrate the service provided by the vehicle fleet with an already existing fixed route service. Furthermore, the formulation presented also allow for performing several requests at the same time. This thesis considers the effects when integrating the service provided by the vehicles with an already existing fixed route service. Both consequences of the cost of the fleet and for the passengers, in addition to technical aspects of the model are considered. The cost of providing the service and the user satisfaction is important elements when considering the balance between demand and supply of the service. The hope is that if an integrated dial-a-ride problem (IDARP) can be solved efficiently, it can be possible to provide a convenient and flexible, in addition to low-cost, mode of transport for a bigger part of the population. Furthermore, if a demand responsive traffic system can be provided for a lower cost, then the government can save on subsidies. An additional benefit is that this might reduce congestion in bigger cities if the advantages for the users of using the public systems are greater than using private vehicles. As a consequence this thesis provides a formulation of the IDARP that considers both costs of operating the fleet and costs associated with user inconvenience. The technical aspects are included because a key element when developing a successful traffic system is developing an efficient decision support system. An efficient formulation and solution technique is important to be able to schedule and organize the vehicle fleet and to provide a reliable service within the desired time window. For instance, the decision support system has to be able to find an acceptable schedule in the limited time window before the service has to be performed. For this reason several technical aspects of the formulation are considered. These three factors, cost of operating the fleet, cost of user inconvenience and technical aspects, can be evaluated so as to examine if it is possible to provide an efficient and user convenient transport service with the help of solving the IDARP.

First, in Section 2 a literature review of some relevant routing problems is considered. Next, in Section 3 the background of the problem and the problem description for this thesis is provided. Then, the important modeling issues are discussed in detail. Next, the model formulation is presented and some implementation issues are discussed. In Section 4 the importance of writing a strong model

and different valid inequalities are discussed. Then, an example is given in Section 5, before the effect of the different reduction techniques are discussed during the process of finding a default option (Section 6.1). In the computational study, a technical analysis is discussed in Section 6.2, while the economical effects are discussed in Section 6.3. Lastly, a short conclusion and some possible extensions and future research are considered (Section 7 and Section 8).

2 Literature Review

The integrated dial-a-ride problem is one of many different routing problems studied in the literature, and the different routing problem have several characteristics in common. Therefore, reviewing existing literature has increased the overall understanding of the routing and scheduling problems. This especially includes the different characteristics of the problems, their modeling techniques and their solution approaches. Thus, this section provides a review of some studies done on routing problems. The focus of the review is on solution approaches, while characteristics and modeling techniques are discussed further in Section 3. To start, the traveling salesman problem (TSP) is considered. The goal of the TSP is to minimize total costs while visiting a finite number of places. The TSP is one of the most widely studied combinatorial optimization problems, and it is said to be an ideal starting point when developing techniques to other problems (Applegate et al., 2006). In fact, the TSP often surface as a subproblem of bigger and more complex routing problems (Christiansen, 1996). After the TSP the review considers the vehicle routing problem (VRP). The VRP is a generic name for a set of routing problems where a vehicle fleet is scheduled so as to satisfy demand in an area. Each of the vehicles has to satisfy a traveling salesman problem for the customers they are supposed to visit. The studies discussed in these sections centers around exact solution methods and mention useful aspect to be used in other generalizations of the routing problems. A generalization of the VRP is the pickup and delivery problem (PDP), which is considered next. In the PDP the routes have to satisfy transportation requests consisting of both pickup and drop off locations. Next, a generalization of the PDP, namely the dial-a-ride problem (DARP), where the requests specified trips to transport passengers from their initial pickup problem to their drop off destination, is considered. Additionally, two variants from the PDP/DARP is considered. First, the pickup and delivery problem with transshipment (PDPT), where the load can transfer between vehicles at specific transfer locations, are considered. The second variant considers pickup and delivery problems where two or more modes of transports are integrated into a combination of fleets. For instance, the model studied in this thesis, which is one version of the integrated dial-a-ride problem (IDARP), includes transfers between homogeneous vehicles and a fixed route service.

The following review starts by defining the problem and some variants of the problem, and then focus on exact and heuristic solution approaches for the problem. The exact solution approaches is centered around branching methods which are a quite successful approach for IP problems. The branching method is usually combined with either cutting planes or column generation or both. Since use of valid inequalities to strengthen the formulation is an effective way to reduce the gap between the LP and IP solution, and since this is the focus of this thesis, studies considering valid inequalities are usually included in the review. Since the routing problem is NP-hard heuristic are usually necessary to solve medium to large size instances. Therefore, a big part of this review focuses on heuristics, and especially metaheuristics, as a solution method. The studies can focus on one heuristic, create a heuristic from different heuristics or compare different heuristic approaches.

Furthermore, the heuristics can be used as a part of the exact solution approach, for example a heuristic can be used to find a column with reduced cost, instead of using an exact method to solve the subproblem optimally, and thus sometimes enhance performance. The review starts with the traveling salesman problem and ends with the problem variant that is considered in this thesis, the integrated routing problem.

2.1 The Traveling Salesman Problem

In a traveling salesman problem (TSP) the goal is to minimize total costs while visiting a number of places exactly once. The most common practical interpretation of the problem is a salesman who wants to find the shortest path through n clients. Another application of the TSP is job sequencing, where the goal is to minimize the changeover cost of performing a set of jobs which must be performed sequentially on a single machine (Laporte, 1992). The traveling salesman problem is NP-complete and, though well studied, is still one of the most challenging problems in Operational Research (Laporte, 1992). In fact, 40 years of studies has not led to a better bound on the running time than the bound, $O(n^2 2^n)$, developed by Held and Karp in 1962 for their TSP algorithm. The most well-known and studied exact solution method for solving the TSP is the branch-and-bound method or its derivatives. These branching methods use an organized and exhaustive search for finding the best solution in each branch. A number of heuristics have also been studied, and especially local search algorithms, simulated annealing, neural network algorithms and genetic algorithms (Applegate et al., 2006). The reader is referred to the book "The Traveling Salesman Problem" by Applegate et al. (2006) for a more thorough review. Below a small historic review of exact solution methods for the TSP is provided. First, a brief selection of Laporte (1992) throughout review until the early 1990s are given, before some newer findings are discussed. The most important findings and milestones for the TSP seems to be from earlier periods, and as a consequence the focus on this review is on earlier studies. The review is centered around branching methods as this approach have proven quite successful and is well represented in generalizations of the TSP which are discussed in later sections.

Laporte (1992) described the study by Carpaneto and Toth (1980) of the branch-and-bound solution approach for solving the traveling salesman problem. First, they solved a modified assignment problem at the root node. Next, if subtours are found during the solution procedure, subproblems prohibiting the arcs of the subtours are created to break these in the next stage of the tree. This procedure is repeated until the optimal feasible solution is found. Later, Balas and Christofides (1981) used a more complex and powerful solution method that result in a smaller search tree than Carpaneto and Toth's (1980) approach. They used a Lagrangean method that considered all constraints in the objective function and then branched on the resulting solution. Furthermore, Baker (1983) presented a branch-and-bound algorithm for the TSP with time windows. To find bounds on the subproblems created in Baker's branch-and-bound tree, acyclic networks were

created to represent the subproblems and were solved by a longest path method. Then, in the late 1980s, Miller and Pekny (1989) studied a parallel branch-and-bound algorithm based on the same relaxation as Baker. They managed to solve randomly generated asymmetrical TSP instances with up to 3000 nodes to optimality.

In the second half of 1990s, Applegate et al. (1995) studied the traveling salesman problem and proposed new ways of finding cuts, new ways of handling the LP relaxations, new ways to select an edge to branch on and a new way to prune on the resulting tree. They further described a new way to find new cuts by using a cutpool, and a way of gluing old constraints together into a conglomerate cut. By using these methods they managed to solve 20 previous unsolved problems from the TSP. The problem ranged from 1000 to 7397 cities. Later, Applegate et al. (1998) continued their work on the TSP formulation and solution approach. During this work they added new cuts to strengthen the formulation. They also used different exact and heuristic separation algorithms to solve the problem. With this formulation and solution algorithm they solved problem instances with about 4500 cities in less than two days.

Bigras et al. (2008) proposed two integer programming formulations for a time dependent TSP. The time dependent traveling salesman problem is a TSP where the transition cost between nodes depend on the period the node is visited, assuming that one period is needed to travel between nodes. Adding cutting planes strengthens the formulation. Subtour elimination constraints and 2-matching cuts are added at each node, while the clique inequalities are generated at the root node of the problem. The problem is solved using a branch-and-price method, branching directly on the variables. A four-phase column generation procedure is included in the solution approach. The initial problem is decomposed into a master problem and pricing problems using Dantzig-Wolfe decomposition. This pricing problem is a shortest path problem on a multipartite network with complicating constraints forcing the path to visit every node exactly once. The shortest path problems are solved using a dynamic programming algorithm, and the solution is added to the master problem as a new column. The model is tested on instances up to 45 requests. Quite recently, Steinerberger (2015) has studied the bounds on the universal constant for the TSP, which was established to be between $0.625 \leq \beta \leq 0.922$ by Beardwood, Halton and Hammersley in late 1950s. He managed to slightly improve both upper and lower bounds and found that $0.629 \leq \beta \leq 0.906$ holds for the TSP constant.

To sum up, the traveling salesman problem, which is the problem of finding the shortest path between n nodes, is a NP-complete problem and though it is a well studied problem is still one of the most challenging problems in Operational Research (Laporte, 1992). The greatest milestones for this problem centers on studies done in the second half of 1900rd, while some of the greater findings in the recent years are due to stronger and more efficient computers. Next, a review of some of the generalizations of the TSP is provided.

2.2 The Vehicle Routing Problem

The vehicle routing problem (VRP) is a routing problem where the vehicle fleet, centered at a central depot, is required to visit and fulfill some requirements at a set of locations. The VRP is a NP-hard problem and several variants of the problem have been studied in the literature. The common variants of the vehicle routing problem is (1) the capacitated vehicle routing problem (CVRP) where the vehicles have a capacity limits, (2) the vehicle routing problem with time windows (VRPTW), where the customers have to be visited within a certain time window, and (3) the vehicle routing problem with pickup and delivery, which is usually referred to as the pickup and delivery problem, and is discussed in the next section. The problem variants can either consider a homogeneous vehicle fleet (all vehicle characteristics are the same) or a heterogeneous vehicle fleet (vehicle characteristics are individual for each vehicle). Additionally, the problems can consider a limited fleet size or a flexible fleet size. Furthermore, the demands can be deterministic (certain and known) or stochastic (uncertain). The problem can also be characterized as static (all requests known before scheduling) or dynamic (requests are known in real time). In addition, variable routing costs are usually considered, and can be dependent on the vehicle, dependent on the site or independent. In addition, some problems consider fixed costs associated with the vehicle fleet. The reader is referred to the survey by Cordeau et al. (2002a) for a more detailed review of VRP variants and with the different mathematical formulation.

Several exact methods have been studied to solve the different routing problems. The research have in particular centered on column generation methods and branch-and-bound methods (Golden et al., 2008). In the short review below three studies done on the VRP considering branch-and-price algorithms are discussed because of the efficiency and quality of the solution approaches. The last contribution to this part of the review considers stochastic demands instead of the usually deterministic demand. Heuristics have played a major part in solving real size instances of the vehicle routing problem. Three classical heuristics mentioned in the review by Cordeau et al. (2002b) are the Clarke and Wright algorithm, the sweep heuristic and the Fisher and Jaikumar algorithm (cluster first, route second algorithm). The two first approaches are quite intuitive, but because of their greedy nature makes it difficult to incorporate new constraints in the problem. The results from the last procedure Fisher and Jaikumar algorithm, a cluster first, route second approach, showed weak convergence and little flexibility. However, these algorithms are often important elements in more advanced heuristics and metaheuristics and are therefore worth mentioning. Recently, metaheuristics have been more popular as a solution technique, and have greater efficiency and flexibility. However, the metaheuristics are less intuitive and simple to understand. In particular, tabu search stands out as one of the more used metaheuristic for the vehicle routing problem (Cordeau et al., 2002b), while other popular metaheuristics are simulated annealing, variable neighborhood search and greedy randomized adaptive search procedure (GRASP), in addition to ant colony algorithms and genetic algorithms (Golden et al., 2008). The short review below considers three studies done on local search metaheuristics that mostly uses construction based heuristic to find a starting point for the search. For a more throughout review the reader is

referred to the book "The Vehicle Routing Problem" by Golden et al. (2008) for a detailed discussion of the different algorithms and studies to date. In addition, the survey by Cordeau et al. (2002a) also discuss different ways to find upper and lower bounds for different variants of the VRP and might be of interest for some readers. Furthermore, the review by Cordeau et al. (2002b) considers classical heuristics and metaheuristics as solution approaches for the VRP.

Chabrier (2006) proposed a model for the vehicle routing problem with time windows which minimize total distance traveled by the vehicles. A branch-and-price method is used to solve the problem. First, the problem is decomposed using Dantzig-Wolfe decomposition that splits the problem to a restricted master problem and pricing problem. The pricing problems are solved using an elementary shortest path algorithm. The branching is done when two fractional routes share the same arc (i, j). One route is allowed to use the arc, while the other is not allowed to use the arc. A lower bound is found solving the relaxed master problem for the integer problem on the current branch. By using the elementary shortest path subproblem and cutting planes, they obtained good lower bounds and pruning of the search tree, and they managed to find exact and improved solutions to 17 of 22 open instances with up to 100 nodes of the Solomon benchmark. Furthermore, Choi and Tcha (2007) proposed a vehicle routing problem with a heterogeneous vehicle fleet. Each vehicle has its own capacity, time and cost between nodes, and the goal is to minimize total routing costs. The problem is solved using a branch-and-price algorithm. They use Dantzig-Wolfe decomposition to decompose the problem into a master problem and subproblems. The master problem is formulated as a set-covering problem, where the covering constraint considers if a vehicle type visits a customer on the route. The subproblems are formulated as shortest path problems with resource constraints, and the solution created a feasible route. The routes represented columns for the master problem and the master problem is re-optimized with the new routes. If the solution found when solving the problem is not an integer solution, branching is done and the procedure is restarted. The problem is tested on three sets of six benchmark instances up to 100 nodes which are generated from the 12 benchmark instances by Golden et al. They showed that the solution approach outperformed existing algorithms both in solution quality and computation time.

While most of the studies on the vehicle routing problem have considered deterministic cases, some studies focus on stochastic problems. One reason for the lack of studies on stochastic problems is that the stochastic parameter adds a new level of complexity to the already NP-hard problem, while the deterministic case is easier to comprehend and draw conclusions from the result. Christiansen and Lysgaard (2006) proposed a model for the capacitated vehicle routing problem with stochastic demands. The demands are assumed to be following a Poisson distribution, and the goal is to minimize the total expected distribution cost. The problem is solved using a bound-a-price approach and uses a column generation method. The master problem is formulated as a set-partitioning problem, while the subproblems are solved using a dynamic programming scheme that solves a shortest path problem on an acyclic network. The model is tested on instances ranging

from 32 to 60 requests.

Bräysy and Gendreau (2002) proposed a model for the capacitated vehicle routing problem with time windows. The goal of the problem was to find the least cost routes while satisfying demand. As a solution approach they used a tabu search algorithm. This is a local search metaheuristic which examine a neighborhood of a solution for finding a better solution and moving to this solution. To avoid cycling attributes are temporarily declared forbidden and stored in a tabu list. The attribute will stay in the tabu list for a specific time (tabu tenure). The solution approach is tested on Solomon's 56 benchmark test problems. The instances have 100 requests, one central depot, capacity constraints, route time constraints and restricted time windows. They found that the tabu algorithms from Gehring and Homberger (2001), Cordeau et al. (2001), Taillard et al. (1997) and Chiang and Russell (1997) showed the best performance.

Bräysy (2003) studied the vehicle routing problem with time windows and introduced a deterministic metaheuristic based on a modification of the variable neighborhood search algorithm by Mladenovic and Hansen (1997). The procedure works as a four-phase approach. First, a route construction heuristic that considers different combinations of parameter values are used to create a set of initial solutions. Secondly, a route elimination procedure, an ejection chain based approach, is used to reduce and improve the solutions considering the number of vehicles. In the third phase the solutions are improved considering total traveled distance using four local search procedures, and among them a variable neighborhood descent algorithm. Finally, in the fourth phase, the best solution is improved by modifying the objective function to avoid local minimum. They got successful results on instances with 100, 200 and 400 requests provided by Solomon (1987), Gehring and Homberger (1999) and two real life problems by Russell (1995).

Pisiger and Ropke (2007) considered a general heuristic which can solve five different variants of the vehicle routing problem. They presented models for the five routing problems, and showed how to transform them into a rich pickup and delivery model and used an adoptive large neighborhood search developed by Pisinger and Ropke (2004) to solve the resulting problem. This approach builds on the framework proposed by Shaw (1998) with an adaptive layer. In the algorithm, the fix operation (destroy) selected a number of variables that are fixed at their current value. Then the optimize operation (repair) is initiated considering only none fixed variables. Afterwards all variables are released. The algorithm used a local search framework at the master level together with a number of fast construction heuristics and a number of destroy heuristics supporting these construction heuristics. The approach was competitive and manages to improve 183 best known solutions of 486 benchmark tests. Both smaller instances and bigger instances up to 400 to 600 or more customers is tested. The Pisinger and Ropke also test the solution approach on the vehicle routing problems with backhauls and the pickup and delivery problem.

To conclude, branching method and especially those which incorporate column

generation algorithms have proven successful for the VRP. To solve real sized instances of the problem heuristic approaches are common. Most algorithms usually start by creating an initial solution using a greedy construction heuristic, before searching systematically for a better solution. The most successful approaches are usually metaheuristics and include some sort of element to avoid getting caught in local optimums. As an example, tabu search algorithms usually perform well for NP-hard problems.

2.3 The Pickup And Delivery Problem

As already mentioned a generalization of the vehicle routing problem is the pickup and delivery problem (PDP). This is a vehicle routing problem where vehicles pickup certain loads at a set of pickup locations and deliver the loads at their respective drop off locations. The goal is to schedule the minimum costs routes for the vehicles. The pickup and delivery problem can also be divided into the same groups as the vehicle routing problems. In addition, the requests can specify certain requirements as load and time windows. Both exact methods and heuristics can be used to solve the routing problems. Small to medium sized instances can be solved by using an exact method but since the routing problems are NP-hard heuristics are needed to solve large instances. Studies considering exact solution methods for the pickup and delivery problem centers around branching methods or column generation approaches (Parragh et al., 2008b). Furthermore, the most common metaheuristics mentioned in the survey by Parragh et al. (2008b) where tabu search and genetic algorithms and their variants. In addition, studies considering other neighborhood searches approaches as for instance (1) large neighborhood search, (2) greedy randomized adaptive search and (3) variable neighborhood search (Parragh et al., 2008b) provided good results. Next, a short review of the solution approaches for routing problems are given, and the readers are referred to the surveys Parragh et al. (2008a) and Parragh et al. (2008b) and the survey Berbeglina et al. (2007) about pickup and delivery problems. Furthermore, the readers are referred to the book by Golden et al. (2008) about the vehicle routing problem for a thorough review of the routing problems and its variants. The exact solution approaches centers around branching methods in combination with cutting planes and column generation. For medium to large size instances heuristics are often the preferred solution approach, and most of the part of the review about PDPs is centered on some form of advanced heuristics or metaheuristics. In fact, even the most recent exact solution approaches use some form of heuristic approach incorporated into the solution technique.

Two main approaches for solving the pickup and delivery problem are branch-and-price and branch-and-cut. The branch-and-price method uses a branch-and-bound algorithm where the lower bound is calculated using a column generation algorithm. In the branch-and-cut method, valid inequalities (cuts) are added to the formulation at each node at the branch-and-bound tree to strengthen the relaxations that are usually solved by the simplex algorithm. Cordeau and Iori M. (2010) proposed a branch-and-cut algorithm for the single vehicle pickup and de-

livery problem with time windows (PDPTW) with last-in-first-out (LIFO) loading. Cordeau and Iori M. (2010) proposed three formulations for the problem, with the goal of minimize the total routing cost. Furthermore, they introduced several existing inequalities for the pickup and delivery problem and some new inequalities for this specific formulation. During the preprocessing part of the branch-and-cut procedure a cut pool is made out of the valid inequalities. The procedure uses both an exact separation approach and a heuristic separation approach based on a tabu search heuristic. Instances up to 17 requests was solved in less than ten minutes, while instances up to 25 request were solved within reasonable computing time. Similarly, Ropke and Cordeau (2008) considered the PDPTW. They study a branch-and-cut-and-price algorithm for solving the problem and uses a column generation approach to find a lower bound on the solution. Two pricing subproblems are used, one elementary and one non-elementary shortest path problem. Furthermore, they dynamically add valid inequalities to strengthen the relaxation. In addition to the exact method to solve the pricing problem, heuristics are used to solve the pricing problem. These heuristics were construction algorithms, large neighborhood search, and truncated label setting algorithms. The solution approach is tested on several instances from Li and Lim (2001) with instances of 100 requests and 500 requests. At the time of this research the results from the algorithm outperformed the recent branch-and-cut algorithms.

Hernándes-Pérez and Salazar-González (2004a) proposed a 0-1 integer linear model for the single vehicle PDP with the goal of minimize total travel costs. The problem is solved using a branch-and-cut algorithm. The lower bounds are computed by solving the linear problem relaxation of the problem, and adding valid inequalities in a cutting plane method tightens the bounds. Furthermore, through a separation problem, Benders' cuts are added dynamically to the model. The branching is done on the variables and the variable with value closest to 0.5 is chosen to branch on. To speed up the branching method, providing lower bounds and good feasible solutions, they use simple tour construction and improvement heuristics. The construction algorithms include elements of nearest insertion, farthest insertion and cheapest insertion. The model was tested on instances between 20 to 75 nodes and with different demand quantities. By 2007, Hernándes-Pérez and Salazar-González (2007) had improved their work on the symmetric single vehicle pickup and delivery problem. The goal of the model was still to minimize the total costs. Here as well, they proposed a branch-and-cut algorithm containing several valid inequalities. The problem is tested on three different classes of instances. One class consists of random generator instances from Mosheiov for the single vehicle PDP instances. The second class consists of a group of capacitated vehicle routing problem instances, transformed to the single vehicle PDP. The last consist of instances from Gendreau, Laporte and Vigo to the single vehicle PDP. The results showed that the problem could solve instances up to 50 customers without the added inequalities, and up to 100 customers with added inequalities. Hernándes-Pérez and Salazar-González (2004b) have also proposed heuristic approaches for the single vehicle PDP. They first proposed a heuristic approach based on a greedy algorithm that is improved with a k-optimality criterion. The heuristic uses a modified version of the TSP nearest neighbor algorithm. The heuristics use 2-

opt and 3-opt edge exchanges to minimize the infeasibility of the tour under some constraints. The heuristic iterates between a greedy algorithm and a tour improvement procedure. The second approach was an incomplete optimization algorithm. The algorithm was based on their earlier work on the branch-and-cut procedure for finding an optimal local solution, which made it possible to determine optimal solutions in restricted feasible regions. The problem is tested on ten random generated instances up to 500 requests.

Furthermore, construction-improvement heuristics are studied by Renaud et al. (2000) (Parragh et al., 2008b). They used a double insertion construction heuristic improved by deletion and reinsertion as introduced by Renaud et al. (1996). Later, Renaud et al. (2002) presented seven different perturbation heuristics to solve the static and single vehicle pickup and delivery problem. The paper written by Renaud et al. (2002) described three types of perturbation heuristics, resulting in a total of seven perturbation heuristics. The perturbation heuristics escaped local optimum during the solving of the pickup and delivery problem. The first perturbation heuristic is called instance perturbation (IP). This heuristic marginally perturbed the instance data when the local optimum is reached, and an improvement algorithm is applied to the modified data. Then the local optimum of the perturbed instance is translated back to the original data. This process can be applied iteratively. The second perturbation heuristic is an algorithmic perturbation (AP) that can be applied to both construction and improvement heuristics. In the first type, the criterion used to generate feasible solutions can be modified between iterations. While in the second type, an improvement algorithm can be used to iteratively move from one solution to another in its neighborhood. To avoid getting caught in local optimums the rule that defined the neighborhood can be altered slightly. Lastly, the third type of perturbation heuristics is the solution perturbation where the local optimum is modified and an improvement procedure is reapplied to the perturbed solution. The seven heuristics is tested on two classes of instances with up to 500 nodes. Their study provided a classification and performance analysis of several perturbation schemes to a wide range of combinatorial optimization problems. They found that applying perturbation could be quite powerful for the combinatorial optimization problems. The results showed improved the previous results of Renaud et al. (2000) by approximately four percent on the class 1 instances.

Gendreau et al. (2006) proposed a dynamic multi vehicle PDPTW formulation which has a goal of minimizing the weighting sum of total costs, sum of lateness over pickup and drop off locations and sum of overtime over all vehicles. The problem is solved by a tabu search algorithm using an ejection chain neighborhood (Glover, 1996). This neighborhood first removes one request from its route and insert it into another route, forcing the removal (ejection) of a request on this route to a third route and so on. The insertion of the pickup and drop off location in the request is done in sequence. First the best position for the pickup location is found and inserted and then the best position for the drop off location is inserted. In addition, an adaptive memory and a decomposition procedure are included in scheme to diversify and intensify the search. A pool of routes taken

from the best solution is used to restart the search in an unexplored region of the search space. The problem is then decomposed to focus the search on smaller subproblems. The problem is tested on instances of 20 vehicles and 24 requests per hour, and for ten vehicles and $24 - 33$ requests per hour. Gribkovskaia et al. (2007) proposed a number of heuristics yielding general solutions for the single vehicle pickup and delivery problem with combined demands. Solution heuristics include construction and improvement procedures and tabu search heuristic. The construction procedure first uses a (1) nearest neighbor procedure with backwards and forward merges, (2) sweep procedure with backwards and forwards mergers or a (3) modified sweep procedure with backward and forward mergers to obtain a possible infeasible Hamiltonian solution. Then, the algorithm relinks nodes to try to find improvements to the solutions. Next, a merging procedure eliminates excess visits. Six different construction procedures and an improvement procedure are used. The tabu search method used in the study is based on the unified tabu search algorithm introduced by Cordeau et al. (2001) for the vehicle routing problem. Gribkovskaia et al. (2007) tested the solution approach on 34 test instances of two types derived from instances to the capacitated vehicle routing problem, and contained 16 to 101 nodes. They found that the best known solutions generated by the heuristics are frequently non-Hamiltonian and may consist of visiting a customer twice.

Alshamrani et al. (2007) developed a periodic single vehicle pickup and delivery problem, where the customers who were not picked up in the current period can be picked up in the next period by adding a penalty cost to the objective value. The route design and pickup strategies are developed simultaneously, where stop volumes are known only probabilistically over a planning horizon. They develop a heuristic procedure for creating the route design-pickup strategy planning problem. As a solution approach they used a heuristic that first constructed a feasible route considering travel cost and then improves this solution considering the penalty costs. Thus they use an or-opt procedure to develop the drop off route while simultaneously determining the best pickup strategy over the planning horizon. They use a composite algorithm that incorporates heuristic rules and strategies to make the algorithm computationally viable while keeping the solution quality acceptable. They solved 900 problems with up to 30 nodes. Furthermore, a two stage heuristic for the multiple vehicle pickup and delivery problem with time windows was studied by Bent and Hentenryck (2006). The first part of the heuristic consists of a simulating annealing algorithm minimizing the number of routes. Afterwards, a large neighborhood search (LNS) algorithm is performed. The solution approach solved up to 600 customers within 90 minutes.

Pankratz (2005) proposed a grouping genetic algorithm for solving the pickup and delivery problem with hard time windows. Each gene is set to represent a group of requests. First a cluster of customers is selected at random. Then, eliminate the cluster from the chromosome and remove the associated route from the phenotype. Lastly, all removed requests are re-inserted into the individual by an insertion heuristic, and allocating a new vehicle if necessary to maintain feasibility. The problem is tested on instances with 50 requests, and the result showed that the

algorithm was competitive with other robust approaches. They also mention that at the time of this article, Pankratz (2005) believed that Jung and Haghani (2000) have reported the only genetic algorithm for solving the multi-vehicle PDPTW. They considered soft time windows, where violations of the time windows lead to penalty costs, but still provide a feasible solution. The problem is tested on instances ranging from five to 30 requests.

Hosny and Munford (2010) presented a single vehicle PDPTW. Furthermore, the travel time between nodes are assumed symmetric, that is that the time it takes to travel from i to j is equal to the time traveling from j to i. The goal of the problem is to minimize the total route duration as well as a degree of infeasibility in capacity and time window constraints. The constraints are treated as soft constraints, meaning that they can be violated, but is then penalized in the objective function. The time delay and capacity violations are penalized together with the total tardiness in the route. The hope is to direct the search towards better quality solutions. The resulting objective function minimize the total route duration including waiting time, the total number of time windows violations in the route, the number of overloading vehicles, and the total time delay from a vehicle arriving at a location later than it is supposed to. The four elements are weighted together such that the weights together equal one. Hosny and Munford (2010) compared three heuristic methods, two versions of each, to solve the single vehicle PDPTW. The approaches were genetic algorithm, simulated annealing and a hill climbing algorithm. They also presented an intelligent neighborhood move guided by time windows that are incorporated into the three solution approaches. Genetic algorithm is well known for being a robust approach for solving a wide range of problem, including ordering and grouping problem together with highly constrained problems (Hosny and Munford, 2010). The approach specifies the same code for the pickup and associated drop off problem. Furthermore, they use duplicated entries to guarantee the precedence feasibility throughout the search. Problem oriented operators;- (1) the 2-child merge crossover operator guided by time window bounds, (2) a simple gene swap mutation, and (3) an intelligent time window directed swap mutation,- are used during the evolutionary process. The second approach simulated annealing is known for being easy to implement because it only requires a method for generating a move in the neighborhood of the current solution together with an appropriate annealing schedule. In addition, simulated annealing usually manages to transform a low quality solution to one with satisfying quality. The approach uses the same solution representation as the genetic algorithm, and even the two neighborhood strategies are inspired by the GA mutation. The neighborhood uses a random blind move and an intelligent move that is directed by the time window. Their third approach, hill climbing is a simple method that creates a solution which is improved and replaced iteratively. The same solution representation and the directed neighborhood move from the genetic algorithm and simulated annealing algorithm are adopted in the hill climbing algorithm. The solution approaches is tested on instances ranging from 10 to 100 requests. In addition, the problems are tested on instances ranging from 100 to 200 requests so as to better test the operators. Their studies showed that comparing the methods, the genetic algorithm seemed to underperform and the 3-stage simu-

lated annealing clearly outperformed the other methods. The hill climbing method seemed to provide the best result if the problem had to be solved under a short processing time, and may therefore be preferable in real world applications.

To sum up, the most successful exact solution methods centers on branching methods either in combination with cutting planes or with a column generation approach, and often with the help of heuristics during the solution approach. For example, instead of finding the best new column in every iteration, a heuristic finds an improving column to add to the problem in the fraction of the time it takes to find the optimal column. To solve medium to large size instances heuristics are necessary, and in the recent years the studies usually centers on metaheuristic approaches that manages to avoid local optimum during the search.

Table 1: Pickup and Delivery problem

Reference	Objective	Solution Approach	Instance
Cordeau and Iori M. (2010)	min TC[1]	BC[2], VI[3]	$\leq 25r$[4]
Ropke and Cordeau (2008)	min TC	BCP[5], CG[6], VI	$\leq 500r$
Hernándes-Pérez and Salazar-González (2004a)	min TC	BC, VI, benders cut	$\leq 75n$[7]
Hernándes-Pérez and Salazar-González (2007)	min TC	BC, VI	$\leq 100r$
Hernándes-Pérez and Salazar-González (2004b)	min TC	GH[8], IOA[9]	$\leq 500r$
Renaud et al. (2002)	min TC	$7PH$[10]	$\leq 500n$
Gendreau et al. (2006)	min $TCm.m$	TS[11]	$\leq 33r/h$
Gribkovskaia et al. (2007)	min TC	CH[12], NNP[13], $(M)SP$[14]	$16 - 101n$
Alshamrani et al. (2007)	min TC	CH,$Or - opt$	$30n$
Bent and Hentenryck (2006)	min TC	SA[15], LNS[16], two-stage	$600r/90min$
Pankratz (2005)		GA[17]	$\leq 50r$
Hosny and Munford (2010)	min TC	GA, SA, HC[18]	$\leq 200r$

1: TC: Total Costs, **2**: BC: Branch and Cut, **3**: VI: Valid inequalities, **4**: r: requests, **5**: BCP: Branch and Cut and Price, **6**: CG: Column Generation, **7**: n: nodes, **8**: GH: Greedy Heuristic, **9**: IOA: incomplete optimization algorithm, **10**: PH: Perturbation Heuristics, **11**: TS: Tabu Search, **12**: CH: Construction Heuristic, **13**: NNP: Nearest Neighborhood Procedure, **14**: (M)SP: (Modified) Sweep Procedure, **15**: SA: Simulated Annealing, **16**: LNS: Large Neighborhood Search, **17**: GA: Genetic algorithm, **18**: HC: Hill Climbing

2.4 The Dial-a-Ride Problem

One version of the pickup and delivery problems is the dial-a-ride problem (DARP). In the DARP the requests specify a number of passengers to be picked up at a location and transported to their respective drop off location. Since the vehicles transport passengers instead of goods, some form of measure on convenience for the passengers are usually included in the problem, either as constraints in the problem or in the objective function. Some of these characteristics are therefore highlighted in the review below. Exact solution approaches are similar as for the other pickup and delivery problems and are centered on branching approaches usually in combination with cutting planes or column generation methods. Therefore, the review below focus on three contribution of solution approaches centered around branching methods. A contribution combining the use of a heuristic in a column generation approach and thus being able to find new columns faster than finding the optimal new column is also included in the review. Important heuristic contributions for solving the dial-a-ride problem consist of classical cluster first route second algorithms. Other important heuristic contributions are different types of local interchanges, insertion algorithms and construction and improvement heuristics. In recent years metaheuristics are popular heuristic solution approaches. Important approaches are tabu search, simulated annealing, genetic algorithm and parallel insertion algorithm. Several of these contributions are summarized below, mostly considering static problem but with a couple of dynamic studies at the end. For a more thorough review of studies done on the dial-a-ride problem readers are referred to Cordeau and Laporte (2002).

In the study by Cordeau (2003) a mixed integer programming formulation of the dial-a-ride problem is given. The goal of the problem was to minimize the total routing cost. Known valid inequalities for the vehicle routing problem and specific valid inequalities for the dial-a-ride problem is added to the formulation and the problem is solved using a branch-and-cut algorithm. The valid inequalities are found by a separation heuristic. First, the LP relaxation is solved. If the solution is not integer an enumeration tree is constructed and valid inequalities are generated at nodes of the tree by a separation heuristic. Some of the families of valid inequalities are well known and used on the traveling salesman problem, vehicle routing and pickup and delivery problems (for example subtour elimination constraints). Other families of inequalities are custom-made inequalities for the DARP. The model presented was tested on randomly generated small to medium sized instances (up to 32 requests). The instances was solved by the branch-and-cut algorithm and compared with CPLEX 8.1 solution, and the results indicated that the branch-and-cut algorithm reduced the CPU time and the number of nodes explored in the branch-and-bound tree. Ropke et al. (2007) proposed two models for the pickup and delivery problem with time windows and one model for the dial-a-ride problem. The three formulations minimize operation costs of the fleet, and are solved using a branch-and-cut technique. They introduced several families of valid inequalities to strengthen the two formulations. These are incorporate into a branch-and-cut solution algorithm. The solution approach managed to solve instances up to eight vehicles and 96 requests (194 nodes) to optimality. Hu and Chang (2013) formulated a model for the demand responsive transit service

(DRTS) with flexible routes and changeable schedules that aims to minimizing total travel costs. They used Dantzig-Wolfe decomposition technique to decompose the problem into a set partitioning master problem and a constrained shortest path subproblem. These models are solved by a column generation approach that is incorporated in a branch-and-price solution approach. The study focuses on the effects on the objective value, computational time, average pickup delay time, average drop off delay time and number of vehicles used. They found that when increasing the size of the time windows, the objective value decreases slightly while the computational time increases exponentially.

Parragh et al. (2010) considered a static case of the heterogeneous dial-a-ride problem, which minimize total fleet cost. In addition, certain drivers constraints are added to the formulation specify for example lunch break and other breaks for the drivers. Furthermore, service quality in the form of service duration is given implicit in the time windows. That is, the time windows are constructed such that maximum service time cannot be exceeded within the time windows. They study both an exact column generation method based on a set partitioning model and a variable neighborhood search (VNS) heuristic. The first neighborhood uses simple swap operations for the DARP, the second is based on the ejection chain idea, while the third sequences the requests. The column generation approach decomposed the problem into a master problem and a subproblem that were solved with a labeling shortest path algorithm. The variable neighborhood search heuristic was incorporated into the column generation method to make a combinational approach. As test instances Parragh et al. (2010) used the data set A introduced by Cordeau (2006) with 16 to 48 requests. The test instance was then randomly divided among original requests and two types of requests with special needs. This approach managed to solve the instances faster and with less computational effort compared to the true column generation method. In addition, their results improved six out of ten instances tested from previous studies using the instances presented by Cordeau (2006).

Coslovich et al. (2006) proposed a two-phase insertion heuristic. First, they used a simple insertion procedure that allow for quick answers with respect to inclusion or rejection of a new customer. This initial solution is then improved by a local search using 2-opt arc swaps. Coslovich et al. (2006) manage to solve up to 50 requests. Wang et al. (2015) considered the benefits of using ridesharing and especially during rush hours. They propose a $0-1$ integer model for the static dial-a-ride problem with time windows. The objective function considered minimizing four factors weighted together. The three first factors are total ride time for all requests, distance and toll fee. The last factor in the objective function is a cost of having to use a taxi for the requests that cannot be serviced by the ridesharing system. The problem is solved by a tabu search heuristic. First, an insertion heuristic is used to construct initial routes. Afterwards, an adjust-pickup time algorithm is used to reduce passenger ride time by postponing the pickup time of the passengers if the full vehicle has to wait at anothers requests location for pickup. Lastly, a tabu search algorithm improved the routing results, and is run five times since the algorithm contains a randomizing element to diversify the search. The results show

that the cost of the trip should be considerably lowered with increased flexibility of the request. Furthermore, the results show savings in toll costs and total distance. The article considers use of a specific line if ride sharing is used and this can lower ride time which gives the passengers an incentive to use ridesharing.

Cordeau and Laporte (2002) proposed a tabu search algorithm for solving the static dial-a-ride problem with time windows. To avoid cycling, some attributes from the solution is saved so as to not allow visits to these solutions during a predefined period. During the search infeasible solutions are allowed through a relaxation mechanism with self-adjusting penalty coefficients. The tabu search allow for a continuous diversification mechanism so as to avoid ending up in a local optimum. The algorithm is tested on real life instances of 295 requests. In addition, Crainic et al. (2005) proposed a model for the DARP. The goal of the model is to maximize the profit of operating the service (the difference between the benefit of transporting a request and the cost of transporting the request). To solve the DARP they developed a tabu search method and a memory-enhanced, multitrial, randomized constructive procedure. In addition, several hybrid strategies were tested. The results showed that the element of additional memories and probabilistic evaluations in the tabu search enhanced performance. However, the increased randomization in the construction heuristic seems to offer a better diversification strategy. All approaches provide good solutions with limited computational effort. However, even though the tabu search seemed to outperform the multitrial constructive heuristic a hybrid of the two seems to be the best choice in this study. Additionally, Chan (2004) solved a static DARP, with the goal of minimize total travel time and excess ride time, while considering maximum ride time, route duration, vehicle capacity and waiting time. The problem is solved using a cluster-first, route-second approach. The clustering is done by a tabu search or a scatter search assigning requests to vehicles. Then two different insertion based algorithms are considered to determine routes for the vehicles. The problem found optimal solutions on instances of up to 80 requests, and found better solutions than presented in earlier literature to real life instances up to 322 requests.

Bergvinsdottir et al. (2004) developed a dial-a-ride problem which minimize fleet operation costs while satisfying customer service level restrictions. A genetic algorithm using a cluster-first, route-second approach solves the problem. The solution approach alternates between assigning requests to vehicles by using genetic algorithm. A genetic algorithm is an adaptive metaheuristics that mimics the natural selection process. A population of candidate solutions (children) is created, a random element is added by the way of mutations, the best solutions are kept, while the rest is removed, and a new iteration starts. After assigning requests to vehicles (clustering) the vehicle routing is done by using a routing heuristic. The problem solved publicly available data sets, and showed results comparable to previous research.

Mauri and Lorena (2006) propose a static multi-objective mathematical model for the dial-a-ride problem. The goal of the model was to minimize total operational costs and user inconvenience. Operational costs consist of travel distance

used by the vehicles and the number of vehicles used. The factors related to user inconvenience were; route duration, customers' ride time and waiting time at the pickup and drop off locations. Mauri and Lorena considered both homogeneous and heterogeneous vehicle fleets, and with a single and multiple depot for the vehicle fleet. A simulated annealing approach is used to solve the problem. The simulated annealing method is an analogy to thermodynamics and mimics the cooling process of heated atoms. The method is a local search heuristic that incorporates diversification by allowing worsening moves and thus escape local optimal of the search. The balance between intensification and diversification of the search is decided by the temperature parameter. Three types of neighborhood moves are used randomly through a uniform distribution. These are re-order route, re-allocate points and change points. The first type, re-order route consists of choosing a point in a route and selecting a new position for the point and change it to this position. The re-allocate points consists of choosing two routes, pick a request from one of the routes (both pickup and drop off position) and insert it in the other route. The last move, change points, selects two routes and a request in each of the routes and changes these two requests. The model managed to find feasible solutions in short processing time for all test instances. The test instances where presented by Cordeau and Laporte (2003), and have a size up to 13 vehicles and 144 requests.

Attanasio et al. (2004) developed a parallel algorithm for the dynamic DARP. The goal of the model was to find a fair balance between costs of operating the fleet and cost of user inconvenience. First, at the start of the planning period a static solution is constructed based on the known requests using a tabu search algorithm previously studied for a static DARP. When a new request arrives, each of the parallel threads inserts the request randomly in the current solution and run a tabu search algorithm to obtain a feasible solution. Afterwards, a post-optimization is done to check if a better solution is found. The problem is tested on randomly generated instances based on data from Montreal Transit Authority and some real life instances provided by a Danish company. Cremers et al. (2008) considered a dynamic planning problem for transportation of elderly and disabled people, referred to as the day-ahead paratransit planning problem. Since the problem was dynamic some requests are known the day ahead, while some part of the requests becomes known on the day of operation. The problem considered two options for transport; either the request can be transported by the company fleets own vehicles, or the request can be outsourced to a taxi service. The goal of the problem was to minimize the costs of serving the requests, and one way of doing this is to cluster the requests. The problem is formulated as a two-stage recourse model. In the first stage model all *known* requests are considered, while the second stage also considers the requests made available during the first stage. In both stages the requests are first clustered into routes, and then the routes are assigned to vehicles. For the clustering a heuristic is used to split groups of requests into a subgroup for each location. The pickup and drop off location that does not share location with another pickup or drop off are clustered into another subgroup. After this the requests are ordered and assigned using another heuristic. Considering the initial solution a genetic algorithm is initiated and works as follows: The fitness of the individuals in the population (solution) is evaluated. Parents are selected from the

population and children are created. The fitness of the children is evaluated and the acceptable children replace a part of the population (solution). The proposed solution method is flexible so as to adjust for new requests and characteristics of the requests. The problem is tested on instances with 50 requests, and with five to 25 arriving requests and the results is said to be promising.

As a concluding remark, the studies of solving DARPs by exact methods are centered on branching methods in combination with cutting planes and column generation as other pickup and delivery problems. As with PDPs heuristic solution approaches are necessary for solving problems of some size. In the recent years metaheuristics are popular approaches, and studies centered on genetic algorithms, simulated annealing and especially tabu search algorithms, seems to perform well for the DARP.

Table 2: Dial-a-ride problem

Reference	Objective	Solution Approach	Instance
Cordeau (2003)	min TC[1]	BC[2], VI[3]	$\leq 32r$[4]
Ropke et al. (2007)	min TC	BC, VI	$\leq 96r$
Hu and Chang (2013)	min TC	BP[5], CG[6]	$132n$[7]
Parragh et al. (2010)	min TC	CG, VNS[8],	$16 - 48r$
Coslovich et al. (2006)	min TC	IH[9], LS[10], $2 - opt$	$\leq 50r$
Wang et al. (2015)	min TD[11], toll, m.m.	TS[12], 1-cluster, 2-route	$100r$
Cordeau and Laporte (2002)	min TC	TS,	$\leq 295r$
Crainic et al. (2005)	max profit	TS, CH[13]	
Chan (2004)	min TC, UI[14]	IH, TS/SS[15]	$\leq 322r$
Bergvinsdottir et al. (2004)	min TC, UI	GA[16], 1-cluster, 2-route	
Mauri and Lorena (2006)	min TC, UI	SA[17]	$24 - 144r$
Attanasio et al. (2004)	min TC, UI	TS	
Cremers et al. (2008)	min TC	GA, two-stage	$50r+$

1: TC: Total Costs, **2**: BC: Branch and Cut, **3**: VI: Valid inequalities, **4**: r: requests, **5**: BP: Branch and Price, **6**: CG: Column Generation, **7**: n: nodes, **8**: VNS: Variable Neighborhood Search, **9**: IH: Insertion Heuristic, **10**: LS: Large Search, **11**: TD: Total Distance, **12**: TS: Tabu Search, **13**: CH: Construction Heuristic, **14**: UI: User Inconvenience, **15**: SS: Scatter Search, **16**: GA: Genetic algorithm, **17**: SA: Simulated Annealing

2.5 The Pickup and Delivery Problem with Transshipments

An extension of the pickup and delivery problem (or equally, the dial-a-ride problem) is problems that also consider one or several transfer locations. In this way passengers or goods can be transferred between vehicles at these transfer locations. In the following literature review studies focusing on the differences between the regular PDP/DARP and the PDP/DARP with transshipments are highlighted. For example, Nakao and Nagamochi (2010) proposed a model for the pickup and delivery problem without and with a transfer option. The problems were solved by a worst-case analysis, and the option of transshipment is evaluated. They analyzed the lower bounds of travel cost saved by using a transfer point. They found that the bounds were proportional to the square root of the number of routes in an optimal pickup and delivery problem with transshipment, and square root of the number of requests. In detail, if $z(PDP)$ represents the optimal solution not considering transfers, $z(PDPT)$ the optimal solution with transfers, and $|R|$ the number of request the following relationship apply:

$$z(PDPT) > \frac{z(PDP)}{6\lceil\sqrt{|\ R\ |}\rceil + 1} \tag{1}$$

The review below focus on exact solution approaches both branching methods, column generation and the use of cutting planes in combination with commercial software, and the size of the problem solved are thus relatively small. A brief review of metaheuristics is also included at the end, and solves larger sized instances.

Mues and Pickl (2005) proposed a model for the pickup and delivery problem with transshipment where the goal was to minimize the handling costs and traversal costs of all arcs. They used a column generation approach to solve the problem. The master problem is formulated as a set partitioning problem, while the pricing problem found feasible routes (columns) to add to the master problem. The pricing problem, an elementary shortest path problem with resource constraints (ESPPRC) is solved by a dynamic programming method. Different methods are used to reduce the enumeration, for example dominance criteria, the 2-opt and or-opt-edge-exchange algorithms. Using these methods only uniforms tours is created. At last, CPLEX is initiated to solve the mixed integer programming. An initial set of columns is needed when using a column generation method. This initial set of columns is created using a limited enumeration of routes. They mentioned that the result considering one transfer location was promising, but had greater difficulties with multiple transfers. They managed to get a solution time of two to six minutes for 70 loads. Furthermore, Cortès et al. (2010) formulated a mixed integer-programming model for the pickup and delivery problem with transfers. The formulation is a static multi-vehicle problem with the goal of transporting passengers while minimizing total costs and user inconvenience. The operation costs are represented by fleet size and total ride time of the fleet, while the user inconvenience is measured by waiting time and ride time of the requests, in addition to time window violations. At the transfer nodes passengers can transfer between vehicles. The transfer nodes are modeled by dividing the transfer location

into a node for drop off and a node for pickup, and connecting the nodes with an arc. In this way the arrival and departure of the vehicle and precedence of the requests are modeled explicitly. In addition, they allow for multiple visits at the transfer location by each vehicle by duplication of the transfer nodes. As a solution approach Cortés et al. developed a branch-and-cut algorithm based on Benders decomposition. The Benders decomposition applies the combinatorial Benders cuts that were introduced by Codato and Fischetti (2004) and is used to decompose the constraints into a pure integer and mixed constraints. The branch-and-cut procedure is applied to the integer problem, while the real variables and associated constraints generated cuts that are added to the branches. They managed to solve instances up to six requests, two vehicles and one transfer location. Rais et al. (2013) presented a mixed integer-programming model for the pickup and delivery problem with transshipments (PDPT). They assumed a heterogeneous vehicle fleet with a flexible fleet size, and the model allowed multiple depots. The goal of the model was to minimize total fleet operation costs. Furthermore, the transshipments nodes allow unlimited transshipment, and several requests can be associated with the pickup and drop off nodes. In addition, Rais et al. (2013) considered restrictions of the number of transshipments per request, and methods to induce the use of transshipment for each request. The authors also mention the use of transshipments or transfers for the dial-a-ride problem. They emphasized the matching of vehicle flow and request flow. As a solution method, they used a commercial solver that uses a branch-and-bound-and-cut algorithm, where the linear-programming relaxations are solved using simplex. With this, they managed to solve small instances of 14 nodes, and found that the use of transshipment on average led to a better objective value than the solution without transshipments.

Other studies focus on pickup and delivery problem where the load can be split into different shipments. Then the locations for the splitting are similar as the transshipment location in a pickup and delivery problem with transshipments. For instance, Kerivin et al. (2008) proposed a pickup and delivery problem with split deliveries and reloads. At transshipment point the whole or a part of the load can be unloaded and picked up at a later state either by the same vehicle or another vehicle, and this process is referred to as reloads. That is, different parts of the load can be transported using different routes and/or different vehicles, by being split up or transferred at the transshipment point. This point is similar to a pickup and delivery problem with transfers, only that the load can also be split up into several shipments at the transfer point. The goal of the problem was to transport all goods restricted to capacity of the vehicles while minimizing the cost of operating the service. Kerivin et al. (2008) presented a mixed integer programming formulation based on an auxiliary graph. In addition, they presented several valid inequalities and some special considerations are done for the associated separation problem. The problem is solved using a branch-and-cut algorithm, and instances from six to ten locations and five to 15 demands were tested. Instances up to 15 demands and eight locations were solvable. However, the instances with 15 demands was solved using twice the solution time needed to solve the instances with ten demands.

In addition, some studies have been centered on metaheuristics as a solution method for pickup and delivery with transshipment. For instance, Oertel (2000) considered a pickup and delivery problem with an intermediary point, similar to a cross-docking platform, for the loads. The problem was solved used tabu search algorithm for solving problems considering at most one transshipment per request and two potential transshipment locations. This solution approach was tested on instances up to 70 requests, both artificial instances and real world instances from a German car manufacturer. Masson et al. (2014) proposed a dial-a-ride problem with transfers (DARPT), which minimize total fleet operation costs subject to service time restrictions. During the scheduling the passengers can then transfer between vehicles at specific locations. To solve the modeling issues at the transfer point, the transfer nodes are duplicated to two nodes, one for drop off and one for pickup. As a solution approach they used a method based on an adaptive large neighborhood search (ALNS) metaheuristic, which is a method that destroy and repair a solution iteratively in order to improve it. The ALNS used operators that remove requests from routes (destroy) and insert requests in another route (repair). The adaptive part of the search heuristic is that the probabilities of choosing destroy or repair methods are reevaluated periodically depending on their efficiency in past iterations. Masson et al. (2014) implemented the heuristic in C++, and the methods are evaluated on real-life and generated instances from 55 to 193 requests. The lower bound on the gain of using transfers is calculated. The results shows that on average a lower bound on savings from providing transfer points are eight percent, however the savings seems to vary with the instance tested.

Summing up, including transfer points into the PDP/DARP creates benefits for the practical problem but increases the complexity of the problem and only small problems are solvable with exact methods. Using metaheuristics bigger instances can be solved. However, at present the studies performed on the problem with transshipments are limited, and there is still great potential for future studies.

Table 3: Pickup and delivery problem with transshipment

Reference	Objective	Solution Approach	Instance
Nakao and Nagamochi (2010)	min TC[1]	WCA[2]	
Mues and Pickl (2005)	min TC	CG[3]	$70r$[4]
Cortès et al. (2010)	min TC, UI[5]	BC[6], Benders	$\leq 6r$
Rais et al. (2013)	min TC	BB[7]	$\leq 14n$[8]
Kerivin et al. (2008)	min TC	BC, VI[9]	$\leq 15r, 8n$
Oertel (2000)		TS[10]	$\leq 70r$
Masson et al. (2014)		$ALNS$[11]	$\leq 193r$

1: TC: Total Costs, **2**: WCA: Worst Case Analysis, **3**: CG: Column Generation, **4**: r: requests, **5**: UI: User Inconvenience, **6**: BC: Branch and Cut, **7**: BB: Branch and Bound, **8**: n: nodes, **9**: VI: Valid inequalities, **10**: TS: Tabu Search, **11**: ALNS: Adaptive Large Neighborhood Search

2.6 The Integrated Pickup and Delivery Problem

A small number of studies focus on pickup and delivery problem with transfers, where cargo or people can be transferred between vehicles at certain transfer locations. However, there are few studies done on the integrated pickup and delivery problem. In the integrated problem cargo or passengers can be transported between different modes of transport during the trip. The review below focus on practical applications of the problem and uses both exact branching methods and heuristic approaches for solving the problem.

Häll et al. (2009) considered a pickup and delivery problem where a fixed route service is integrated with more demand responsive dial-a-ride vehicles. They formulated an integrated dial-a-ride problem with time windows that minimizes the total routing costs of the vehicles. Furthermore, they proposed several ways to strengthen the formulation. For example, arc elimination rules for the standard dial-a-ride problem and some custom-made arc elimination rules for the integrated formulation are included in the formulation. In addition, they considered a new variable that can be substituted with the vehicle routing variables and thus eliminating a big part of the binary variables. They also proposed some subtour elimination constraints and a heuristics which cluster the locations. Furthermore, they provided a transfer node strengthening which also made it possible to relax some binary variables connected to the transfer locations. The problem was implemented in AMPL and solved by CPLEX 11.0.0 and tested on medium sized instances. Furthermore, they presented an example that showed how the model worked and can be visualized in GIS.

The integrated dial-a-ride problem is similar to the binomial dial-a-ride problem presented by Liaw et al. (1996) which considered paratransit vehicles and fixed route bus systems. They developed a decision support system (DSS) that schedules paratransit vehicle routes. The model was tested on simulated instances and actual data with up to 85 requests and the results indicated a ten percent increase in the number of requests which could be transported and a decrease of ten percent in the number of paratransit vehicles used during scheduling.

Horn (2002) described a software system used to manage deployment of a fleet of demand responsive passenger vehicles, where the passenger can specify a mode of transport. The modes of transport included special services for disabled or aged people, general-purpose maxi-taxi services, ride-sharing arrangements, and conventional taxies. Booking requests can be implemented in advance of travel (static scheduling) or immediate (dynamic scheduling). In a dynamic DARP requests are made available in real time, and are subject to uncertainty and thus have to be efficiently integrated between all parties: between the passengers and the scheduling centers, between the scheduling centers and the individual drivers. The goal of the model was to minimize total travel time, or maximize total passengers transported by the service. First, an atomic insertion is used to insert each trip in the schedule while minimizing marginal cost of the insertion. Afterwards a periodically executed steepest descent improvement procedure is applied to the fleet and a rank homing heuristic incorporating information about future pattern of demand are used as

post insert improvement procedures. The results from the simulations driven in the study indicated that the improvement procedure yielded substantial benefits to the real-time application. Another study considering integrating different modes of transport is Horn (2004) study, which considered a multi-modal problem consisting of modes from the fixed route to the entire demand responsive services. The different modes of transport are a fixed route, which is a service between specific points and on pre-specified timetables. In addition, a smart shuffle that is similar to a fixed route service that only go between points where there is a demand that has been notified. Furthermore, the problem included a roving bus that is a multi-hire free-range service that works between certain points. And last, the entirely demand responsive mode of transport is the single taxi and the multi taxi services which transport a single request or combining requests. To solve the problem an approached based on branching is introduced, specialized bounding and five reduction techniques are implemented to reduce computation time.

Hickman and Blume (2001) discussed how to schedule rules for the integrated service between demand-responsive transit service with a fixed route service, and used a case study from a transit service in Houston, Texas, to show the possible advantages and changes in passenger level of service. Their research incorporated both fleet operating costs and passenger service quality in the model. The customer convenience considered minimizing travel time, transfer time and the number of transfers. They developed a two-stage heuristic that scheduled integrated trips while minimizing operation costs subject to passenger level of service constraints. First, the heuristic found passengers trips taking the passenger from origin to destination while maximizing customer convenience. Secondly, the paratransit trip legs are added to the vehicle schedule using a vehicle routing heuristic. Furthermore, sensitivity analysis is performed on the method. The results showed that the cost savings and consumer convenience are sensitive of the standards of passenger eligibility for the integrated service (if they are able to use the integrated option), the minimum and maximum passenger trip lengths and the assumed penalty for each transfer. In addition, Aldaihani and Dessouky (2003) considered a hybrid routing problem where a fixed route service is integrated in the general pickup and delivery problem, so as to reduce the distance traveled by the demand responsive vehicles while keeping the service quality satisfactory. To solve the problem they used a heuristic algorithm which provided an approximate solution. The solution approach was computationally efficient for solving large sized problems, and is tested on real sized data.

To conclude, though both exact and heuristic solution approaches are studied for the integrated PDP/DARP, the studies performed on this part of the routing problems are limited and usually centered on specific practical cases. Thus, the potential for future studies are great, and the interest for this part of routing problems has increased in later years.

Table 4: Integrated pickup and delivery problem

Reference	Objective	Solution Approach	Instance
Häll et al. (2009)	min TC[1]	BC[2], VI[3]	
Liaw et al. (1996)	min TD[4] m.m.	DSS[5], (BA^*)	$\leq 85r$[6]
Horn (2002)	min TD, m.m.	IH[7]	
Horn (2004)	min TC	BB[8], RT[9]	$40000r/20$min
Hickman and Blume (2001)	min TC,UI[10]	CH[11], two-stage	
Aldaihani and Dessouky (2003)	min TC,UI	AH[12]	

1: TC: Total Costs, **2**: BC: Branch and Cut, **3**: VI: Valid inequalities, **4**: TD: Total Distance, **5**: DSS: Decision Support System, **6**: r: requests, **7**: IH: Insertion Heuristic, **8**: BB: Branch and Bound, **9**: RT: Reduction Techniques, **10**: UI: User Inconvenience, **11**: CH: Construction Heuristic, **12**: AH: Approximation Heuristic

3 Problem Background and Description

This section describes the complexity of the real life problem, gives the problem description, and discusses some important modeling issues to consider before the mathematical model is introduced in Section 3.4.

3.1 Problem Background

Most people travel on a daily basis, for example to and from work, school, healthcare facilities, social events, or other happenings. However, often people are not able to walk or transport themselves to and from their destination. Thus, different sorts of transportation systems have evolved, and with them everything from fixed to flexible routing problems have been studied on a strategic, tactical and operational level. In addition, combining different modes of transport to possible be able to provide a more robust and efficient service have been studied in the literature (for instance see Section 2.6). For example, coordination of timetables between buses and trams arriving at train stations, or the train and buses departing after planes arrive. This section considers different transportation systems and different elements of uncertainty inherent in the system.

3.1.1 Transportation Systems

Several public transportation systems are available, and the different types can be characterized on a scale from fixed route systems to entirely demand responsive systems (Errico et al., 2013). The most basic is the fixed route services, which consists of a specific traffic network. The fixed route departs from the given locations at specific times. Examples of fixed route systems are the public system in the cities, or more specific, the public transportation system ruter.no in Oslo or the train service NSB in Norway. The greatest problems considering the fixed route system is deciding where the vehicles should stop, and the departure time and rate for the route. Once these details are decided there is only a few ad-hoc problems to consider for the driver, for example disturbance on the route causing for example delays, and problems considering deviations from the drivers workshift. For the users, the fixed route systems are among the less convenience modes of transport since they have to transport themselves to and from the stop locations for the fixed route and they have to make sure that they are at the stop before the departure of the vehicle. However, considering the relatively uncomplicated process of organizing the service for the operator, the price the users have to pay is relatively low and for users who value low price above convenience in time and place this might be a attractive mode of transport. For some fixed routes the departures might also be so frequent that driver does not have to consider the time table for the fixed route. This makes the service more convenient for the user as they can leave at their own account.

Next, the hail-a-ride service is similar to the fixed route services when relaxing the specific stops (Errico et al., 2013). That is, the vehicles have specific departure

times and drive a predetermined and fixed route, but can pickup passengers along the whole route and not just at predetermined stops. For the operator the service is only slightly more complicated to operate. The routes and timetables are still fixed, but the increased flexibility creates more room for problems and deviations which has to be considered. In addition, the drivers job is slightly more complicated as they have to consider the whole route as a stop and not only some predefined stops. This creates flexibility for the users as the average distance to the service is reduced. Furthermore, the service should still be a cheap alternative for the users, though the price might be a bit higher. An extension of the hail-a-ride service is when the vehicles are able to do short detours during their route based on requests made in advance, so called route deviation systems (Errico et al., 2013). In short, this increases the work for the operator to maintain a liable service as the deviations have to be accounted for during limited time. For the user the convenience of the service increases as those who prefer to be picked up closer to their pickup location can pay a extra fee for this service, while those who prefer a cheap fixed route can use the predefined route.

On the other half of the scale, towards the more demand responsive services, a group of services which provide a more flexible transportation service exists. These services can be more demand responsive considering departure times, load and the number of people to be carried and pickup and drop of points, together with privacy during the service. For the operators point of view, these problems are more complicated to schedule and operate. The operator has to find new routes and departure times, and the driver have to visit new places at new times. Thus, there is no stability in the system, and the operator has to solve different problems for each day. On the other hand, the convenience for the user is high as they can be picked up at the place they want and within the time window they specify. For instance some part of the public, that is mainly elderly and disabled do not manage to transport themselves to and from the public station. Others prefer the convenience of being picked up at home and delivered at their final destination, above the lower cost of a fixed route. For some part of the public, a fixed route might not exist for the trip the consumer is undertaking, or they are traveling at a time when there are few or none departure of the existing fixed route.

The entirely demand responsive system is a system where the vehicle fleet picks up the requests and transport it to its drop off location (Errico et al., 2013). The service thus has no predefined stop, routes or timetables to follow and the operator has to schedule the vehicle fleet in advance of each shift. Furthermore, the driver has to drive different routes and have to fulfill new requests each shift. This complicates the process of operating the service and the service is expensive to keep operating. On the other hand, the service provides great flexibility for the users, though at a higher service cost. The demand responsive connector is a demand responsive service where vehicles are assigned different areas and certain transfer points with connect to areas for other vehicles (Errico et al., 2013). Thus, if the request is traveling from one area to another area, one vehicle picks up the request transport it to the transfer point between the areas where another vehicle picks up the requests and transports it to its drop off location. A less demand respon-

sive but flexible system is the point deviation service, where only a few points are scheduled and the rest of the service works on a demand responsive service (Errico et al., 2013). These systems creates more stability for the operator and driver, but are still quite expensive to operate. On the other hand, the convenience for the user is great, though the price might be a bit high.

Several integrated transportation systems also exist. For example, schedule busses and trams so their arrival corresponds with departures or arrival of trains, or schedule trains and buses with departures and arrivals of planes at the airport. Also the aspect of integrating a demand responsive service with a fixed service might provide a good alternative for the entirely demand responsive service. For instance, in Trondheim one can use a combined system by taking the taxi to the airport buses and taken the airport buses the last part of the trip. The resulting trip is cheaper than a taxi service while maintaining most of the flexibility and user convenience of the service. Similar systems might also be possible for the elderly and disabled with special permit provided by the state, and thus lower the cost of these systems. However, it is necessary to keep in mind that those who will not manage to make the transfer still has to be transported from their pickup to drop off location without transfers.

To sum up, transportation services range from completely fixed route service to entirely demand responsive services. In addition, there have been a development of mixed systems which considers both fixed route services and demand responsive services to provide a flexible and cheap service for the passengers.

3.1.2 Complexity of Routing Problems

For the operator it is a complex assignment to provide a transport service that is flexible and demand responsive, and at the same time cheap. For a demand responsive service the operator has to decide which requests to perform in-house and which to outsource. A request usually specifies a number of people traveling and if there are any special requirements associated with the request. An example of a special requirement can be wheelchair access or special luggage which has to be transported on the same trip. Furthermore, the customer can specify an earliest pickup time and/or arrival, or they can specify pickup "as fast as possible". In addition, the operator has to decide how big the vehicle fleet are suppose to be, and what types of vehicles the fleet should be composed of. Should all vehicle be equal, or should they have different characteristics, and what should these characteristics be. In addition, the operator has to decide which vehicle should transport which requests and in what order the requests are to be fulfilled during the route. After deciding on the schedule the operator has to supervise to make sure that the vehicles manages to perform their routes as they are suppose to, and that all requests are fulfilled and the customers found the service satisfying. Furthermore, requests might be made available or canceled while the fleet is operating, and characteristics, for instance pickup time or drop off location, of a request might change during operation. For the fixed network the operator only has to make sure that

the drivers are able to perform their routes satisfactory. This is because the routes with stops, the time tables and the size of the vehicles are already set. However, the process of finding the best routes might be quite difficult. Furthermore, after deciding on a set of routes, the constructor has to decide on the timetables and the capacity of the trip. For example, there might be more frequent departures during rush hours and/or the size of the vehicles might be greater during rush hours.

However, both services are exposed to several types of uncertainty. For instance, a demand responsive service depends on road access, which can easily be obstructed without warning. Bad weather, accidents, road work and other incidents can force roads to close, or change the time and cost of a route. Furthermore, the road conditions and the areas, in addition to the quality of vehicles and experience of drivers influence the time and cost of a route. In addition, a driver might be disabled or have other reasons for not being able to come into work or a vehicle might break down or need service during normal operation time. Similarly, the fixed route network is exposed to many of the uncertainties discussed above. In addition, both services are affected by the density and congestion of traffic, and service time is usually longer during high-demand periods.

Furthermore, when the operator sets the schedule for the driver they might have to consider requirements from the drivers and the customers. Examples of these specifications might be that the drivers might prefer special routes, some might prefer city driving, some high way driving. Furthermore some might prefer driving the morning, day or night shift. In addition, some prefer several short breaks while other prefer one long break. The operator will probably not manage to satisfy all requests but should consider these when scheduling the routes. Considering the customer some customers value a cheap cost of the trip, while others prefer short service time, short time windows for pickup and/or drop off, no detours, transfers or the like. As a consequence, if the operator maintains a satisfactory level of the customers service for the factor which the customer sets the highest, their requests experienced quality of the service is increased.

To sum up, there are therefore several considerations to make when providing the demand responsive and/or fixed route service. In a demand responsive service the operator has to decide on the routes for the vehicle fleet and schedule the requests so all requests are fulfilled with a certain service quality. In addition, several elements of uncertainty, as for example road work, changes in the requests and congestion, have to be accounted for. In comparison, as long as the fixed routes and time tables are set for the fixed route service, only the ad hoc and uncertainty elements need to be considered for the operator. As a consequence, the price of operating a demand responsive service is usually quite expensive compared to a fixed route service.

3.2 Problem description

In the integrated dial-a-ride problem a set of requested journeys are to be scheduled using a fleet of demand responsive vehicles and if possible by using a fixed route service for some part of the journey. The problem is deterministic and it is assumed that all requests are known before scheduling begins.

The requests:
Each request has a specific pickup and drop off location. The pickup location and/or drop off location can be associated with a transfer location, and it is assumed that the cost and time passed when traveling from a pickup or drop off location to its associated transfer location is zero. The request also specifies a maximum walking distance between a pair of locations. Thus, if the distance between two locations is less than this distance the request is able to walk between the locations. The request can walk either in or out from a location, but not both. An average and fixed walking velocity is used for all requests. It is assumed that all passengers manage to do the transfers even if their maximum walking distance is set to zero. In addition, there are specified time windows for the respective locations, and a maximum service time for the requests. Furthermore, each request takes up a given load (number of passengers) of the vehicle during the service. It is assumed that the customers in the same request cannot be split up and transferred in different vehicles, and thus the model is restricted to requests with load less than an empty vehicle's capacity. Each request has to be serviced and can be carried out by a demand responsive vehicle from the pickup to the drop off location, or the vehicle transfers the passengers to a transfer location where a fixed route service is operating. The fixed route takes the passengers to the destination point or another transfer location where a dial-a-ride vehicle transfers the passengers the next or last part of the trip. A further possibility is for the request to walk some part of the trip, for example walk from (to) their pickup (drop off) point to (from) a transfer location or another location nearby, or they can walk between two locations during the trip.

The vehicle fleet:
The fleet of demand responsive vehicles is homogeneous. Thus, the vehicles are assumed to be identical in terms of cost and in terms of their usage and load capacity. The usage capacity is the amount of time each vehicle can be used (that is, the maximum time away from depot), while the load capacity specifies the maximum number of passengers that can be transported by the vehicle at any time. Furthermore, there is assumed a maximum number of vehicles, all stationed at the same depot. However, the start time of the vehicle is individual for each vehicle and is set to the time the vehicle leaves depot. Furthermore, the problem assumes that no disruption, for example accidents or breakdowns, can occur.

The fixed route:
The fixed route network can consist of one connected network or many separate networks. If there are separate networks the requests cannot transport themselves from one network to another network without being transported by a vehicle if not the walking distance between two locations in their separate networks are less than

the specified allowable walking distance by the request. There are no capacity or usage limits on the network. Furthermore, the network has its own velocity, and the mode of transport (i.e. boat, train, tram etc.) is not specified. It is also assumed that no disruptions can occur during the service, and that the fixed route service is assumed to leave right after the customers arrive at the transfer point.

The objective:
Most studies focus on minimizing the costs of the service, which can be divided into two main groups. (1) The first cost is the cost associated with operating the fleet (cost per vehicle used, variable and fixed operational costs (e.g. wages) and cost of extra vehicles etc.). (2) The other criteria is the quality of service which includes route deviation, route length, customer waiting time, customer ride time, and difference between actual and desired drop off times. This thesis considers the operator and drivers point of view and how they can manage to create cheap and efficient routes for the vehicles. At the same time, the users are considered through the use of penalties, so as to maintain an acceptable quality of the service. The objective is then to minimize the costs of operating the fleet and the costs associated with increased user inconvenience. The fleet operating costs depend on the number of vehicles used, the total distance travelled and the total vehicle usage. The penalties paid for increased inconvenience is associated with the total number of transfers, total increased service time or the total distances walked.

3.3 Modeling Issues

In this section previous studies and modeling techniques are discussed before specifying which approach is used in the formulation described in Section 3.4.

3.3.1 Static and Dynamic Models

The integrated dial-a-ride problem can be divided into two versions, static and dynamic, based on how the requests are made available. In a static version, called off-line scheduling, all requests are made available before the optimizing is initiated. In the dynamic version, called on-line scheduling, some requests are available initially, and some are made available in real-time. As a consequence, the problem can usually be re-optimized at some point, and the program can consists of a sequence of static problems. When looking at realistic and detailed models it is often possible to first solve the static problem and then find methods for re-optimizing the problem (Savelsbergh and Sol, 1995).

Used in the IDARP formulation:
As mentioned in Section 3.2 a static situation of the problem is considered in this formulation, and it is assumed that all requests are known before scheduling.

3.3.2 Locations and Nodes

In a classical dial-a-ride problem (DARP) there are three types of locations. The depot is where the vehicles are stationed and thus where they depart from and return to. In addition, there is a set of pickup locations, with a corresponding set of drop off locations. Each request has a pickup location and a corresponding drop off location. For an integrated dial-a-ride problem there is an additional set of locations, transfer locations (Figure 1), which specifies drop off and pickup locations for the fixed service. The pickup location and/or drop off location for a request can be associated with a transfer location, and it is then assumed that the locations are physically located at the same place. In this model it is also allowed for an individual radius for each request around the locations, where the passengers are able to travel on their own.

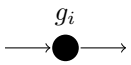

Figure 1: Transfer node g_i

It is common practice when modeling the transfer locations to duplicate locations that might be used several times so each duplication can only be visited once. A set of artificial nodes are created for each physical location allowing several visits to each location but only one visit per node. Then, if there are two requests located at the same node, either pickup or drop off, or if several requests are using the same fixed route, the corresponding node can be duplicated to allow for two visits. The transfer locations can be modeled in different ways, and a few of these methods are mentioned below. In some studies they split the transfer locations in two, one node for drop off and one node for pickup, and connect these nodes with an arc (Masson et al., 2014). Then, for every route which is using the transfer location a set of pickup and drop off location for the transfer location is created. In this way the vehicle enters the drop off node, drops off customers if the vehicle is supposed to, moves cost- and time-free to the pickup node picking up customers if it is supposed to, and continues on its trip to the next node. With this modeling several vehicles can visit the transfer nodes, one or several times since a pair of nodes are created for each vehicle route using the transfer locations. This is illustrated in Figure 2.

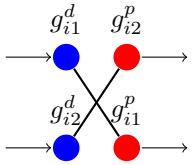

Figure 2: Transfer node divided into one drop-off (blue) and one pickup node (red) for each route

Another methods is to calculate the maximum number of visits that can be

made (Cortès et al., 2010), here this number is represented by M_i. For instance, it is intuitionally easy to see that the maximum number is never greater than the total number of requests. For each of these possible visits a set of transfer nodes (one drop off and one pickup for each visit) are created, as illustrated in Figure 3. Then, each node within the location can only be visited by one vehicle at most one time. As shown in the Figure 3, the vehicles have to follow the black lines and visit the drop off node g^d_{im} and the corresponding pickup node g^p_{im} before leaving the transfer location. The passengers on the other hand can travel cost-and time-free from the drop off node they arrive in g^d_{im} to any of the pickup nodes within the location, shown with green lines. If they travel to the corresponding pickup node g^p_{im} the passengers have to follow the vehicle. There is no need for the request to travel from a pickup node to a drop off node, so the flow is restricted to going from a drop off node to a pickup node. Furthermore, the passengers can travel to another transfer location as long as it is connected to the one they are at.

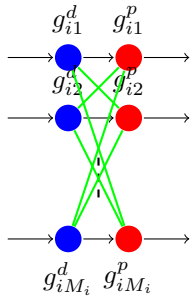

Figure 3: Transfer pair of nodes created for maximum possible visits

Furthermore, it is possible to assign a set of transfer nodes (drop off and pickup) for each request for each transfer location (Häll et al., 2009). This approach is illustrated in Figure 4. A vehicle has to visit the drop off node which corresponds to the request to drop off customers for that request. Afterwards, if the vehicle is dropping off another request the vehicle has to go to the drop off node corresponding to that request. Then, if the vehicle is supposed to pickup customers it has to travel to the pickup node corresponding to that request, and so on. Each node corresponding to a request cannot be visited more than one time, and only by one vehicle. It is possible to restrict the vehicles to visit the nodes within the location in a systematic way to limit the number of symmetric solutions. For example, the vehicle should visit all drop off nodes before pickup nodes, and that the nodes should be visited in increasing order. By using this form of sequencing the vehicle flow is restricted to going from the upper left corner to the lower right corner of the Figure 4. The customers travel from their drop off node, to a transfer location which is connected to the one they are at, or they travel cost- and- time free to their corresponding pickup node within the transfer location and waits to be picked up by another vehicle. The black lines illustrate the feasible flow of vehicles, while the green illustrate feasible flow for the requests.

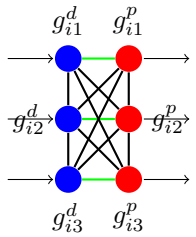

Figure 4: Transfer pair of nodes created for each request

It is difficult to know which modeling approach results in the most efficient formulation. If the maximum number of transfer visits per location is considered the total number of nodes are probably less than if each request gets their own set of nodes. However, the methods creating the maximum number of visits per location is more difficult considering reducing symmetric solutions.

Used in the IDARP formulation:
In this thesis a variation of the method showed in Figure 3 is considered, and is illustrated in Figure 5. For each transfer location g in the set of transfer location $\mathcal{G}$ a set of nodes $\mathcal{N}_g^G \in (1, \dots M_g)$ is created, where M_g represents the maximum number of visits per transfer location g. Each of these artificial nodes can only be visited once, and a vehicle travels to a node (g, m) drops off people if it is supposed to, picks up people if it is supposed to, before it leaves. This has to be done within the constructed time window, and if necessary a vehicle can wait at a node until service is allowed to start or to a passenger arrives. In addition, a customer can be dropped off by one vehicle, travel to another node (cost- and time-free), within the same location, to be picked up by another vehicle. Furthermore, the requests can travel to any transfer location connected to the transfer location they are in. For each request that starts (or ends) in a transfer location a pickup (or drop off) node associated with the transfer location g is created. However, only the request belonging to the associated pickup or drop off location can travel between the transfer location and the associated location. For instance, a passenger can travel cost- and time-free from a transfer node (g, m) to the drop off node associated with it, as illustrated in the Figure 5. The black lines correspond to feasible flow of the vehicles, while the green lines illustrate feasible flow of the requests.

In this model the maximum number of node visits can be set to be individual for each transfer location. This is because in a fixed route network there might be one or a few locations which connect several lines and/or is centered in the heart of the city or for example near a hospital. Therefore the number of visits at these important points can be set higher than the transfer locations in less populated areas. This solution reduces the size of the problem compared to the option of setting the maximum number of node visits equal to the high visitation number at the few central locations.

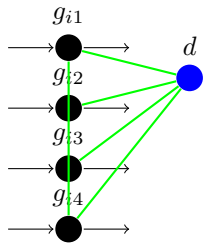

Figure 5: Set of transfer nodes created for maximum possible visits and corresponding drop off node (blue)

3.3.3 Time Windows and Time Constraints

In a dial-a-ride problem the customer usually specifies a desired pickup time or/and a desired drop off time. Given these times it is often assumed that the customer allow for pickup after desired pickup time, and allow for drop off before desired drop off time (Savelsbergh and Sol, 1995). Furthermore, this information can be used to develop so-called soft time windows or hard time windows (Cortès et al., 2010). Soft time windows are windows that can be violated, but if it is violated a penalty is added to the objective function. Hard time windows on the other hand must be strictly followed. A vehicle can arrive earlier than earliest service time, but have to wait at the node until service can begin within the specified time window. The time windows significantly complicate the problem of finding an initial solution to the problem, but since the solution space is smaller the optimal solution might not be more difficult to find. There is often a maximum service time associated with each request. In addition, the customer might specify a maximum deviation or error rate. That is, the service time cannot be above the direct travel time plus a fraction of the direct travel time, or the service time can exceed direct travel time plus the fraction, but then a penalty has to be paid. Furthermore, some studies have considered so-called deadhead restrictions, which restricts the waiting time for the customer in a vehicle (Savelsbergh and Sol, 1995).

The time windows can be constructed by several methods. If the customers specify a wanted arrival time (and departure time) and a maximum deviation, time windows can be constructed given earliest and latest arrival (departure). Liaw et al. (1996) used both maximum deviation of desired drop off time (MD_i) and maximum allowable excess riding time of a request (ME_i) to calculate the time windows, where:

(desired drop off time) - (actual drop off time) $\leq MD_i$

(actual riding time) - (direct riding time) $\leq ME_i$

They use a desired drop off time to calculate the time window for drop off $[\underline{T}_{\overline{r}+i}, \overline{T}_{\overline{r}+i}]$, where the latest drop off time $\overline{T}_{\overline{r}+i}$ corresponds to desired drop off time. Then, earliest and latest departure time $[\underline{T}_i, \overline{T}_i]$ can be calculated by consid-

ering the direct and shortest travel time $T_{i,\overline{r}+i}$ and maximum service time S^{max}. This maximum service time S^{max} is equal to Liaw et al. (1996)'s maximum allowable excess riding time (ME_i). Earliest departure is earliest arrival less maximum service time ($\underline{T}_i = \underline{T}_{\overline{r}+i} - S^{max}$), while latest departure time is latest arrival less direct travel time ($\overline{T}_i = \overline{T}_{\overline{r}+i} - T_{i,\overline{r}+i}$). The maximum service time can be written as $S^{max} = T_{i,\overline{r}+i}(1+E)$ (Liaw et al., 1996), where $E \in [0,1]$ is an error rate. The error rate represents the maximum deviation from direct travel time.

Used in the IDARP formulation:
This thesis considers hard time windows. In addition, it is assumed that the maximum service time for the request is $S^{max} = T_{i,\overline{r}+i}(1+E)$. The requests have to satisfy both the time window and the maximum service time, and this is done by register the arrival and departure time at each node. The approach is illustrated in the Figure 6. With the notation used in the model: Let $\underline{T}_i$ denote earliest service start at location $i \in \mathcal{P}$, and $\overline{T}_i$ denotes the latest service start at the location. Then, $\underline{T}_{\overline{r}+i}$ and $\overline{T}_{\overline{r}+i}$ is the interval for arrival, set according to the customer preferences. Given the arrival time window $[\underline{T}_{\overline{r}+i}, \overline{T}_{\overline{r}+i}]$, the time window for departure, $[\underline{T}_i, \overline{T}_i]$, can be calculated as

$$\underline{T}_i = \underline{T}_{\overline{r}+i} - T_{i,\overline{r}+i}(1+E) \qquad i \in \mathcal{P} \tag{2}$$

$$\overline{T}_i = \overline{T}_{\overline{r}+i} - T_{i,\overline{r}+i} \qquad i \in \mathcal{P} \tag{3}$$

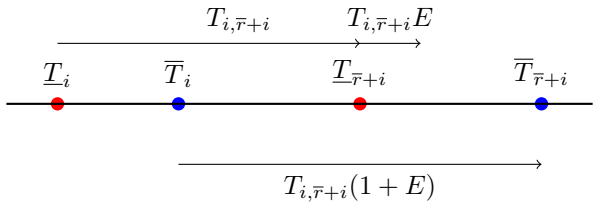

Figure 6: Time Windows

3.3.4 The Vehicle Fleet and Routes

Earlier studies have been done on both homogenous and heterogeneous fleets and/or with one or multiple depots (Savelsbergh and Sol, 1995). In a homogeneous fleet all the characteristics of the vehicles in the fleet are equal, that is, they have the same load and usage capacity etc. A heterogeneous fleet, on the other hand, the vehicle characteristics are different and they might have different depots. Furthermore, some studies have been done on separate time windows for the vehicles so that the drivers, for instance, have their separate scheduled breaks (Savelsbergh and Sol, 1995).

The arcs and nodes visited by the request from it is picked up to it is delivered to its final destination point is defined as a trip. A route for a vehicle is the part of the

trips which are undertaken by the vehicle. In addition to capacity and time window constraints, the scheduled trips must satisfy pairing and precedence constraints on pickup and drop off locations. As for other pickup and delivery problems (PDP) the pickup location must precede the drop off location for a request, and pickup and drop off for each part of the trip has to be associated with the same vehicle. That is, if the customer is transported the first part with a demand responsive vehicle, the next with a fixed route service, then a demand responsive vehicle to the final drop off, then the first part of the trip has to be done by the same vehicle and similarly with the last part of the trip.

Used in the IDARP formulation:
This thesis focuses on a homogeneous fleet with a single depot and a continuous usage time of three full eight hours work shift.

3.3.5 The Fixed Network

At present, most studies done on routing problems consider one mode of transportation. There are a few studies considering different flexible transportation systems and quite recently there has been a focus on combining a fixed route and a demand responsive service. For example, Horn (2002) considers several modes of transport, both variants of the taxi service, services operating between specific stop and time tabled bus systems. Similarly, Errico et al. (2013) provide a review of transportation modes from fixed to flexible systems. The systems ranged from completely fixed, both in routes and schedules, fixed in routes or fixed in schedule, or completely flexible systems. Furthermore, Liaw et al. (1996) studied a binomial dial-a-ride problem with a flexible vehicle fleet and a bus system with fixed routes and schedules. Similarly, Häll et al. (2009) consider a fixed network where the transportation mode departs when passengers arrive at a transfer location. It is assumed that the capacity is well above the demand so that the service never reaches its full capacity. Furthermore, using the service is assumed cost free. It is also assumed that the velocity of the fixed route never exceeds the vehicle velocity. The network used in this model is inspired by this work.

Used in the IDARP formulation:
The fixed network consists of several stops (locations) and with an arc going directly between stops. Passengers can travel from i to j or from j to i for any pair of locations (i, j). The cost of traveling with the fixed route service is assumed zero, and it is assumed that demand will always be less than capacity. The variable w_{imjnr} is set equal to 1 if the request enters the fixed route at (i, m) and travels to location (j, n) where it either has reached is final destination, are picked up by a vehicle or walks to another location.

3.3.6 Requests

The requests can have certain requirements specified that restrict the solution space. These specifications are given in the data instances, and can for example

be maximum service time, maximum number of transfers, time windows for departure and/or arrival, passenger load and/or special luggage and other special requirements. Furthermore, if the request are to be transported with an integrated service it might be necessary to either assume that they are able to do the transfer or that the system divide the requests into a group which can use the fixed route and a group which needs to be transported door-to-door. Other requirements can be if the passengers in one request are allowed to be split or have to be transported on the same trip.

A special consideration is needed if the total load of one request exceeds the capacity of the vehicles. To solve this problem a new and separate request for the load exceeding capacity can be created, and then this procedure can be repeated if the new request is above capacity as well. Another method is to split the load into two new requests instead of the old request. Special care is needed if the load is odd, since the requests have to have integer loads. In addition, the procedure has to be repeated if one of the new requests is above maximum capacity. Furthermore, it can be specified if the new divided requests have to be picked up and/or delivered at the same time, and/or have to take the same route. Anyhow, both modeling approaches consider equal physical locations of pickup and drop off. It can for example allow for several visits at the same location by creating several node visits at the location, similar to the transfer nodes. Another possibility is to allow the operator to decide on the number of passengers to be picked up at each visit as long as the total number of passengers are transported from their pickup to drop off location. This provide flexibility for the operator during scheduling and can be modeled similarly as studied by Kerivin et al. (2008)

Used in the IDARP formulation:
The model described in this thesis considers a pickup and drop off location, a load, and a time window for the departure. The load is assumed to only consist of people and is thus integer. It is assumed that load cannot be split into different trips, and it is assumed that the load is less than vehicle capacity for all requests. Furthermore, each request specifies a maximum walking distance between a set of locations. The request can walk between any locations which have a distance less than the maximum walking distance, but cannot walk both in and out of any location. Thus, a request can for instance walk from the pickup location, between two fixed networks, to the final destination, or between other locations if necessary. It is assumed that all requests are able to transfer between different modes of transport.

3.3.7 The Objective Function

Since dial-a-ride systems often are highly subsidized systems, cost minimization is usually the main objective of the problem. The main element of the cost is the cost of operating the fleet, i.e. the number of the vehicles used and the usage of the vehicles in time or distance. The cost of the usage is nonlinear in real life, and depends on the road conditions, weather conditions and the conditions of the vehicle and driver. Many studies, however, assumes a linear usage cost depend-

ing on distance. In addition, some studies are associated around minimizing user inconvenience. For instance, Jaw et al. (1984) considered the amount of time the pickup and drop off time could deviate from the desired pickup or drop off time, together with maximum service time. Furthermore, Grönross (1984) (Paquette et al., 2012) divide service quality into two groups, one for technical quality and one for functional quality. Technical quality corresponds to what the consumers receive from the service. In a dial-a-ride problem this corresponds to the interaction between the user and the service during the trip (from pickup to drop off), that is the product of the service. In addition, the functional quality corresponds to the service experience, the process, for the user. Technical quality is necessary for qualifying customer satisfaction, while the functional quality is necessary for good and excellent customer satisfaction.

Used in the IDARP formulation:
The objective of this model is to minimize the cost of operating the vehicle fleet and the costs associated with user inconvenience. The fleet operating costs consists of the number of vehicles used and the total cost of usage of the vehicles. The usage cost is assumed linear dependent on distance and a linear cost dependent on the time the vehicle is away from the depot. Furthermore, the objective considers minimizing some factors of user inconvenience. A linear cost or penalty is paid for the total distance above direct travel time between pickup and drop off location for each request, for the total distance walked by all requests and for the total number of transfers for all requests.

3.4 The Integrated Dial-a-Ride Model

There is a set of requests $\mathcal{R} \in (1, 2, \dots \overline{r})$, where $\overline{r}$ denote the number of requests. Each request r consists of a pickup location r and a drop off location $\overline{r} + r$. Each request r also specifies an integer load L_r, which states the number of passengers. Furthermore, there are a set of pickup locations $\mathcal{P} \in (1, 2, \dots \overline{r})$ and a set of drop off locations $\mathcal{D} \in (\overline{r} + 1, \overline{r} + 2, \dots 2\overline{r})$. In addition, there are a vehicle depot and a set of transfer locations $\mathcal{G}$. For each pickup and drop off location i a node $(i, 1)$ is created, resulting in the corresponding sets of nodes, $\mathcal{N}^{\mathcal{P}}$ and $\mathcal{N}^{\mathcal{D}}$ respectively. Furthermore, for each transfer location g a set of nodes $\mathcal{N}_g^G \in (1 \dots M_g)$ is created. The node $(g, m) \in \mathcal{N}_g^G$ represent the m^{th} visit to transfer location g, and M_g represents the maximum number of visits at the location. Furthermore, $g(i)$ denotes the transfer node that the pickup or drop off node $(i, 1)$ is associated with. The depot consists of two nodes, the vehicles leave from node $(0, 1)$ and return to node $(2\overline{r} + 1, 1)$.

Then, the directed graph $G = (\mathcal{N}, \mathcal{A})$, $\mathcal{N} = \mathcal{N}^P \cup \mathcal{N}^D \cup_{g \in \mathcal{G}} \mathcal{N}_g^G \cup \{0, 1\} \cup \{2\overline{r} + 1, 1\}$, is the set of all nodes, and each node $(i, 1) \in N^P \cup N^D$ has an associated time window $[\underline{T}_i, \overline{T}_i]$. Each vehicle has a capacity Q and a maximum usage time U. However, the vehicles are free to start and end their route individually, as long as the times satisfies the other constraints of the problem. An arc between locations i and j has an associated travel time T_{ij} and an associated cost C_{ij} and a

travel distance between the locations D_{ij}. The travel time, distance and costs are assumed symmetric, that is $C_{ij} = C_{ji}$. In the cost C_{0j}, that is the cost of traveling from the depot location to a location j, the fixed cost of using a vehicle is included. In addition, a cost for the time the vehicle is away from depot. Furthermore, the fixed route has its own velocity and thus time to travel from node (i, m) to node (j, n). In addition, each request r specifies a walking distance D_r^R for which they are willing to walk by themselves. This means that a request can walk from location i to location j, if the distance is less than their maximum allowed walking distance ($D_r^R \geq D_{ij}$). Furthermore, if a request r walks to a location j it cannot walk out from that location to another location h. The model is also limited to pickup or drop off the request associated with the location j when a vehicle visits the location. For example, when a vehicle k visits the pickup location j to drop off request r it also have to pickup request j associated with location j. Given this the model wants to schedule all requests while minimizing the vehicles used, their total usage and distance traveled. In addition, the problem minimize three factors of user inconvenience, the total time above direct service time, total walked distance by the requests and the total number of transfers.

Notation:
In the formulation capital letters is used to write sets, while the indices on parameters and variables are given in lower case subscripts. Parameters are written in capital letters, while lower case letters illustrate variables.

3.4.1 Sets

$\mathcal{P}$	Set of pickup locations
$\mathcal{D}$	Set of drop off locations
$\mathcal{G}$	Set of transfer locations
$\mathcal{N}$	Set of all nodes
$\mathcal{N}^{\mathcal{P}}$	Set of pickup nodes
$\mathcal{N}^{\mathcal{D}}$	Set of drop off nodes
$\mathcal{N}_g^G$	Set of transfer nodes corresponding to transfer location g
$\mathcal{R}$	Set of requests
$\mathcal{K}$	Set of vehicles

3.4.2 Parameters

P^S	Penalty cost for service time above direct travel time for each request
P^R	Penalty cost for each request walking a certain distance
P^T	Penalty per transfer for each request
T_{ij}^K	Travel time when using a vehicle from location i to location j
T_{ij}^F	Travel time when using a fixed network from location i to location j
T_{ij}^W	Travel time when walking from location i to location j
D_{ij}	Distance of traveling from location i to location j
C_{ij}	Cost of traveling from location i to location j using a vehicle
C^U	Cost per time unit of using each vehicle
Q	Capacity of a vehicle
U	Maximum usage time of a vehicle
D_r^R	Allowed walking distance by request r
L_r	Load of request r
$\underline{T}_i$	Earliest time at which service may begin at location i
$\overline{T}_i$	Latest time at which service may begin at location i
Z_{ij}	$\begin{cases} 1 & \text{if there exists a fixed route from transfer location } i \text{ to transfer location } j \\ 0 & \text{otherwise} \end{cases}$
F_{ir}	$\begin{cases} 1 & \text{if node } i \text{ is the pickup node of request } r \\ -1 & \text{if node } i \text{ is the drop off node of request } r \\ 0 & \text{otherwise} \end{cases}$
E	Fraction of direct service time the total service time is able to exceed

3.4.3 Variables

x_{imjnk}	$\begin{cases} 1 & \text{if vehicle } k \text{ travels from node } (i,m) \text{ to node } (j,n) \\ 0 & \text{otherwise} \end{cases}$
y_{imjnr}	$\begin{cases} 1 & \text{if request } r \text{ travels from node } (i,m) \text{ to node } (j,n) \text{ on a vehicle} \\ 0 & \text{otherwise} \end{cases}$
w_{imjnr}	$\begin{cases} 1 & \text{if request } r \text{ travels from node } (i,m) \text{ to node } (j,n) \text{ using a fixed network} \\ 0 & \text{otherwise} \end{cases}$
v_{imjnr}	$\begin{cases} 1 & \text{if request } r \text{ walks from node } (i,m) \text{ to node } (j,n) \\ 0 & \text{otherwise} \end{cases}$
t_{im}^A	arrival time at node (i,m)
t_{im}^D	departure time at node (i,m)
t_k^A	arrival time at final depot for vehicle k
t_k^D	departure time at initial depot for vehicle k
t_r	number of transfers of request r

3.4.4 Mathematical model

The model considers an objective function which has a goal of minimizing the cost of operating the fleet, equations (4)-(5), and the penalties paid because of user inconvenience given in equations (6)-(8). The operation costs given in equation (4) consist of the number of vehicles and the cost of traveling with the vehicles. The

cost of using a vehicle is included in the cost for a vehicle to travel from depot to a location (j). In addition, the variable cost associated with the time the vehicles are operating (difference between arriving at final depot and leaving initial depot) is given in equation (5).

$$\sum_{(i,m)\in\mathcal{N}}\sum_{(j,n)\in\mathcal{N}}\sum_{k\in\mathcal{K}} C_{ij}x_{imjnk} \tag{4}$$

$$+ C^U \sum_{k\in\mathcal{K}} (t_k^A - t_k^D) \tag{5}$$

Several factors of user inconvenience is considered in the model. The first part of the equation (6) considers the excess ride time, which is the difference between the real service time and the direct service time between the pickup and drop off location of each request. The next part of the equation (7), penalizes the distance the passengers have to walk during the trip. The maximum walking distance for each request is given in the details for each request. Furthermore, the number of transfers are penalized in equation (8), and is calculated in constraints (36).

$$+ P^S \sum_{r\in\mathcal{R}} ((t^A_{r+\overline{r}} - t^D_r) - T_{r,r+\overline{r}}) \tag{6}$$

$$+ P^R \sum_{(i,m)\in\mathcal{N}}\sum_{(j,n)\in\mathcal{N}}\sum_{r\in\mathcal{R}} D_{ij}v_{imjnr} \tag{7}$$

$$+ P^T \sum_{r\in\mathcal{R}} t_r \tag{8}$$

Constraints (9)-(11) state the node balance for the vehicles. Constraints (9) determine that each vehicle maximum leaves the depot once. That is, if the vehicle is used it has to leave the depot once, but if it is not used it does not have to leave. Constraints (10) verify that each vehicle that leaves the depot returns to the depot. Furthermore, constraints (11) ensure node balance for the vehicles on the remaining nodes. If a vehicle enters a node it also has to leave this node, as long as the node is not the depot for the vehicles. In addition, each node is restricted to only be visited one time by a vehicle, as given in constraints (12).

$$\sum_{(j,n)\in\mathcal{N}} x_{01jnk} \leq 1 \qquad k \in \mathcal{K} \tag{9}$$

$$\sum_{(j,n)\in\mathcal{N}} x_{jn,2\overline{r}+1,1k} - \sum_{(j,n)\in\mathcal{N}} x_{01jnk} = 0 \qquad k \in \mathcal{K} \tag{10}$$

$$\sum_{(j,n)\in\mathcal{N}} x_{jnimk} - \sum_{(j,n)\in\mathcal{N}} x_{imjnk} = 0 \qquad (i,m) \in \mathcal{N}, k \in \mathcal{K} \tag{11}$$

$$\sum_{(j,n)\in\mathcal{N}}\sum_{k\in\mathcal{K}} x_{imjnk} \leq 1 \qquad (i,m) \in \mathcal{N} \tag{12}$$

Furthermore, constraints (13)-(14) ensure that a request can only travel by a vehicle from node (i, m) to node (j, n) if a vehicle is traveling between the same set of nodes. At the same time the constraints restrict the vehicle to only visit a pickup location or drop off location if the vehicle is performing service for the request associated with the location. The vehicle can also pickup or drop off passengers which can walk to or from the location. Constraints (15)-(17) ensure node balance at the requests. Constraints (15) state that a request can only leave a node without entering it if it is its pickup location, or is a transfer location with its associated pickup location (or within walking distance of the location). Similarly, constraints (16) state that a request can only enter a node without leaving it if it is its drop off location, or is a transfer location with its associated drop off location (or within walking distance of the location). Constraints (15) - (16) also ensure that all requests are performed. Each request has to leave its pickup location either by vehicle, by the use of a fixed route or by walking. Similarly, each request has to arrive at their destination by vehicle, a fixed route or by walking. Furthermore, constraints (17) ensure that if a request enters a transfer node by a vehicle or a connected transfer node it also has to leave this node if this is not its pickup or drop off node.

$$\sum_{(j,n)\in\mathcal{N}} y_{r1jnr} - \sum_{(j,n)\in\mathcal{N}} \sum_{k\in\mathcal{K}} x_{r1jnk} = 0 \qquad r \in \mathcal{R} \tag{13}$$

$$\sum_{(j,n)\in\mathcal{N}} y_{r1jnr} - \sum_{(j,n)\in\mathcal{N}} \sum_{k\in\mathcal{K}} x_{r1jnk} = 0 \qquad r \in \mathcal{R} \tag{14}$$

$$\begin{aligned} &\sum_{(j,n)\in\mathcal{N}} (y_{i1jnr} - y_{jni1r}) \\ &+ \sum_{(j,n)\in\mathcal{N}} (v_{i1jnr} - v_{jni1r}) \\ &+ \sum_{(g(i),n)\in\mathcal{N}^G_{g(i)}} w_{i1g(i)nr} = F_{ir} \qquad (i,1) \in \mathcal{N}^P, r \in \mathcal{R} \end{aligned} \tag{15}$$

$$\begin{aligned} &\sum_{(j,n)\in\mathcal{N}} (y_{jni1r} - y_{i1jnr}) \\ &+ \sum_{(j,n)\in\mathcal{N}} (v_{jni1r} - v_{i1jnr}) \\ &+ \sum_{(g(i),n)\in\mathcal{N}^G_{g(i)}} w_{g(i)ni1r} = F_{ir} \qquad (i,1) \in \mathcal{N}^D, r \in \mathcal{R} \end{aligned} \tag{16}$$

$$\sum_{(j,n)\in\mathcal{N}} y_{imjnr} + \sum_{j\in\mathcal{G}} \sum_{(j,n)\in\mathcal{N}_j^G} w_{imjnr} + \sum_{(j,n)\in\mathcal{N}} v_{imjnr} - \sum_{(j,n)\in\mathcal{N}} v_{imjnr} - \sum_{(j,n)\in\mathcal{N}} y_{jnimr} - \sum_{j\in\mathcal{G}} \sum_{(j,n)\in\mathcal{N}_j^G} w_{jnimr} = 0 \qquad i \in G, (i,m) \in \mathcal{N}_i^G, r \in \mathcal{R} \tag{17}$$

The compatibility constraints are given by constraints (18)-(24). Constraints (18) state that in order for a vehicle to service nodes (i,m) and (j,n) in sequence, the arrival time at location j must be after the departure from location i plus the direct driving time between location i and location j. Similarly, constraints (19) illustrate the same only for fixed services. If a request travels between (i,m) and (j,n) by a fixed route the departure from node (j,n) cannot be earlier than arrival at node (i,m) plus direct travel time between the two nodes T_{ij}^F. In addition, constraints (20)-(21), specify that the departure and the arrival at the pickup and drop off locations associated with a fixed transfer location follow precedence. For example, the departure time from the pickup location have to be earlier or equal to the departure time at the associated transfer location. Furthermore, constraints (22) ensure that if the request is walking between node (i,m) and node (j,n) arrival at node (j,n) has to be after departure at node (i,m) plus direct walking time T_{ij}^R between the vehicles. In addition, constraints (23) verify precedence of the trips. That is, the pickup location has to be visited before the drop off location for each request, and the arrival at the drop off location cannot be earlier than departure at the pickup location plus direct travel time between pickup and drop off location of the requests. Constraints (24) state that the time arriving at a node has to be earlier than the time leaving the same node.

$$\sum_{k\in\mathcal{K}} x_{imjnk}(t_{im}^D + T_{ij}^K - t_{jn}^A) \leq 0 \qquad (i,m) \in \mathcal{N}, (j,n) \in \mathcal{N} \tag{18}$$

$$\sum_{r\in\mathcal{R}} w_{imjnr}(t_{im}^A + T_{ij}^F - t_{jn}^D) \leq 0 \qquad i,j \in \mathcal{G}, (i,m) \in \mathcal{N}_i^G, (j,n) \in \mathcal{N}_j^G \tag{19}$$

$$\sum_{r\in\mathcal{R}} w_{imjnr}(t_{im}^D - t_{jn}^D) \leq 0 \qquad (i,m) \in \mathcal{N}^P, j \in \mathcal{G}, (j,n) \in \mathcal{N}_j^G \tag{20}$$

$$\sum_{r\in\mathcal{R}} w_{jnimr}(t_{jn}^D - t_{im}^A) \leq 0 \qquad (i,m) \in \mathcal{N}^D, j \in \mathcal{G}, (j,n) \in \mathcal{N}_j^G \tag{21}$$

$$\sum_{r\in\mathcal{R}} v_{imjnr}(t_{im}^D + T_{ij}^R - t_{jn}^A) \leq 0 \qquad (i,m) \in \mathcal{N}, (j,n) \in \mathcal{N} \tag{22}$$

$$t_{i1}^D + T_{i,\overline{r}+i}^K - t_{\overline{r}+i,1}^A \leq 0 \qquad (i,1) \in \mathcal{N}^P \tag{23}$$

$$t_{im}^A - t_{im}^D \leq 0 \qquad (i,m) \in \mathcal{N} \tag{24}$$

The Constraints (25) set the starting time for the vehicle. That is, the departure from the initial depot has to be earlier than the arrival at the first node visited

(i,m) less the direct travel time between depot $(0,1)$ and (i,m). Similarly, the arrival time at the final depot is found in constraints (26). Constraints (27) make sure that the usage of the vehicle is below maximum usage time U. That is, the arrival at final depot less the departure at the initial depot has to be less than the usage limit. Constraints (28) ensure that the service duration for a request is less or equal to maximum service time for the request. Furthermore, constraints (29) verify that the time the node is serviced, which is equal to when the vehicle leaves the node, is within the time window.

$$x_{01imk}(t^D_k + T^K_{0i} - t^A_{im}) \leq 0 \qquad (i,m) \in \mathcal{N}, k \in \mathcal{K} \qquad (25)$$

$$x_{im,2\overline{r}+1,1k}(t^D_{im} + T^K_{i,2\overline{r}+1} - t^A_k) \leq 0 \qquad (i,m) \in \mathcal{N}, k \in \mathcal{K} \qquad (26)$$

$$t^A_k - t^D_k \leq U \qquad k \in \mathcal{K} \qquad (27)$$

$$t^A_{\overline{r}+r,1} - t^D_{r1} \leq T^K_{r,\overline{r}+r}(1+E) \qquad r \in \mathcal{R} \qquad (28)$$

$$\underline{T}_i \leq t^D_{im} \leq \overline{T}_i \qquad (i,m) \in \mathcal{N} \qquad (29)$$

Constraints (30) calculate the number of transfers per request. If the request uses the fixed route during the trip the request usually has two transfers. The passengers first transfer from a vehicle to the fixed route and after traveling with the fixed route transfer to the same or another vehicle. However, if the pickup or drop off location is associated with a transfer location, only one transfer occurs and the other has to be subtracted from the number of transfers.

$$\begin{aligned} &2 \cdot \sum_{i \in \mathcal{G}} \sum_{(i,m) \in \mathcal{N}^G_i} \sum_{j \in \mathcal{G}} \sum_{(j,n) \in \mathcal{N}^G_j} w_{imjnr} \\ &- \sum_{i \in \mathcal{G}} \sum_{(i,m) \in \mathcal{N}^G_i} \sum_{(g(i),n) \in \mathcal{N}^G_{g(i)}} w_{img(i)nr} \\ &- \sum_{i \in \mathcal{G}} \sum_{(i,m) \in \mathcal{N}^G_i} \sum_{(g(i),n) \in \mathcal{N}^G_{g(i)}} w_{g(i)nimr} \leq t_r \qquad r \in \mathcal{R} \end{aligned} \qquad (30)$$

Constraints (31) ensure that the load in the vehicle at all times is below maximum capacity. $\sum_{r \in \mathcal{R}} L_r y_{imjnr}$ calculates the combined load driving from node (i,m) to node (j,n), this number have to be zero if no vehicle travels between the two nodes, and less than the vehicle capacity Q if a vehicle travels between the nodes. Constraints (32)-(35), constraints (36), and constraints (37)-(40) set the binary, integer and continuous restrictions on the variables respectively.

$$0 \leq \sum_{r \in \mathcal{R}} L_r y_{imjnr} \leq \sum_{k \in \mathcal{K}} Q x_{imjnk} \qquad (i,m) \in \mathcal{N}, (j,n) \in \mathcal{N} \qquad (31)$$

$$x_{imjnk} \in [0,1] \qquad (i,m) \in \mathcal{N}, (j,n) \in \mathcal{N}, k \in \mathcal{K} \tag{32}$$

$$y_{imjnr} \in [0,1] \qquad (i,m) \in \mathcal{N}, (j,n) \in \mathcal{N}, r \in \mathcal{R} \tag{33}$$

$$w_{imjnr} \in [0,1] \qquad i,j \in \mathcal{G}, (i,m) \in \mathcal{N}_i^G, (j,n) \in \mathcal{N}_j^G, r \in \mathcal{R}, Z_{ij} = 1 \tag{34}$$

$$v_{imjnr} \in [0,1] \qquad (i,m) \in \mathcal{N}, (j,n) \in \mathcal{N}, r \in \mathcal{R}, D_r^R \geq D_{ij} \tag{35}$$

$$t_r \in [0,1,...] \qquad r \in \mathcal{R} \tag{36}$$

$$t_{im}^A \geq 0 \qquad (i,m) \in \mathcal{N} \tag{37}$$

$$t_{im}^D \geq 0 \qquad (i,m) \in \mathcal{N} \tag{38}$$

$$t_k^A \geq 0 \qquad k \in \mathcal{K} \tag{39}$$

$$t_k^D \geq 0 \qquad k \in \mathcal{K} \tag{40}$$

3.5 Linearizing the Model

When implementing the model it is usually easier to consider a linear model instead of the nonlinear model. Since some of the constraints in this model are nonlinear it is preferable to rewrite them in linear form when optimizing the model.

Constraints (18)-(22) can be rewritten as

$$t_{im}^D + T_{ij}^K - t_{jn}^A \leq M_{ij}^V (1 - \sum_{k \in \mathcal{K}} x_{imjnk}) \quad (i,m) \in \mathcal{N}, (j,n) \in \mathcal{N} \tag{41}$$

$$t_{im}^A + T_{ij}^F - t_{jn}^D \leq M_{ij}^F (1 - \sum_{r \in \mathcal{R}} w_{imjnr}) \quad i,j \in \mathcal{G}, (i,m) \in \mathcal{N}_i^G, (j,n) \in \mathcal{N}_j^G \tag{42}$$

$$t_{im}^D + T_{ij}^R - t_{jn}^A \leq M_{ij}^W (1 - \sum_{r \in \mathcal{R}} v_{imjnr}) \quad (i,m) \in \mathcal{N}, (j,n) \in \mathcal{N} \tag{43}$$

For instance, take constraints (41): If a vehicle travels between node (i,m) and node (j,n), that is x_{imjnk} is equal to 1, then the arrival at node (j,n) has to be equal to or greater than departure at the previous node (i,m) plus the travel time between the two nodes. On the other hand, if no vehicle travels between these two nodes the constraint has to be redundant. Here, M_{ij} is a big number which makes the constraint always hold, and different values of M_{ij} can be used as is discussed next.

A simple value for M_{ij} is the maximum value. For example, for the constraints (41) M_{ij}^V can be replaced by the maximum usage time of the vehicle U times the number of vehicles in the fleet $\mathcal{K}$. Another possibility is to replace the M_{ij}'s with the largest value connected to the node (i,m). That is,

- For constraints (41) set $M_{ij}^V = \overline{T_i} + T_{ij}^K - \underline{T_j}$, that is M_{ij}^V has to be greater than the difference of the latest time for service at location (i) plus direct service time T_{ij}^K and the earliest service time of location (j).

- For constraints (42) set $M_{ij}^F = \overline{T_i} + T_{ij}^F - \underline{T_j}$, that is M_{ij}^F has to be greater than the difference between latest service time at location (i) plus travel time T_{ij}^F and earliest service time at location (j).
- For constraints (43) set $M_{ij}^W = \overline{T_i} + T_{ij}^R - \underline{T_j}$ has to be greater than the difference between latest service time at location (i) plus travel time T_{ij}^R and earliest service time at location (j).

When replacing M with these values all constraints considering $M < 0$ are redundant and are not created in the model.

3.6 Implementing the Model

The model is written in the algebraic modeling language Mosel, implemented in Xpress IVE optimization suite and solved by Xpress Optimizer. This optimizing suite uses a simplex or interior point method for solving linear programming and uses a branch-and-bound method for solving mixed integer problems. The Xpress IVE include a pre solver which tightens the problem by removing redundant constraints and variables, adding valid inequalities to tighten the solution space before solving the problem. After the pre solver the linear relaxation of the problem is solved using simplex and using the branch-and-bound method to branch if fractional solutions occur. The solution of the linear relaxation represents the optimistic bound of the problem, while the software at the same time searches for feasible solutions which gives pessimistic bounds.

Dynamic declarations have been applied to all variables and constraints. In Xpress-IVE all dynamic arrays are created empty, and thus takes up no memory, and the variables and constraints have to be created explicitly. In this way only feasible elements, which are non-redundant, are created. This makes it possible to sum over sets, without summing over non-feasible elements, and reduce the problem size since elements which does not affect the solution are not created.

There are four binary variables in the model ($x_{imjnk}, y_{imjnr}, w_{imjnr}, v_{imjnr}$) and one integer variable (t_r). However, specifying integer requirements for the variables increase computation effort, so it is only specified integer and binary requirements on those variables where it is strictly necessary for the model. For instance, if it is specified that x_{imjnk} is binary, y_{imjnr} is binary without explicitly stating this. Furthermore, when implementing the model, since all variables on the left hand side of constraints (30) is binary, the variable t_r is integer without explicitly programming this.

4 Reduction Techniques and Valid Inequalities

The formulation of the multi-objective dial-a-ride problem that is discussed in this thesis is a mixed integer linear formulation. When this formulation is implemented in Mosel, and solved by Xpress. The optimizing suite removes the integer requirement on the variables, and solves the remaining LP problem. This LP problem is referred to as a LP relaxation of the integer programming problem (IP problem). Since integer problems are not defined on continuous space it is much more difficult to solve, especially for such large-scale routing problems. While the solution space for the linear problem is convex, the solution space for the integer point represents specific points in the linear solution space, where the neighborhood of points is non-integer. Thus, when solving the relaxed problem instead of the integer problem fractional solutions can occur. In the case of a fractional solution the optimizing suite branch on the variables. The branch-and-bound approach makes it possible to divide the solution space into two disjunct solution spaces.

A common solution method for integer programming problems is to develop an iterative method that establishes both optimistic and pessimistic bounds on the optimal value of the problem, and the software packaged used to solve this problem works in this way. Since the problem under consideration is a minimization problem the optimistic bound is referred to as lower bound and is represented by the linear relaxation. The pessimistic and upper bound represents a feasible solution to the mixed integer problem. Then, when the difference between the upper and lower bound is within an allowable limit, the optimal solution is found. Thus, it is favorable to structure the problem so the lower bound is as close as possible to the upper bound, so as to reduce number of iterations and solution time. Furthermore, valid inequalities are added to the problem to cut away some part of the solution problem which is not feasible for the integer problem. However, adding these additional constraints increases the complexity and makes it more difficult to solve. There is thus a trade-off between complexity and reducing the solution space.

In Figure 7 the black lines create a linear relaxation solution space. The black circles represent integer feasible solutions, and the dashed lines represent the convex hull. The goal is to represent the problem so that the solution space of the relaxed problem is as equal as possible to the convex hull, because when the solution space is a convex hull solving the LP relaxation yields the optimal integer feasible solution. Thus, when the solution space represents the convex hull better, the branch-and-bound algorithm has a smaller space to search and needs to search through fewer subproblems, or nodes, to find the optimal solution. To strengthen the formulation valid inequalities are added to the formulation. The inequalities are valid if it cuts away some part of the LP solution space without cutting away any of the integer feasible solutions. In the figure, the blue and the red line represent valid inequalities. If the valid inequality represents a surface of the convex hull, like the blue line, it is called a facet and it is the strongest valid inequality that can be found. By including valid inequalities in the formulation and thus represent the convex hull better, the Xpress Optimization Suite manages to solve the problem faster.

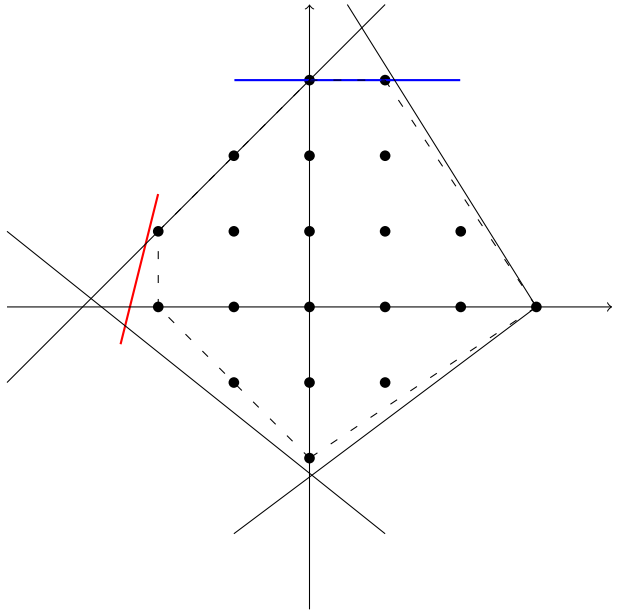

Figure 7: Solution space

In this section several groups of valid inequalities are discussed. First, the arc elimination constraints are discussed in Section 4.1, these arc eliminations are added to the problem so as not to add those arcs which cannot be used. Next, the symmetry breaking constraints in Section 4.2 explain ways to reduce the number of symmetric solutions. Then the subtour elimination constraints are discussed in Section 4.3, and the mixed integer rounding constraints, in Section 4.4.

4.1 Arc Elimination Rules

Since IDARP is NP-hard and grows exponentially with the number of requests, keeping the solution space as small as possible is beneficial for solving the problem as efficient as possible. For this reason, several arc elimination rules have been proposed to minimize the solution space. The model presented in this thesis has used several of the arc elimination rules discussed in Häll et al. (2009), in addition to some custom-made arc elimination constraints for the formulation. The arc elimination constraints are implemented so that the arc in question is not considered (instead of creating the arc and forcing it to be non-active). This is done by dynamically creating only those variables and constraints that can be used in the mixed integer linear formulation.

x_{imjnk} **- 1 if vehicle** k **travels from** (i, m) **to** (j, n)**:**
The variable x_{imjnk} specifies if a vehicle k travels from (i, m) to (j, n), however not all possible arcs represents feasible travel routes for the vehicle. First of all in the

formulation given in Section 3.4 only one node needs to be created for every pickup location, drop off location, or initial or final depot. As a consequence, is cannot go any arc to or from those with greater visitation number than 1. Furthermore, the only valid arcs going out from the initial depot are to pickup and transfer locations. That is, there is no point going from the initial depot to the initial depot or the final depot. Furthermore, there is no point for a vehicle to go from initial depot to a drop off location without picking up the request, that is $(0, 1, \overline{r} + i, 1)$ for $(i, 1) \in \mathcal{N}^{\mathcal{P}}$ are not allowed. In the same way, when entering the final depot, the vehicle can only come from a transfer or drop off location, as the picked up request also has to be delivered. That is, $(i, 1, 2\overline{r} + 1, 1)$ for $(i, 1) \in \mathcal{N}^{\mathcal{P}}$ are not feasible. Furthermore, no arc can go from any node into the initial depot (i.e. $(i, m, 0, 1)$ is infeasible for all $(i, m) \in \mathcal{N}$), or from the final depot to any node (ie. $(2\overline{r}+1, 1, i, m)$ for all $(i, m) \in \mathcal{N}$ are infeasible).

From a pickup location i a vehicle can go to a pickup location j where the combined load does not exceed the vehicle capacity ($L_i + L_j \leq Q$). A vehicle cannot go between the same pickup location (that is, (i, m, i, m) for $(i, m) \in \mathcal{N}$ are infeasible). Furthermore, a vehicle can go from a pickup location to transfer and drop off locations. There are vehicle arcs between all transfer nodes, except between nodes at the same location. Additionally, from a drop off location a vehicle can travel to non-associated pickup locations (i.e. $(\overline{r} + i, 1, i, 1)$ where $(i, 1) \in \mathcal{N}^{\mathcal{P}}$ are infeasible), transfer locations and other drop off locations.

In addition, if symmetry breaking constraints considering the order of nodes (Section 4.2) is active, a vehicle traveling between nodes in the same transfer location can only travel from a node with a lower index to one with a higher index.

Furthermore, arcs can be eliminated taken into account the time windows (Häll et al., 2009). For instance, there is no point to add arcs which connect one location i to a location j if a vehicle cannot reach location j from location i within the time window. That is, if $(\underline{T}_i + T^K_{ij} > \overline{T}_j)$ then arc (i, j) is infeasible. This can be written:

$$\sum_{m \in \mathcal{N}_i} \sum_{n \in \mathcal{N}_j} \sum_{k \in \mathcal{K}} x_{imjnk} (\underline{T}_i + T^K_{ij} - \overline{T}_j) \leq 0 \quad i \in \mathcal{P} \cup \mathcal{D} \cup \mathcal{G}, j \in \mathcal{P} \cup \mathcal{D} \cup \mathcal{G} \quad (44)$$

This inequality can in principle be strengthened by replacing the beginning of the time window at location i to the departure from the node (which is equal to service start) t^D_{im}, but will then get a constraint for each node and not each location. That is, if $(t^D_{im} + T^K_{ij} > \overline{T}_j)$ then arcs going from node (i, m) to location j is infeasible. Written in full this gives:

$$\sum_{n \in \mathcal{N}_j} \sum_{k \in \mathcal{K}} x_{imjnk} (t^D_{im} + T^K_{ij} - \overline{T}_j) \leq 0 \qquad (i, m) \in \mathcal{N}, j \in \mathcal{P} \cup \mathcal{D} \cup \mathcal{G} \qquad (45)$$

In the model, the constraints (44) is implemented. In addition, if visiting location j after picking up a request at i there has to be enough time to get back to

drop of location $(\overline{r}+i)$. That is, if $(T_{ij}+T_{j,\overline{r}+i} > T_{i,\overline{r}+i}(1+E))$ then arc (i,j) and arc $(j,\overline{r}+i)$ are infeasible. This can be written as:

$$\sum_{n\in\mathcal{N}_j}\sum_{k\in\mathcal{K}} x_{i1jnk}(T^K_{ij}+T^K_{j,\overline{r}+i}-T^K_{i,\overline{r}+i}(1+E)) \leq 0 \quad i\in\mathcal{P}, j\in\mathcal{P}\cup\mathcal{D}\cup\mathcal{G} \tag{46}$$

$$\sum_{n\in\mathcal{N}_j}\sum_{k\in\mathcal{K}} x_{jn,\overline{r}+i,1k} - \sum_{n\in\mathcal{N}_j}\sum_{k\in\mathcal{K}} x_{i1jnk} \leq 0 \quad i\in\mathcal{P}, j\in\mathcal{P}\cup\mathcal{D}\cup\mathcal{G} \tag{47}$$

These constraints are valid also including the possibility of walking when it is assumed that vehicles are the fastest mode of transport. Similarly, considering the capacity limits on a vehicle and the load to be picked up or delivered at the locations, arcs can be eliminated. That is, if the combined loads exceed vehicle capacity, the vehicle cannot go between these two locations.

$$\sum_{k\in\mathcal{K}} x_{i1j1k}(L_i+L_j-Q) \leq 0 \qquad (i,1)\in\mathcal{N}^P, (j,1)\in\mathcal{N}^P \tag{48}$$

$$\sum_{k\in\mathcal{K}} x_{i1j1k}(L_i+L_j-Q) \leq 0 \qquad (i,1)\in\mathcal{N}^D, (j,1)\in\mathcal{N}^D \tag{49}$$

Bound on the Time Windows:
In addition, bounds on time variable can be strengthened, for example as was suggested by Desrochers and Laporte (Ropke et al., 2007):

$$\underline{T}_{im} + \sum_{j\in\mathcal{N}\setminus(i)}\sum_{k\in\mathcal{K}} max(0,\underline{T}_{jn}-\underline{T}_{im}+T^K_{ij})x_{jnimk} \leq t^D_{im} \quad (i,m)\in\mathcal{N},(j,n)\in\mathcal{N} \tag{50}$$

$$t^D_{im} \leq \overline{T}_{im} - \sum_{j\in\mathcal{N}\setminus(i)}\sum_{k\in\mathcal{K}} max(0,\overline{T}_{im}-\overline{T}_{jn}+T^K_{ij})x_{imjnk} \quad (i,m)\in\mathcal{N},(j,n)\in\mathcal{N} \tag{51}$$

Constraints (50) state that service time at the location is later than the latest of the earliest service time at the node and the first possible time the vehicle traveling to the location reaches the location. Similarly, constraints (51) state that the service time is earlier than the earliest of the latest service at the location and the latest time the vehicle visiting the location has to leave to be able to visit its next location before its latest service time.

y_{imjnr} **- 1 if request** r **travels from** (i,m) **to** (j,n) **by vehicle:**
The variable y_{imjnr} which specifies if the request travels by vehicle between (i,m) and (j,n) and needs only to be created if the variable x_{imjnk} is created. Furthermore, the request is not allowed to travel to or from the depot, and each request cannot travel into its pickup location or from its drop off location.

Furthermore, if the combined load from request i and request j exceeds the capacity of a vehicle, the request i is not allowed to travel in or out from pickup or drop off location of j and in to pickup location j if the request i then can walk to its drop off node or a fixed network. The same hold for request j and the locations associated with request i.

w_{imjnr} - 1 if request r travels from (i,m) to (j,n) using a fixed network: The variable w_{imjnr} specifies if the request travels using a fixed network between (i,m) and (j,n), and needs only to be created if there exists a fixed network which makes it possible to reach location (j,n) from (i,m). In addition, variables from and to their pickup location to an associated transfer location should be created. However, a variable from another pickup and drop off location associated with the transfer location needs not be created.

Furthermore, if the request does not travel by any mode (vehicle, fixed route, walks) into a transfer location, it cannot travel by a fixed route from this transfer location. If the trip of a request traveling from its pickup location to a transfer location to its drop off problem exceeds maximum service time for the requests then the request cannot use the fixed network from this location. Furthermore, if the trip of a request traveling from its pickup location to a transfer location to another transfer location to the drop off location exceeds the maximum service time then the request cannot travel between the transfer locations.

v_{imjnr} - 1 if request r walks from (i,m) to (j,n):
The variable for walking between locations, v_{imjnr}, needs only be created when it is in theory possible for a request to go by vehicle between two locations and the distance is less than the maximum walking distance.

$$v_{imjnr}(D_{ij} - D_r) \leq 0 \qquad r \in \mathcal{R}, (i,m) \in \mathcal{N}, (j,n) \in \mathcal{N} \tag{52}$$

4.2 Symmetry Breaking Constraints

During the solution process of the problem several groups of symmetric solutions can occur, and thus find solutions that look different on paper, but in practical terms are equal. For instance, vehicle k can visit location i by visiting all the artificial nodes attached to the location, but if we look at the practical use of the system the solutions are equal. Equally, for the practical use of the model it does not matter if request r is fulfilled by vehicle k or $k+2$ because the vehicle fleet is homogeneous. To shorten the solution time of the problem symmetry breaking constraints can be created to force the system to not search for symmetric solutions. Below some examples of possible symmetry breaking constraints are presented, and constraints (53) to constraints (56) are implemented in the model:

$$t^A_{im} - t^A_{i,m+1} \leq 0 \qquad (i,m) \in \mathcal{N} \setminus (i, M_i) \tag{53}$$

$$t^D_k - t^D_{k+1} \leq 0 \qquad k \in \mathcal{K} \setminus |K| \tag{54}$$

$$(t^A_k - t^D_k) - (t^A_{k+1} - t^D_{k+1}) \leq 0 \qquad k \in \mathcal{K} \setminus |K| \tag{55}$$

$$\sum_{(i,m)\in\mathcal{N}} \sum_{(j,n)\in\mathcal{N}} D_{ij} x_{imjnk} - \sum_{(i,m)\in\mathcal{N}} \sum_{(j,n)\in\mathcal{N}} D_{ij} x_{imjn,k-1} \leq 0 \quad k \in \mathcal{K} \setminus |K| \tag{56}$$

$$\sum_{(i,m)\in\mathcal{N}} \sum_{(j,n)\in\mathcal{N}} T^K_{ij} x_{imjnk} - \sum_{(i,m)\in\mathcal{N}} \sum_{(j,n)\in\mathcal{N}} T^K_{ij} x_{imjn,k-1} \leq 0 \quad k \in \mathcal{K} \setminus |K| \tag{57}$$

$$\sum_{(i,m)\in\mathcal{N}} \sum_{(j,n)\in\mathcal{N}} C_{ij} x_{imjnk} - \sum_{(i,m)\in\mathcal{N}} \sum_{(j,n)\in\mathcal{N}} C_{ij} x_{imjn,k-1} \leq 0 \quad k \in \mathcal{K} \setminus |K| \tag{58}$$

$$\sum_{(i,m)\in\mathcal{N}} \sum_{(j,n)\in\mathcal{N}} x_{imjnk} - \sum_{(i,m)\in\mathcal{N}} \sum_{(j,n)\in\mathcal{N}} x_{imjn,k-1} \leq 0 \quad k \in \mathcal{K} \setminus |K| \tag{59}$$

Constraints (53) force the arrival at node (i, m) to be earlier than the arrival at node $(i, m+1)$, which in practical terms means that the nodes corresponding to a location i has to be visited in sequence. As before, M_i represents the maximum number of visits at the location. Figure 8 shows a location i with three nodes. Furthermore, it shows that a vehicle k has to visit node 1 before it visits node 2. Constraints (54) state that the time vehicle k leaves depot has to be earlier than the time vehicle $k+1$ leaves depot, that is vehicle k has to be used before vehicle $k+1$. Constraints (55) ensure that the time vehicle k is away from depot has to be longer than the time vehicle $k+1$ has to be away from depot. Constraints (56)-(59) say that the route vehicle k travels has to have a greater distance travelled, spend longer time driving, be more expensive, or travels on more of the arcs, than the route of the vehicle $k+1$. These constraints break two kinds of symmetries. Firstly, the constraints ensure that the vehicles are used in sequence. Secondly, sequencing the vehicles after highest route costs ensure that given a number of routes and vehicles there is only one valid combination. Figure 9 shows a small network and a long route for vehicle 1 and a short route for vehicle 2.

Constraints (53) can be used together with all the other symmetry breaking constraints without creating conflicts. The resulting constraints (54)-(59) might cause conflicts if used together. First, consider constraints (56)-(59). These are similar in structure, only constraints (56) considers total distance traveled, constraints (57) considers the total time the vehicle use driving, constraints (58) considers the total cost of traveling and (59) considers the number of arcs traveled. Similarly, constraints (55) considers the total time the each vehicle is away from depot, while constraints (54) sets that the start of the first vehicles have to be earlier than the later vehicles. When using two or more of constraints (55)-(59) together problems can occur since different variants might force different vehicles to be used first, second or in some other order. In the implementation of this model the parameters D_{ij}, T^K_{ij} and C_{ij} are proportional and should thus provide the similar results. During the result and discussion Section 6.1 the different combinations of the symmetry

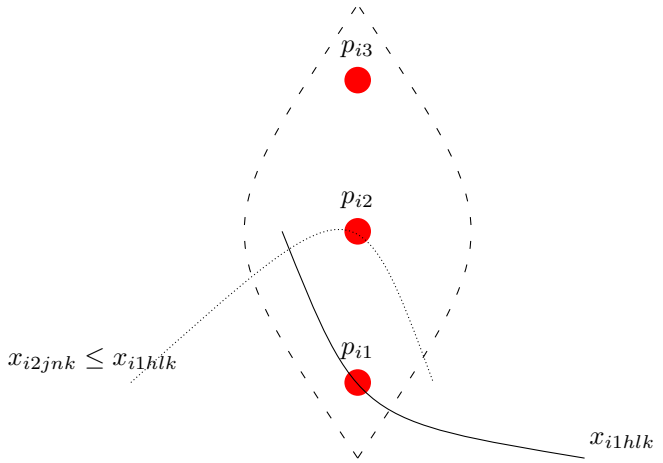

Figure 8: Symmetry within a Location

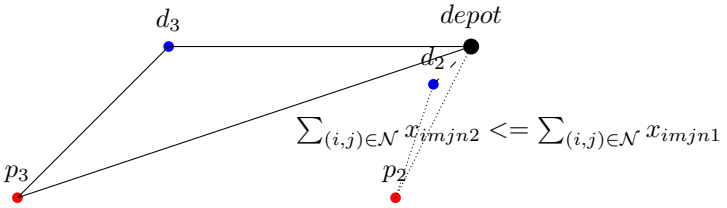

Figure 9: Symmetry in the Fleet

breaking constraints are tested. However, one complication using constraints (54) and constraints (55) forces the model to create all starting times (arrival times) for the vehicles.

In addition, a set of constraints that restricts the vehicle to visit transfer nodes only when they are dropping off or picking up passengers are included in the formulation. These constraints help the model to avoid vehicles to travel between nodes which there are no cost of visiting, and are given as below in constraints (60). Since only one vehicle can visit each node it is possible to aggregate the constraints for all vehicles. These constraints are illustrated in Figure 10, where the dotted lines represents a likely route without constraints (60) while the solid line represents the route with the constraints (60).

$$\sum_{r\in\mathcal{R}} \sum_{(j,n)\in\mathcal{N}} (w_{jngmr} + v_{jngmr} + w_{gmjnr} + v_{gmjnr}) \geq \sum_{k\in\mathcal{K}} \sum_{(j,n)\in\mathcal{N}} x_{jngmk} \quad (g,n) \in \mathcal{N}^G \tag{60}$$

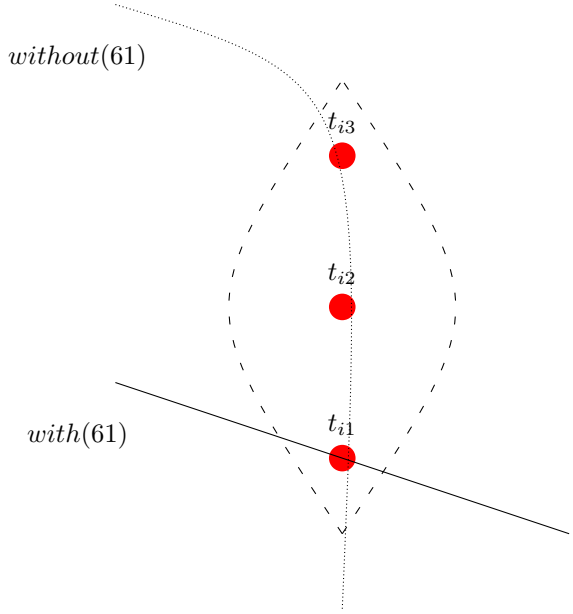

Figure 10: Avoiding unnecessary routes

4.3 Subtour Elimination Constraints

In the linear relaxation of the problem the variables which are 1 if a vehicle or a request travels from location i to location j, or 0 otherwise, might be fractional. For example, $x_{imjnk} = 0.3$ means that 0.3 of the vehicle k is supposed to travel between location i and location j, but intuitively this is not feasible in practical cases. These fractional solutions make it possible for subtours to be created in the linear problem and thus lowering the lower bound while satisfying the problem constraints. When restricting these subtours in the linear formulation the resulting problem is stronger.

Dantzig, Fulkerson and Johnson (Applegate et al., 2006) proposed the subtour elimination constraints (SEC) (61)-(62) for the traveling salesman problem. Here e is an arc, S is a subset of nodes N. Then $E(S)$ is all arcs in the subset S, while $\delta(S)$ is all arcs where one end of the arc is in the subset S and one arc is not in the subset $N \setminus S$. The constraints works by eliminating solutions consisting of two or more disjoint subtours: given the set of integer solutions of the LP problem and eliminate those that are disconnected leaving only the incidence vectors of tours. However, since there are too many possible subtours to add them all to the LP relaxation, Dantzig, Fulkerson and Johnson added the inequalities in an iterative way by attacking the subtours as they occur.

$$\sum_{e \in \delta(S)} x_e \geq 2 \tag{61}$$

$$\sum_{e \in E(S)} x_e \leq |S| - 1 \tag{62}$$

One possible lifting of the constraint is to incorporate the capacity restriction into the inequality. That is, since the capacity of the vehicle restricts the number of nodes in a subset that is possible to visit, this information can be used to strengthen the subtour elimination constraints.

$$\sum_{e \in E(S)} x_e \leq |S| - max\left(1, \frac{\sum_{r \in S} L_r}{Q}\right) \tag{63}$$

Several other lifted formulations have been proposed in the literature. Firstly, Dumitrescu et al. (2006) (Golden et al., 2008) proposed a lifted subtour elimination constraints for the single vehicle pickup and delivery problem, which are given as follows. Let $S \subseteq \mathcal{P} \cup \mathcal{D}$ be such that there exists $i \in \mathcal{P}$ such that $i \in S, n+i \in S$. Then, $x(S)$ represents the x's in the subset S, and the Inequality (64) is valid.

$$x(S) + \sum_{j \in P \cap S, n+j \in \overline{S}} x_{i,n+j} \leq |S| - 1 \tag{64}$$

Dumitrescu et al. (2006) also proved the generalization of the lifted subtour elimination constraint (Constraint (64)). This is given by letting $S \subset \mathcal{P} \cup \mathcal{D}$ be such that there exists $i \in \mathcal{P} \cap S$ with $n+i \in S$. Let $T_k \subset \mathcal{P} \cup \mathcal{D}$, $k = 1, .., p$ be p sets such that there exists $i_k \in \mathcal{P} \cap \mathcal{D}$ and $n + i_k \in T_k, T_k \cap S = i$ for $k = 1, ..., p$ and $T_j \cap T_k = i$ for all $j, k = 1, ..., p$, $j \neq k$. Then the Inequality (65) is valid.

$$x(S) + \sum_{k=1}^{p} x(T_k) \leq |S| - 1 + \sum_{k=1}^{p} (|T_k| - 2) \tag{65}$$

Another possible lifting of the subtour elimination constraints are the ones presented in Cordeau and Iori M. (2010) and introduced by Balas et al. (1995). Balas et al. (1995) took advantage of the fact that in the dial-a-ride problem, each node $i \in P \cup D$ is either the predecessor or the successor of exactly one other node. Then for any set $S \subseteq \mathcal{P} \cup \mathcal{D}$, let $\pi(S) = \{i \in \mathcal{P} | n+i \in S\}$ and $\sigma(S) = \{n+i \in \mathcal{D} | i \in S\}$ denotes the sets of predecessors and successors of S, respectively. The resulting inequalities were called σ inequalities and π inequalities.

$$x(S) + \sum_{i \in \overline{S} \cap \sigma(S)} \sum_{j \in S} x_{ij} + \sum_{i \in \overline{S} \setminus \sigma(S)} \sum_{j \in S \cap \sigma(S)} x_{ij} \leq |S| - 1 \qquad S \subseteq \mathcal{P} \cup \mathcal{D} \qquad (66)$$

$$x(S) + \sum_{i \in S} \sum_{j \in \overline{S} \cap \pi(S)} x_{ij} + \sum_{i \in S \cap \pi(S)} \sum_{j \in \overline{S} \setminus \pi(S)} x_{ij} \leq |S| - 1 \qquad S \subseteq \mathcal{P} \cup \mathcal{D} \qquad (67)$$

In addition, the ordinary subtour elimination constraints (62) can also be lifted by taken into account the orientation of the arcs, like Grötschel and Padberg (1985) (Cordeau and Iori M., 2010) did and proposed the D_k^+ and D_k^- inequalities for the asymmetric TSP:

$$\sum_{j=1}^{h-1} x_{i_j i_{j+1}} + x_{i_h i_1} + 2 \sum_{j=2}^{h-1} x_{i_j i_1} + \sum_{j=3}^{h-1} \sum_{l=2}^{j-1} x_{i_j i_l} + \sum_{n+i_p \in \overline{S} \cap \sigma(S)} x_{n+i_p i_1} \leq h - 1 \quad S = \{i_1, ..., i_h\} \subseteq N, h \geq 3 \qquad (68)$$

$$\sum_{j=1}^{h-1} x_{i_j i_{j+1}} + x_{i_h i_1} + 2 \sum_{j=3}^{h} x_{i_1 i_j} + \sum_{j=4}^{h} \sum_{l=3}^{j-1} x_{i_j i_l} + \sum_{i_p \in \overline{S} \cap \pi(S)} x_{i_1 i_p} \leq h - 1 \quad S = \{i_1, ..., i_h\} \subseteq N, h \geq 3 \qquad (69)$$

Used in the IDARP formulation:
Three groups of subtour elimination constraints are added to the formulation. One, two or three of the groups can be added to the formulation, and the effectiveness of the groups is discussed in the Section 6.1. The first group consists of a simple inequalities restricting the subtour between two nodes, and taken advantage of the fact that only one vehicle can travel between a set of nodes (equations (70)). Furthermore, it restricts the vehicle to only travel between any set of locations if the vehicle travels from depot to any node, as given in equations (71).

$$\sum_{k \in \mathcal{K}} x_{imjnk} + \sum_{k \in \mathcal{K}} x_{jnimk} \leq 1 \qquad (i, m) \in \mathcal{N}, (j, n) \in \mathcal{N} \qquad (70)$$

$$\sum_{(j,n) \in \mathcal{N}} x_{i1jnk} - \sum_{(j,n) \in \mathcal{N}} x_{01jnk} \leq 0 \qquad (i, 1) \in \mathcal{N}^P \cup \mathcal{N}^D, k \in \mathcal{K} \qquad (71)$$

The second group consists of iteratively added subtour elimination constraints as described by Dantzig, Fulkerson and Johnson (Applegate et al., 2006). The class of constraints is strengthened by aggregating the vehicles since only one vehicle can travel between a pair of nodes. Furthermore, the constraints are strengthened by

considering the capacity restrictions on the vehicles. The resulting class of constraints are shown in constraints (72).

$$\sum_{k\in\mathcal{K}}\sum_{(i,m)\in\mathcal{S}}\sum_{(j,n)\in\mathcal{S}} x_{imjnk} \leq |\mathcal{S}| - max\{1, \frac{\sum_{r\in\mathcal{S}} L_r}{Q}\} \quad \mathcal{S}\subseteq\mathcal{N}\setminus\{0, 2\overline{r}+1\}, 2\leq|\mathcal{S}|\leq N \tag{72}$$

The third group of subtour elimination constraints considers a set of lifted subtour elimination constraints. First, out of the two lifted subtour elimination constraints (66)-(67), six set of valid inequalities are made, considering that for each request r, node $(r,1)$ has to be visited before node $(\overline{r}+r,1)$. Figure 11 to Figure 16 shows the network for constraints (73)-(78), and for the four first constraints (73)-(76) at most one arc can be used, while at most two arcs can be used in constraints (77)-(78).

$$\sum_{k\in\mathcal{K}} x_{i1j1k} + \sum_{k\in\mathcal{K}} x_{j1i1k} + \sum_{k\in\mathcal{K}} x_{\overline{r}+i,1,j1k} + \sum_{k\in\mathcal{K}} x_{\overline{r}+j,1,i1k} \leq 1 \qquad (i,1)\in\mathcal{N}^P, (j,1)\in\mathcal{N}^P \tag{73}$$

$$\sum_{k\in\mathcal{K}} x_{\overline{r}+i,1,\overline{r}+j,1k} + \sum_{k\in\mathcal{K}} x_{\overline{r}+j,1,\overline{r}+i,1k} + \sum_{k\in\mathcal{K}} x_{\overline{r}+i,1,j1k} + \sum_{k\in\mathcal{K}} x_{\overline{r}+j,1,i1k} \leq 1 \qquad (i,1)\in\mathcal{N}^P, (j,1)\in\mathcal{N}^P \tag{74}$$

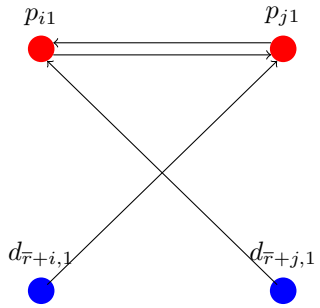

Figure 11: Subtour Elimination Constraints 73

$$\sum_{k\in\mathcal{K}} x_{i1,\overline{r}+j,1k} + \sum_{k\in\mathcal{K}} x_{\overline{r}+j,1i1k} + \sum_{k\in\mathcal{K}} x_{\overline{r}+i,1,\overline{r}+j,1k} \leq 1 \quad (i,1)\in\mathcal{N}^P, (j,1)\in\mathcal{N}^P \tag{75}$$

$$\sum_{k\in\mathcal{K}} x_{i1,\overline{r}+j,1k} + \sum_{k\in\mathcal{K}} x_{\overline{r}+j,1i1k} + \sum_{k\in\mathcal{K}} x_{i1j1k} \leq 1 \quad (i,1)\in\mathcal{N}^P, (j,1)\in\mathcal{N}^P \tag{76}$$

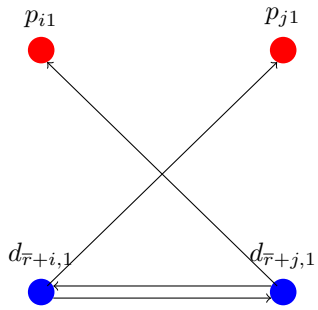

Figure 12: Subtour Elimination Constraints 74

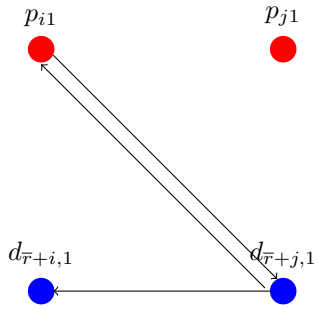

Figure 13: Subtour Elimination Constraints 75

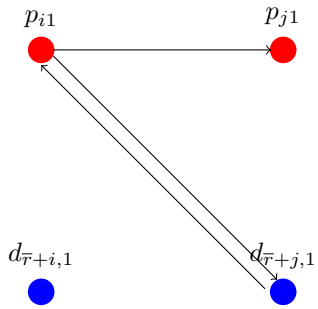

Figure 14: Subtour Elimination Constraints 76

$$\sum_{k\in\mathcal{K}} x_{i1j1k} + \sum_{k\in\mathcal{K}} x_{j1i1k} + \sum_{k\in\mathcal{K}} x_{i1,\overline{r}+i,1k} + \sum_{k\in\mathcal{K}} x_{j1,\overline{r}+i,1k} + \sum_{k\in\mathcal{K}} x_{\overline{r}+i,1j1k} + \sum_{k\in\mathcal{K}} x_{\overline{r}+j,1i1k} + \sum_{k\in\mathcal{K}} x_{\overline{r}+j,1,\overline{r}+i,1k} \leq 2 \quad (i,1)\in\mathcal{N}^P, (j,1)\in\mathcal{N}^P \quad (77)$$

$$\sum_{k\in\mathcal{K}} x_{i1,\overline{r}+j,1k} + \sum_{k\in\mathcal{K}} x_{\overline{r}+j,1i1k} + \sum_{k\in\mathcal{K}} x_{i1,\overline{r}+i,1k} + \sum_{k\in\mathcal{K}} x_{\overline{r}+j,1,\overline{r}+i,1k} + \sum_{k\in\mathcal{K}} x_{\overline{r}+i,1,\overline{r}+j,1k} + \sum_{k\in\mathcal{K}} x_{i1j1k} + \sum_{k\in\mathcal{K}} x_{\overline{r}+i,1j1k} \leq 2 \quad (i,1)\in\mathcal{N}^P, (j,1)\in\mathcal{N}^P \quad (78)$$

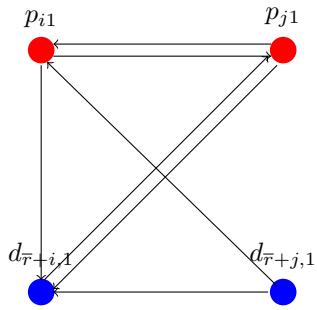

Figure 15: Subtour Elimination Constraints 77

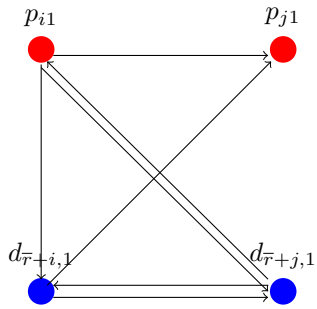

Figure 16: Subtour Elimination Constraints 78

Furthermore, the third group include two classes of the D_k^+ and D_k^- Inequalities (68)-(69). Figures of the networks and arcs in the subset are illustrated in the representative networks; here as well two of the arcs can be used in the same solution.

$$\sum_{k\in\mathcal{K}} x_{\overline{r}+i,1j1k} + \sum_{k\in\mathcal{K}} 2x_{j1,\overline{r}+i,1k} + \sum_{k\in\mathcal{K}} x_{j1i1k} + \sum_{k\in\mathcal{K}} x_{i1,\overline{r}+i,1k} + \sum_{k\in\mathcal{K}} x_{\overline{r}+j,1,\overline{r}+i,1k} \leq 2 \quad (i,1)\in\mathcal{N}^P, (j,1)\in\mathcal{N}^P \tag{79}$$

$$\sum_{k\in\mathcal{K}} x_{i1,\overline{r}+i,1k} + \sum_{k\in\mathcal{K}} x_{\overline{r}+i,1,\overline{r}+j,1k} + \sum_{k\in\mathcal{K}} x_{\overline{r}+j,1i1k} + \sum_{k\in\mathcal{K}} 2x_{i1,\overline{r}+j,1k} + \sum_{k\in\mathcal{K}} x_{i1j1k} \leq 2 \quad (i,1)\in\mathcal{N}^P, (j,1)\in\mathcal{N}^P \tag{80}$$

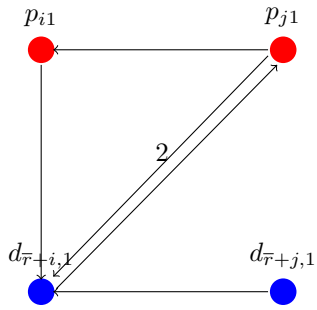

Figure 17: Subtour Elimination Constraints 79

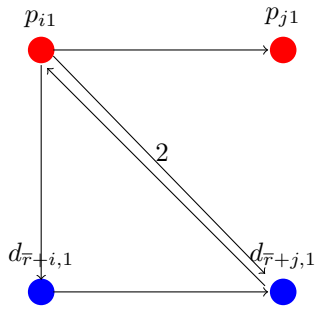

Figure 18: Subtour Elimination Constraints 80

4.4 Mixed Integer Rounding Capacity Constraints

Laporte, Nobert and Desrochers (1985) and Cordeau (2006) (Golden et al., 2008) have discussed the use of mixed integer rounding (MIR) capacity constraints in the dial-a-ride problem, and found that the standard vehicle routing problem capacity constraints (inequality (81)) where valid for the dial-a-ride problem. Here,

Q is the capacity of the vehicle, while $q(S)$ is the load in the subset to be picked up.

$$x(\delta(S)) \geq 2\lceil \frac{q(S)}{Q} \rceil \qquad S \subseteq \mathcal{P} \cup \mathcal{D} \tag{81}$$

Used in the IDARP formulation:
In the implementation of this model the following mixed integer rounding constraints are considered. First of all, for subsets of pickup locations or subsets of drop off locations, which are not connected to a fixed network or are not within walking distance to another node in the network, the free capacity going in to the subset has to be greater than the load to be picked up in the subset. The resulting equation (82) is given below, and here, y_e is the flow going into the subset. Furthermore, $\delta(S)$ represents the set of arcs going into the subset S.

$$\sum_{e \in \delta(S)} (Qx_e - y_e) \geq \sum_{i \in S} L_i \tag{82}$$

For subsets S which consists of only one pickup location or one drop off location one vehicle has to visit this location. Furthermore, subsets S consisting of two pickup locations or two drop off locations with the combined load to be picked up or dropped off exceeding the vehicle capacity the subset has to be visited twice. Similar, other valid inequalities can be computed for bigger subsets S, and in general the constraint can be written as.

$$\sum_{e \in \delta(S)} x_e \geq \lceil \frac{\sum_{i \in S} L_i}{Q} \rceil \tag{83}$$

Another version of this valid inequality also considers the flow going into the subset. For example, consider the subset S consisting of two pickup locations i and j not associated with a fixed network or not within walking distance off another node.

$$\sum_{e \in \delta(S)} x_e \geq \lceil \frac{\sum_{i \in S} L_i}{Q} \rceil + \frac{\sum_{e \in \delta(S)} L_r y_e}{Q} \tag{84}$$

A further extension to be considered is subsets containing pickup locations (drop off locations) also including locations where a request can go to another location within the subset. For these subsets, the inequalities given above are also valid.

The last mixed integer rounding constraint considers the minimum number of vehicles that has to leave the depot in the optimal solution. Since it is possible to

find fractional solutions in the linear problem, the LP relaxation might provide a solution where only 0.5 of a vehicle is used. For example, if request 1 has a load of 2, 0.5 of the vehicle with capacity 4 has to use the arc. When less of the vehicle is occupied by traveling on one specific arc, the rest of the vehicle can transport other requests at the same time on other arcs. However, this is not a possible IP solution. As a consequence, the LP solution might need less vehicles to manage to fulfill all requests than the IP solution. Thus, by trying to find areas where at least n vehicles has to leave the depot reduces the gap between the LP and IP solution. To do this the implementation searches different sets of pickup locations to see if it is possible to travel from depot to the requests and back to depot without restricting either the time windows for the requests or for the vehicle. In addition, the program search between all set of two pickup location to check if there is two requests which cannot be fulfilled by the same vehicle considering both time windows for requests and the vehicle and at the same time considering loads to be picked up and delivered and the vehicle capacity. That is, to find this subset, the model searches through two and two pickup locations including also their drop off locations in the subset. None of the locations can be associated with a transfer location, and it cannot be possible to walk to a node not in the subset. Then, if a vehicle does not manage to visit all locations considering load or usage capacity of the vehicle, time windows of the service or maximum service time, then it is evident that at least two vehicles have to leave the depot. Thus, it is possible to force at least two vehicles to leave the depot in the LP solution, because at least two vehicles have to be used in a feasible IP solution.

In addition, to the strengthening factors mentioned in the following subsections to additional ways of reducing the gap between the LP solution and IP solution is given below (these are not included in any options but are activated for all combinations of options.) Firstly, in the LP solution the usage of the vehicle can be reduced if only fractions of the vehicle is used by manipulating the time of start and end of each vehicle. By forcing the times to be more equal to the times in the IP solution the gap is reduced. This restriction is given in the constraints (85).

$$t_k^D + \sum_{(i,m)\in\mathcal{N}} \sum_{(j,n)\in\mathcal{N}} T_{ij}^K x_{imjnk} - t_k^A \leq 0 \qquad k \in \mathcal{K} \tag{85}$$

Similarly, in the LP solution fractional parts of the vehicles can be used, while in the IP solution this of course is not possible. Since a fixed cost occur for each vehicle leaving the depot, the sum of the fractions of vehicles leaving are forced as small as possible in the LP solution. Thus, to reduce the gap between the LP and IP solution the fraction leaving the depot can be forced to exceed the fraction traveling between the rests of the locations. These equations are given constraints (86).

$$\sum_{(j,n)\in\mathcal{N}} x_{imjnk} \leq \sum_{(j,n)\in\mathcal{N}} x_{01jnk} \qquad (i,m) \in \mathcal{N} \tag{86}$$

5 An Example

This section illustrates an example instance, provides the details and structure of the data set, how this data are transformed into a practical problem, and discuss the solution found.

5.1 Dataset of the example

The dataset consists of details and parameters for the data instance to be run. This is details for the vehicle fleet, the requests, the fixed network, and details for mapping the locations and network of the instance, and is illustrated in Table 5.

Table 5: IDARP test

01 :	nTransferLocations	: 5
02 :	nRequests	: 3
03 :	nVehicles	: 3
04 :	PenaltyServiceTime	: 2
05 :	PenaltyWalking	: 3
06 :	PenaltyTransfer	: 4
07 :	CostOfArc	: 2
08 :	CostOfUsage	: 2
09 :	CostOfVehicle	: 5
10 :	VehicleLoadCapacity	: 4
11 :	VehicleUsageCapacity	: 480
12 :	Depotx	: 10
13 :	Depoty	: 10
14 :	Px	: [00 70 20]
15 :	Py	: [00 20 -90]
16 :	Dx	: [-70 -70 20]
17 :	Dy	: [-20 00 -30]
18 :	Tx	: [-70 00 70 00 00]
19 :	Ty	: [00 00 00 -70 70]
20 :	RequestLoad	: [1 4 3]
21 :	RequestDist	: [25 4 0]
22 :	EarliestServiceTime	: [0000 000 030 120]
23 :	LatestServiceTime	: [1440 090 120 210]
24 :	TravelTimeFixed	: 1.2
25 :	Lines	: [(1,1) 1 (1,2) 2 (1,3) 3 (2,1) 4 (2,2) 2 (2,3) 5]
26 :	ErrorRate	: 0.8

The three first parameters $1-3$ specifies the total number of transportation locations, requests and vehicles in the fleet. Next, parameter $4-9$ sets the unit penalty and cost paid by the fleet, while $10-11$ specify the load and usage capacity of the vehicles in the fleet. All time units, that is, usage capacity and service times are given in minutes. $12-19$ sets the x- and y-coordinate for the depot, the pickup locations, the drop off locations and the transfer locations, respectively. For example, the depot is located at $[10, 10]$, the first pickup location is located at $[00, 00]$ and the second pickup location at $[70, 20]$ etc. $20-23$ specifies details of the requests, the number of people in the request is given in 20, the maximum allowed walking distance is given in 21. Furthermore, $22-23$ gives the earliest and latest pickup time for the vehicles and the requests. In addition, this model assumes that the travel time for the fixed network is proportional the distance traveled with the fixed network and this, and the proportional factor is specified by the parameter $TravelTimeFixed$, given in number 24. Number 25 gives the lines in the fixed network. If a request only uses one line the request uses two transfers, when changing to the fixed network and when leaving the fixed network. In this example, the first line contains of transfer locations 1, 2 and 3, while the second line consists of transfer locations 4 to 5 plus 2. The last parameter 26 sets the $ErrorRate$ that is used when calculating the maximum service time. Thus, with an $ErrorRate$ of 0.8 maximum service time is direct service time multiplied by $(1 + 0.8)$ ($T_{r,r+\overline{r}}(1 + 0.8)$ for request r).

5.2 Details of the example

In this section the data from the example dataset is illustrated, and the network is illustrated in Figure 19. The big black circle represents the depot. In addition, the other black circles represent the fixed network, and the arcs between the transfer locations represents possible trips. When changing between the routes $t_1 - t_2 - t_3$ to $t_4 - t_2 - t_5$ the passengers have to make a transfer at t_2. For each location in the fixed network a set of nodes are created as described in Section 3.3.2, however these are not illustrated in the Figure 19. The blue circles represent pickup locations while the red circles represents drop off locations. For these locations only one node is created. The locations are then numbered as illustrated in the figure. The initial depot is numbered 0, while the final depot, where the vehicle returns after its route is numbered 7. The blue pickup locations are numbered $1-3$ and the red drop off locations $4-6$ respectively. Furthermore, the black transfer locations are numbered $8-12$. The total number of nodes in the network is then: $2 + 3 + 3 + 5 \cdot nNodeVisits$ nodes.

For each arc between two locations there is an associated distance, which is calculated using Euclidean distance. For instance, the distance between location i at (x_i, y_i) and location j at (x_j, y_j) is given below:

$$Distance(i, j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{87}$$

Furthermore, if the request travels with a vehicle there is an associated cost and time for each arc that is proportional to the distance between the locations in this study. If the vehicle travels from initial depot 0 to any location j a fixed cost representing the cost of using a vehicle is added to the cost of traveling on the arc. If the distance is less than the specified maximum walking distance for the request, and the request walks an associated penalty and time for walking between the pair of nodes are added. These values are also proportional to the distance walked in this thesis. Finally, if the two locations are connected so it is possible to use a fixed network an associated time is added for the arc, which is also proportional to the distance traveled. However, since the requests then have to follow fixed network the distance traveled by the fixed route from transfer location 1 to transfer location 5 is $Distance(1,2)+Distance(2,5)$ while the distance if using a vehicle from transfer location 1 to 5 is $Distance(1,5)$. Furthermore, there is a cost of using the vehicle per unit time, a penalty paid for the service time which exceeds direct travel time per unit time, and a penalty paid per transfer.

Furthermore, there is a fleet of homogeneous vehicles, which in this test instance is three vehicles. Each vehicle has a capacity of four passengers and a usage capacity of one working shift of eight hours. In this case, there are two requests specifying a load and time frame that needs to be fulfilled. In addition, each request specifies a maximum walking distance between a pair of locations (Figure 5). The request cannot walk both in and out from a location, but there is no limit on the number of stretches they can walk. Furthermore, the pickup location for request 1 is associated with transfer location 2, while drop off location for request 2 is associated with transfer location 1. Furthermore, as specified in the instance (Figure 5), request 1 specifies a load of one passengers, while request 2 and request 3 specify transportation of four and three passengers respectively. The time window for pickup is given in the data, and the drop off time window is calculated using equations (2) and (3) from Section 3.3.3.

5.3 Solving the example

The problem is to find a feasible travel route for the fleet while minimizing costs and fulfilling all requests. The problem is run with all arc elimination options, strengthened time windows and symmetry breaking constraints on the node visits at the transfer locations and usage of the vehicles together with the simplest option of subtour elimination constraints. The details for some of the runs are provided in the Table 6. The table contains the run of a model without transfers, cooperation and walking options (Test 1000). Furthermore, four runs with all options activated are including, one with one node visit (Test 1111), one with two node visits (Test 2111), one with three node visits (Test 3111) and a run with two node visits and relaxing the $ErrorRate$ so that maximum allowable service time is increased.

With one node visit per transfer location and activating all three options (transfer, cooperation, walking) the problem has 626 rows and 442 columns. When including another node the number of node visits the number of original rows of

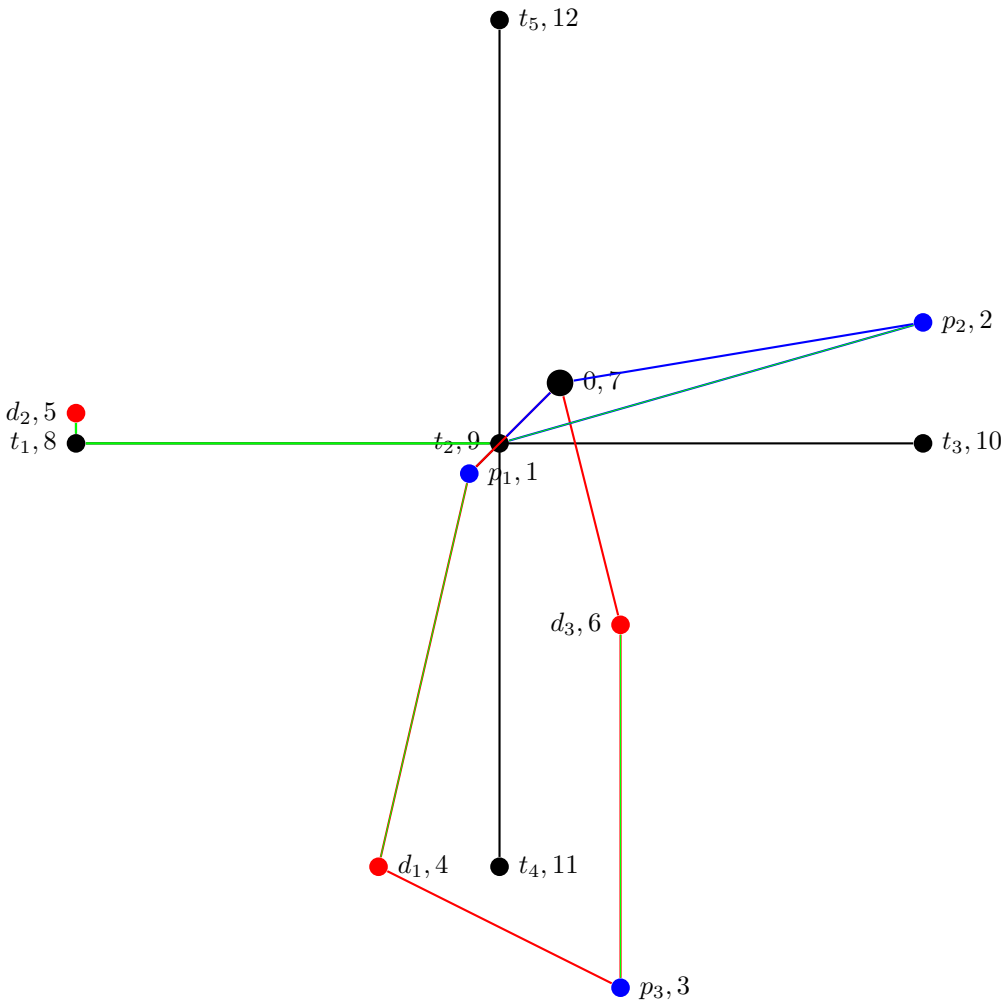

Figure 19: Test Example

the problem seem to double, while the number of columns seems to almost be three times the original value. Furthermore, the pre solving manages to reduce the problem size quite significantly with few node visits, but when including several node visits the total number of rows reduced increase but the reduced percentage decrease significantly, from 67 percent with one node visit to 37 percent with three node visits. Thus, the problem with three node visits is actually so complicated that the instance cannot be solved to optimality within the hour. In comparison, the instance with two node visits was solved within the first two minutes and the instance with one node visit was solved within the first second. Furthermore, the total number of columns reduced decrease with the number of node visits. The benefit of increasing the number of node visits is to potentially find a more efficient solution with a corresponding lower objective value. However, with three requests,

one node visit per location is enough and provides the best possible solution when relaxing the restriction on node visits.

The best feasible solution for the instance activating all options corresponds to a objective value of 1710, where 62 corresponds to the penalties for user inconvenience. The greatest cost corresponds to the distance and usage of the vehicles, and have the values 908 and 730, respectively. The different cost parts and values are given in Table 6. First is the fixed cost of using the vehicles given. Then the variable costs associated with the total distance the vehicles travels and the total usage time of the vehicles. Afterwards, the penalties associated with the user inconvenience are given. First, a variable penalty associated with the excess service time compared to direct service time for each request. Then, the variable penalty paid for the total walking distance by all request, and lastly the penalty paid per transfer. The greatest cost corresponds to the distance and usage of the vehicles, and have the values 908 and 730, respectively.

The best travel route found used two vehicle and is illustrated in Figure 19. The first vehicle travels the red route, while the second vehicle uses the blue route. Green lines illustrate the trips for the requests. Thus, the first vehicle travels from depot to pickup location of request 1 (location 1). There the vehicle picks up request 1 and transport this request to its drop off point, location 4. From there the vehicle travels to pickup location of request 3 (location 3), picks up request 3 and transports it to its final destination at location 6. Afterwards the vehicle returns to the depot. The second vehicle travels from its initial depot to pickup location of request 2 (location 2) to pickup request 2 and transport the request to transfer location 9. Afterwards, the request travels by a fixed route from 9 to 8 and then to its associated drop off location 5. The route of the vehicles and trips for the three requests are given in the Table 6. The sign $-$ illustrate that the arc is driven by a vehicle, $=$ illustrates that the arc is used by a fixed route, while $--$ illustrate that the request walks the arc. For example, request 1 in Test $2111r$ uses the fixed route from its pickup location (location 1) to location 9 to location 8, and then walks from location 8 to its drop off location (location 4).

The example illustrates the benefits of including a fixed route network and the option of walking into the regular dial-a-ride problem. Including these options allows for the objective value to be reduced from 2134 to 1710 and request 2 are able to use the fixed route the last part of the trip. Furthermore, increasing the $ErrorRate$ so request 1 can walk the last part of the trip allows request 1 to travel by itself from pickup to drop off and further reduces the objective value to 1423. However, while the total objective value decreases the part corresponding to penalties increases. Thus, there is a tradeoff between the cost of the service and the convenience for the passengers.

Table 6: Details Example

Detail	Test 1000	Test 1111	Test 2111	Test 3111	Test 2111r
Cooperation On	0	1	1	1	1
Transfer On	0	1	1	1	1
Walking On	0	1	1	1	1
NodeVisits	1	1	2	3	2
ErrorRate	0.8	0.8	0.8	0.8	1.8
Nodes	8	13	18	23	18
Org. Rows	573	626	1330	2271	1360
Org. Cols	410	442	1197	2253	1282
Pre Rows	37	206	763	1426	853
Pre Cols	23	190	1032	2092	1156
Time	0.01	0.77	110.17	3600	36.73
Best integer feasible solution	2134	1710	1710	1814	1423
Best bound	2134	1710	1710	1323	1423
Cost Vehicles	10	10	10	10	5
Cost Distance	1178	908	908	920	572
Cost Usage	946	730	730	740	460
P Service Time	0	58	58	140	318
P Walking	0	0	0	0	60
P Transfers	0	4	4	4	8
Vehicle 1	0-1-4-3-6-7	0-1-4-3-6-7	0-1-4-3-6-7	0-2-10-3-6-7	0-2-10-3-6-7
Vehicle 2	0-2-5-7	0-2-9-7	0-2-9-7	0-9-4-7	-
Request 1	1-4	1-4	1-4	1=9-4	1=9=8- -4
Request 2	2-5	2-9=8=5	2-9=8=5	2-10=8=5	2-10=8=5
Request 3	3-6	3-6	3-6	3-6	3-6

6 Computational Analysis

The model is written in the algebraic modeling language Mosel, implemented in Xpress IVE optimization suite. Several test instances have been solved using Xpress Optimizer. The instances have been run with a HP dl165 $G5$ operation system using a $2x$2.4GHz AMD Opteron 2431 - 6 core processor, with a 24Gb RAM memory.

The instances are generated by using specific fixed network and varying the other characteristics as for example number of requests. Three different standard fixed networks is considered, $S1.4$, $S2.4$ and $S3.5$, where the number 4 and 5 states the number of transfer locations respectively in the networks. The details of the networks are provided in Appendix A.2. For each fixed network instances with three to 15 requests are generated. The number of requests in the instance is added in the manner *.requests*, after the number of transfer locations. For example, the network $S1.4$ with three requests are written $S1.4.3$. Furthermore, an r is added after the number of requests if the $ErrorRate$ is increased from 0.8 to 1.8 (for example $S1.4.3r$). Similarly, if a maxi vehicle with a capacity of eight is used, instead of the regular size of four passengers, a m is added to the name. Additionally, stw specify that the time windows are cut in half, while btw signalize that the time windows are double the regular size. In addition, the number of node visits are given in brackets, while the problem features are given with three binary numbers, where 1 says the option is on, while 0 says that the option is off. The fixed network and the pickup and drop off locations for the instance with seven requests ($S1.4.7$, $S2.4.7$ and $S3.5.7$) are illustrated in Appendix A.2. Furthermore, instances considering bigger fixed network $B1.7$, $B2.8$ and $B3.9$, of seven to nine transfer locations are tested (Appendix A.3). In addition, three case instances, $H1.4$, $C2.7$ and $R1.4$ are tested. First of all, $H1.4$ illustrates a setting where the majority of the requests are going to a common location, in this case a hospital, and the details are given in Appendix A.1.1. The instance is run with narrow time windows ($H1.4.6stw$), regular time windows ($H1.4.6$) and big time windows ($H1.4.6btw$). $C2.7$ represents the case with a city center and a more well established fixed network. The fixed network consists of the seven transfer locations and is illustrated in Appendix A.1.2 together with details for the results with three requests ($C1.7.3$), with regular time windows ($C1.7.3$) and narrow time windows ($C1.7.3stw$). $R1.4$ represents a sparse fixed network where two requests with load less than a single vehicle load capacity is located far from the rest of the requests. The fixed network has four transfer locations and consists of two separate lines located in different part of the town. The fixed network and the instance with six requests ($R1.4.6$) is illustrated in Appendix A.1.3 for the case with a regular vehicle capacity of four ($R1.4.6$) and for maxi vehicles with capacity of eight ($R1.4.6m$).

This section starts by going through a preliminary study (Section 6.1) discussing the effect of different reduction techniques and valid inequalities on the problem complexity. The discussion ends by stating a default option that is used in the rest of the analysis (Section 6.2 and Section 6.3). Section 6.2 considers the technical aspects of different characteristics of the problem and especially incorporating the problem features (transfer, multiple requests, walking). In addition, the effect of

varying the number of node visits is discussed. Furthermore, changing the details of the vehicle fleet and the requests are considered. In Section 6.3 the economical effects of the problem features and details of the vehicle fleet and requests is discussed from the operator and drivers, and users point of view. In short, the problem features considerably complicate the problem from a technical perspective. However, the features provide economical benefits from the operator, especially the feature of transfer, while the characteristics might decrease the convenience for the user.

The resulting default setting is given in Section 6.1, and this setting is used in the rest of the analysis. In short, incorporating the problem features of transfer, multiple requests and walking considerably complicates the problem and makes it more difficult to solve the problem and find acceptable solutions. However, incorporating the problem features makes it possible to reduce the cost of the fleet while at the same time maintaining an acceptable level of convenience for the users.

6.1 Preliminary Analysis

One of the main focuses in this thesis is to incorporate reduction techniques and valid inequalities into the formulation so as to reduce the complexity of the problem. In detail, this means to formulate the problem in a way so the resulting solution space is as similar as possible to the convex hull. When the solution space is equal to the convex hull the LP relaxation (lower bound) is equal to the optimal integer solution, and it is possible to solve the LP relaxation to find the optimal IP solution. Solving the LP problem can be done using simplex and is a lot easier than solving the IP problem. Several ways of strengthening the model is discussed in Section 4, and the reduction techniques are grouped into different options. This section goes through the different groupings of reduction techniques, referred to as run options of the model, and explains their effect on the solution process and on the problem. The options should not have an impact on the practical solution of the problem, but can lead to a better solution given the same amount of time, and the options can have differing effects with different problem features. The section goes through the different groupings of reduction techniques, referred to as run options of the model, and explains their effect on the solution process and on the problem. This discussion is centered around finding a default option for further analysis, and the use of the option is specified at the end of each subsection.

The default option is to activate all arc elimination options, the strengthening of time windows, the symmetry breaking constraints ordering the nodes and ordering the usage of the vehicle, and all subtour elimination constraints. The options considering mixed integer rounding capacity constraints and setting the least number of vehicles is set off in the default option.

6.1.1 Arc Elimination Constraints

The first group of options to consider is the arc elimination constraints (see Section 4.1. The first of the AEC options consider basic arc elimination and thus for example which nodes can be visited after which nodes. Without the option all arcs between all nodes have to be created and the problem would be quite complicated to solve, and considerably bigger in size. Including the AEC options considering time windows (AEC TW) or capacity limits (AEC Q) tend to reduce the size of the problem further. On average, the option of considering time windows tend to have a greater effect than the option of including capacity limits, but combining the two clearly have the greatest effect. In fact, including this option also increases size of the LP relaxation from 777 to 799 and thus creates a better initial lower bound for the solution. The reduction in columns of the problem (reduction in variables) is greatest, but some reduction in rows (constraints) exists as well. This is because arc elimination constraints eliminate variables (arcs) but in doing so some constraints are eliminated as well. For example, in Table 7 and Table 8, where the number of original rows and columns are provided for the instances $S1.4.3(2)$, $S2.4.5(2)$, $S2.4.5(3)$ and $S3.5.7(3)$, these effects are evident. For instance, the first row of the Table 7 gives the rows (1070) and columns (1286) for instance $S1.4.3$ with two node visits and AECbasic, while the second row shows the rows (1004) and columns (1068) for the instance $S1.4.3$ also including AEC TW. The last column gives the reduction in rows (6.17 percent) and columns (16.95 percent). Furthermore, similar effects appear for the pre solved number of rows and columns as well. In the tables, the number in brackets represents the maximum number of node visits per location, and the options given afterward states which options is included when solving the instance. The percentage reduction in rows and columns compared to the original problem is calculated as:

$$\frac{plainInstance - withOptionInstance}{plainInstance} \tag{88}$$

Looking at instance $S2.4.5$ in Table 8, where the size of the problem with two and three node visits are shown, increasing the number of node visits reduces the effects of including AEC TW and AEC Q. In fact, the same number of rows is eliminated in both problems. Considering the number of columns that are eliminated, this number increases only slightly. Because of the reduction in size of the problem, the software manages to solve the problem a bit faster on average. Thus, as a consequence of the positive effects when solving the problem all AEC options should be included in the default option.

Table 7: Effect with Arc Elimination Options 1

Instance	Original rows /cols	Percentage Reduction
S1.4.3 (2)	1070/1286	
S1.4.3 (2) AEC TW	1004/1068	6.17/16.95
S1.4.3 (2) AEC Q	1032/1128	3.55/12.29
S1.4.3 (2) AEC TW,AEC Q	986/956	7.85/25.66
S3.5.7 (3)	4855/10959	
S3.5.7 (3) AEC TW	4471/8660	7.91/20.98
S3.5.7 (3) AEC Q	4711/9394	2.97/14.28
S3.5.7 (3) AEC TW,AEC Q	4414/7626	9.08/30.41

Table 8: Effect with Arc Elimination Options 2

Instance	Original rows /cols	Percentage Reduction
S2.4.5 (2)	1710/2912	
S2.4.5 (2) AEC TW	1485/2176	13.16/25.27
S2.4.5 (2) AEC Q	1620/2338	5.26/19.71
S2.4.5 (2) AEC TW,AEC Q	1473/1916	13.86/34.20
S2.4.5 (3)	2552/4599	
S2.4.5 (3) AEC TW	2327/3721	8.82/19.09
S2.4.5 (3) AEC Q	2462/3906	3.53/15.07
S2.4.5 (3) AEC TW,AEC Q	2315/3342	9.29/27.33

6.1.2 Strengthened Time Windows

Another option considers strengthening of time windows which is also discussed in Section 4.1. When running the model with and without this option all arc elimination constraints are activated. The effects of strengthening the time windows for the service at the location increase the number of rows of the problem slightly. Furthermore, the effect on the solution time seems to be very instance specific, but strengthening the time windows does not affect the LP relaxation of the solution. In the Table 9 the time in seconds at certain gaps and the total solution time for the smallest dataset are provided. The gap is calculated similarly as the reduction, that is:

$$\frac{BestFeasible - BestBound}{BestFeasible} \tag{89}$$

As always, the number in brackets represents the maximum number of node visits per location, and the options given afterwards state which options is included when solving the instance. For example, on the first row in Table 9, the details of the running of instance $S1.4.3$ with two node visits and activating the arc elimination constraint options. The second column state the time in seconds at 20 percent gap (8.12 seconds), and then at ten and five percent gap, while the last column give the total solution time in seconds. The next row shows the solution

of the instance also including the option of time window strengthening. For two ($S1.4.3$ and $S3.5.3$) out of three instances the solution time is reduced to half the regular time if time window strengthening is incorporated into the implementation. Furthermore, the time until 20 percent gap is shorter for two ($S1.4.3$ and $S2.4.3$) out of three instances, though not the same two dataset. Similarly, in Table 10 the percentage gap is less for two ($S1.4.7$ and $S2.4.7$) out of three instances. Table 10 states the best solution, best bound and the resulting gap after 30 minutes solution time. Notice that the two bigger instances (seven requests) and the two smaller instances (three requests) with the shorter time until 20 percent gap are the same instances. Thus, it is possible that by allowing the program to finish solving the bigger instances, the option of time window strengthening will shorten the solution time for instance $S1.4.7$ and $S3.5.7$ similarly as with the smaller instances. In conclusion, including time window strengthening will more often than not result in a better solution time either for finding the optimal solution or for finding an acceptable solution for small and big datasets. As a consequence, this option should be included as a default option in further analysis.

Table 9: Effect with Time Window strengthening 1

Instance	Time 20% gap (seconds)	Time 10% gap (seconds)	Time 5% gap (seconds)	Total time (seconds)
S1.4.3 (2) AEC	8.12	433.46	777.91	9187.27
S1.4.3 (2) AEC,TW	4.30	60.76	140.59	870.75
S2.4.3 (2) AEC	31.24	154.57	229.56	277.07
S2.4.3 (2) AEC,TW	28.78	229.60	346.27	412.43
S3.5.3 (2) AEC	674.96	1628.73	2078.66	13762.73
S3.5.3 (2) AEC,TW	1437.25	4087.47	5974.91	8045.14

Table 10: Effect with Time Window strengthening (1800) 2

Instance	Best Solution	Best Bound	Percentage Gap
S1.4.7 (3) AEC	4089.29	2372.84	41.97
S1.4.7 (3) AEC,TW	2694.61	2364.55	12.25
S2.4.7 (3) AEC	2436.05	2084.90	14.41
S2.4.7 (3) AEC,TW	2459.02	2245.47	8.68
S3.5.7 (3) AEC	1596.98	1517.00	5.01
S3.5.7 (3) AEC,TW	1880.32	1517.00	19.32

In the Section 6.1.3 to Section 6.1.5 all arc elimination options and the time window strengthening option is activated when testing the options.

6.1.3 Symmetry Breaking Constraints

In Section 4.2 several symmetry constraints are discussed, and constraints (53) to constraints (56) is implemented in the model. The symmetry breaking constraints are divided into four options, one considering ordering of the nodes (SBC, constraints (53)) while the others consider ordering of the vehicles (SEC1 constraints (54) to SEC3 constraints (56)). SEC1 orders the vehicles after their number (constraints (54)), SEC2 orders the vehicles after their usage (constraints (55)) and SEC3 orders the vehicles after their total distance traveled (constraints (56)). The option of ordering the nodes can be combine with one of the options which set the order of the vehicles.

All symmetry elimination options have a great effect on the complexity of the problem and reduce the solution time of the problem. However, none of the options manages to increase the value of the lower bound (LP-relaxation). The option of ordering node visits seems to reduce the number of columns considerably and the number of rows slightly. Ordering of the vehicles has no effect on the number of columns, and quite small effect on the number of rows. The symmetry breaking constraints reduce the total solution time quite significantly. For the instances provided in Table 11, $S1.4.3$, $S2.4.3$ and $S3.5.3$, the total time in seconds without symmetry breaking constraints is given in the first row. In the following rows the total times including SBC, and then SBC and SBC1, SBC and SBC2 or SBC and SBC3 is stated. Including ordering of the nodes creates a saving of 94, 96 and 97 percent in solution time, respectively for instances $S1.4.3$, $S2.4.3$ and $S3.5.3$. Furthermore, including one of the options SBC1, SBC2 and SBC3 in combination with SBC provide possible further savings. Thus, the findings clearly show that the options of including symmetry breaking constraints are efficient and should be included as default options. Since only one way of ordering vehicles can be used in combination with ordering nodes, ordering the vehicles after usage seems to be the best performing option on average. Consequently, the combination of SBC and SBC2 is used in the default option.

6.1.4 Subtour Elimination Constraints

The next option considers constraints which eliminates subtours in the solutions in the branch and bound tree (see Section 4.3). The first option SEC1 (constraints (70) and constraints (71)) considers basic subtour elimination constraints between two nodes. The second option SEC2 (constraints (72)) considers iteratively added subsets with three to six locations. The last option SEC3 (constraints (73) to constraints (78)) considers a set of lifted subtour elimination constraints.

The technical effects of the different options seem to vary with the instance run. On average, SEC1 seems to have the greatest positive effects on the smaller data instances (three requests), while SEC2 and SEC3 sometimes complicate the solution procedure for smaller instances. However, for bigger instances with five requests SEC2 and/or SEC3 seems to perform best and improve the solution procedure. Some of the bigger instances favor a combination of two or three SEC-

Table 11: Effect with Symmetry Breaking Constraints

Instance	Total Time (seconds)
S1.4.3 (2)	868.15
S1.4.3 (2) SBC	54.26
S1.4.3 (2) SBC,SBC1	20.73
S1.4.3 (2) SBC,SBC2	10.44
S1.4.3 (2) SBC,SBC3	13.26
S2.4.3 (2)	412.43
S2.4.3 (2) SBC	14.93
S2.4.3 (2) SBC,SBC1	10.77
S2.4.3 (2) SBC,SBC2	8.05
S2.4.3 (2) SBC,SBC3	17.32
S3.5.3 (2)	8045.14
S3.5.3 (2) SBC	222.94
S3.5.3 (2) SBC,SBC1	311.13
S3.5.3 (2) SBC,SBC2	135.85
S3.5.3 (2) SBC,SBC3	156.03

options. However, none of the subtour elimination constraints manages to increase the lower bound (LP-relaxation). As an example, test instances with fixed route network $S3.5$ is provided in Table 12. The first row states the best feasible solution, the best bound and the resulting gap for the instance $S3.5.3$ without subtour elimination constraints. The next three rows gives the value for the instance with option SEC1, SEC2 and SEC3 respectively, while row five provides the best solution, best bound and resulting gap if all SECs are activated when running $S3.5.3$. Considering the SEC options individually for $S3.4.3$ it is clear that SEC1 provides the best savings, while SEC2 complicates the solution process. However, for instance $S3.5.3$ including all three SEC options provides additional savings during the solution process. For the majority of the instances this is usually not the case. Next, for instance $S3.5.5$, SEC2 provides the best solution, while SEC1 and SEC3 slightly complicates the solution process. Notice that for this instance as well, including all subtour elimination constraints seems an acceptable alternative. Unfortunately, it is difficult to clearly state which option is the best subtour elimination option in general, but it seems that including all options in combination provides the best performance on average. This is because in a few cases activating all options creates additional synergies and further savings, while in most cases the savings from including all SEC options are similar to the savings from the best single option. However, it is not possible to know which single option will provide the greatest savings. Thus, including all SECs as a default option seems to be the favorable alternative.

Table 12: Effect with Subtour Elimination Constraints

Instance	Best Solution	Best Bound	Percentage Gap
S3.5.3 (2)	1710.00	1401.44	18.04
S3.5.3 (2) SEC1	1710.00	1549.89	9.36
S3.5.3 (2) SEC2	1710.00	1325.00	22.51
S3.5.3 (2) SEC3	1710.00	1442.44	15.65
S3.5.3 (2) SEC123	1710.00	1710.00	0.00
S3.5.5 (2)	2435.00	1558.00	36.02
S3.5.5 (2) SEC1	2435.00	1547.00	36.47
S3.5.5 (2) SEC2	2435.00	1668.63	31.47
S3.5.5 (2) SEC3	2435.00	1552.16	36.26
S3.5.5 (2) SEC123	2435.00	1645.50	32.42

6.1.5 Mixed Integer Rounding Constraints

Some options considering mixed integer rounding capacity constraints are included in the implementation and stated in Section 4.4. The first option MIRQ1 (constraints (83)) considers the load to be picked up or delivered at the respective pickup or drop off locations in the subsets, while the second option MIRQ2 (constraints (84)) also considers the flow in and out of the subsets. Lastly, an option considering calculating the least number of vehicles that have to be included in the optimal solution is discussed. These options do not seem to have the greatest effect on reducing the problem complexity, probably because they only considers subsets which consists of pickup (or drop off) locations not associated with a transfer locations or within walking distance of other nodes.

By looking at some of the characteristics of the problem and instances, the options can have a greater positive effect on the complexity of the problem. These characteristics are: (1) Increase the size of the instance, or at least the proportion of requests which are not associated with the transfer locations or within walking distance of other locations, (2) narrowing the time windows for the requests, and (3) excluding one or more of the problem features. By excluding the feature of walking the groups of constraints considering mixed integer rounding and setting the least number of vehicles used in the solution have a greater positive effect on the solution process. Reduce the proportion of requests associated with a fixed route also improve the effect of these technical options. For example, consider the instance with 14 requests (Table 13) and the small fixed networks $S1.4$, $S2.4$ and $S3.5$, excluding the option of walking and with narrow time windows. Then there are some technical savings to be made for all instances and the resulting gap is reduced. Notice, however, that the benefits are still quite small. In the Table 13, the instance is stated in the first column, and the three binary variables represents which of the options (transfer, multiple requests, walking) which is on, and $+MIRs$ state if the three technical options considered in this subsection is activated. For each of the instances the best feasible solution found, the best bound and the percentage gap after two hours solution time is given in the table.

Table 13: Effect with Mixed Integer Rounding (7200)

Instance	Best Solution	Best Bound	Percentage Gap
S1.4.14 (1) 110	8109.80	5263.05	35.1
S1.4.14 (1) 110 MIRs	8109.80	5288.90	34.8
S2.4.14 (1) 010	7357.00	6285.54	14.6
S2.4.14 (1) 010 MIRs	7357.00	6302.87	14.3
S3.4.14 (1) 110	6502.00	4740.56	27.1
S3.4.14 (1) 110 MIRs	6502.00	4786.69	26.4

These options are not activated in the default option, as the technical advantages are quite neglectable in general.

6.1.6 The Resulting Default option

As a default setting all arc elimination options and the time window strengthening are set to 1. The symmetry breaking constraints considering ordering the nodes after increasing number (SBC) and ordering the vehicles after usage (SBC2) is activated. At last, all subtour elimination constraints are included.

Table 14: Default option

Option	On/Off
AEC Basic	On
AEC TW	On
AEC Q	On
TW	On
SBC	On
SBC1	Off
SBC2	On
SBC3	Off
SEC1	On
SEC2	On
SEC3	On
MIRVe	Off
MIRQ1	Off
MIRQ2	Off

6.2 Technical Analysis

The technical aspect of the problem is important to consider since if the problem is so complicated that it is too difficult to find a feasible and acceptable solution within the time provided it does not matter how efficient the solution could have been because the operator will not manage to find the solution. Furthermore, as the complexity of the problem increases the investments in software and expertise, and consequently the cost of the service, increases as well. Thus, being able to find a good enough solution in the limited time provided is necessary to be able to provide a reliable and efficient service. Routing problems are NP-hard problems, and integrating a demand responsive service with a fixed route service seems to create an even more complicated problem. The focus in this thesis have been on incorporating valid inequalities to find a solution space which is as similar as possible to the convex hull of the solution space.

In this section the technical effects on solving the problem with the different problem features (transferring, multiple requests, walking), and varying the maximum number of node visits, varying the service time, the number of vehicles and the size of the time windows are discussed. The problems solved in this section is set with the following default setting of technical options; all arc elimination options, the time window strengthening option, the symmetry breaking constraints ordering the nodes after number and vehicles after usage and all subtour elimination constraints.

To solve the problem instances Xpress IVE optimization suite is used and it is a great tool to solve complex IP problems. The problem consists of several complicating issues. When increasing the number of node visits the complexity of solving the problem seems to increase exponentially. The same effects are evident when including the option of transfer and/or the option of walking. The option of transporting multiple requests at the same time, increasing the maximum service time of the request, increasing the number of vehicles and regulating the time windows increases the size of the problem to different extents. Small instances of the problem are solvable within reasonable time.

6.2.1 Problem Features

Three problem features are specified and affect the practical solution of the problem. The different characteristics can be used independently or in combination:

- $TransferOn$: The implementation allows for the possibility to transfer to another vehicle or a fixed network, and thus using the fixed network a part of the trip from pickup to drop off location.
- $CooperationOn$: The implementation allows for the possibility to transport several requests in the same vehicle at the same time.
- $WalkingOn$: The implementation allows for the possibility to walk between nodes.

In general the three characteristics seems to complicate the already NP-hard problem and makes it harder and more complex to solve. On the other hand, these characteristics often have positive economical effect and the resulting cost of operating the fleet can decreases, though sometimes at a consequence of decreased convenience for the users. To balance the reduced costs and increased user inconvenience penalties are added to the objective function, and the results shows that the objective value is usually still lower compared to the objective value without the problem features.

In the Table 15 and Table 16 the number in brackets represents the maximum number of node visits at a location. The three last numbers represents which of the problem features (transfer, multiple requests, walking) that are activated. That is, if the first number is 1 the option of transfer is on, the second 1 represents that the option of multiple requests is on, while the last number is 1 when the option of walking is on. The r represents that the $ErrorRate$ has been set to 1.8 instead of 0.8. In Table 15 the best solution, the total time in seconds and the pre solved rows and columns are stated. In Table 16 the original rows and columns and the number of rows and columns after the pre solve is given, together with the percentage reduction in rows and columns.

Table 15: Problem Features 1

Instance	Best Solution	Total Time (% Gap) (seconds)	Pre. Rows/Cols
S1.4.5 (2) 000	3899.00	0.07	119/83
S1.4.5 (2) 001	3899.00	9.47	791/1216
S1.4.5 (2) 010	3269.00	1.24	260/174
S1.4.5 (2) 100	3289.20	68.66	892/1335
S1.4.5 (2) 111	2659.20	263.23	1116/1817
S1.4.5r (2) 111	2659.20	1298.90	1337/2057
S3.5.5 (2) 000	3209.00	0.31	138/85
S3.5.5 (2) 001	3123.00	25.16	968/1622
S3.5.5 (2) 010	3153.00	0.21	168/100
S3.5.5 (2) 100	2435.00	89.28	1076/1768
S3.5.5 (2) 111	2435.00	14400 (9)	1274/2140
S3.5.5r (2) 111	2144.00	11523.94	1435/2328

Transferring between Transportation Modes:
One of the main characteristics of the model is the possibility of changing between modes of transportation. Incorporating this possibility into the model allows the vehicle to set off the request at the transfer locations, where another vehicle can pick up the request or the request can continue its trip using the fixed route. The vehicle does not have to transport the request from pickup and all the way to its drop off location. Incorporating this feature into the model increases the flexibility the operator has when he or she schedule the routes for the vehicles, and makes it possible to find cheaper combination of routes that fulfill all requests.

Table 16: Problem Features 2

Instance	Org. Rows/Cols	Pre. Rows/Cols	Percent Reduction
S1.4.5 (2) 010	2491/1912	260/174	90/91
S1.4.5 (2) 011	2491/1957	1058/1767	58/10
S1.4.5 (2) 110	2773/1952	1119/1801	60/8
S1.4.5 (2) 111	2773/1997	1116/1817	60/9
S1.4.5 (3) 010	3159/3164	259/172	92/95
S1.4.5 (3) 011	3159/3257	1603/3047	49/6
S1.4.5 (3) 110	3792/3263	1710/3063	55/6
S1.4.5 (3) 111	3792/3356	1750/3160	54/6

However, since the requests might have to transfer between different modes of transportation, the convenience for the users might decreases. Incorporating this characteristic into the model considerably complicates the solution process. First of all a set of transfer locations have to be created in addition to the regular sets of pickup and drop off locations. Furthermore, for each transfer location a set of nodes equal to the maximum number of node visits have to be created. Thus, $nTransferLocations * nNodeVisits$ additional nodes have to be created. In addition, all possible arcs going in and out from these nodes have to be created. Thus, the number of binary variables increases considerably since the number of arcs between pair of nodes increases. For each arc a binary variable has to be created for each vehicle which can travel on the arc, $x_{imjnk}, (i,m) \in \mathcal{N}, (j,n) \in \mathcal{N}, k \in \mathcal{K}$, each request able to travel on the arc, $y_{imjnr}, (i,m) \in \mathcal{N}, (j,n) \in \mathcal{N}, r \in \mathcal{R}$ and each request able to walk on the arc, $v_{imjnr}, (i,m) \in \mathcal{N}, (j,n) \in \mathcal{N}, r \in \mathcal{R}$. Furthermore, if the arc represents a part of a fixed route, $Z_{ij} = 1$, binary variables have to be created for all requests able to use the fixed route on this arc, $w_{imjnr}, (i,m) \in \mathcal{N}, (j,n) \in \mathcal{N}, r \in \mathcal{R}$. The number of ways these arcs can be combined to set of routes fulfilling all requests increases with the increased flexibility. Thus, the solution space is increased from the ordinary set of solutions to a set of ordinary and new solutions. As a result, the optimal solution cannot be worse incorporating the feature of transfer, but it is possible that the optimal solution is improved. However, because of increased size of the problem and solution space it might take longer and the software have to search through a greater number of possibly poor solutions before finding the improve solutions. The possibility of transfer seems to be the problem characteristic which creates the greatest economical savings, and by looking at instance $S3.5.5$ in Table 15 this possibility seems to create just as good savings as including all three problem features in combination. However, including the other two problem features usually enhance the savings created by using the fixed route, since for example a vehicle can simultaneously pickup or drop off several requests at a transfer location. For example, the objective value of $S1.4.5$ is clearly improved by including walking and transporting multiple requests at the same time together with using the fixed network as seen in Table 15. On the other hand, the possibility of transfer also causes the greatest increase in the number of rows and columns of the problem, and considerably complicates the process of finding the optimal solution. The respective solution time is thus

increased from less than a second to over a minute.

Transporting Multiple Requests at the Same Time:
The second characteristic of the model is the possibility to transport multiple requests in the same vehicle at the same time. For example, consider two requests located in a place quite far from all the other requests, and with a combined load of less than a single vehicle capacity. Incorporating the feature of transporting multiple requests at the same time, one vehicle can pickup both requests and transport them to their drop off locations. If the requests are going to a similar place, say into town, the economical savings are greater because the vehicle almost saves a whole trip compared to the situation where only one request can be transported at a time. The economical savings are also often great if the requests are going to different locations and especially if it is possible to take advantage of the fixed route for one or both trips. Considering the total objective value, the cost of operating the fleet is reduced, but because at least one request usually has a longer service time than the direct service time, the penalties might be slightly higher. In technical terms transporting multiple requests in the same vehicle at the same time is the same as letting $\sum_{r\in\mathcal{R}} y_{imjnr} \in [2, 3\dots], (i,m) \in \mathcal{N}, (j,n) \in \mathcal{N}$. Incorporating this feature thus means that it is, among others, not possible to limit $\sum_{r\in\mathcal{R}} y_{imjnr} \in [0,1], (i,m) \in \mathcal{N}, (j,n) \in \mathcal{N}$, and the size of the problem increases slightly. Considering Table 15, one sees that this possibility of transporting multiple requests creates economical savings and at the same time only slightly complicates the solution process. The possibility of transporting multiple requests at the same time is also beneficial to include together with other problem features to increase flexibility.

Walking some Part of the Trip:
The third characteristic implemented in this model is to allow for the requests to walk an individually based distance between a pair of node. That is, without this feature vehicles have to transport the requests from (to) their exact pickup (drop off) location if the location is not associated with a transfer location. However, implementing this feature in the model allow request to walk out from (in to) its pickup (drop off) location or between locations during the trip. Considering the example with the two requests located far from other request, one of the requests can walk to the other request's pickup location so they can be picked up together, so the vehicle only has to make one stop when picking up the requests. Furthermore, if there are two fixed network with two transfer locations located in proximity of each other a request might manage to walk between the two networks instead of having to be transported by a vehicle between the two fixed networks. Thus, the flexibility for the operator when scheduling the trips are increased and possible economical savings may occur. However, since walking takes longer compared to driving, the maximum service time might restrict the use of this possibility. In addition, the benefits of the feature are only evident in combination with at least one of the other problem features. From a technical point of view, the possibility of walking complicates the solution procedure because the number of arcs increases slightly. However, the greatest complicating factor for including this feature in the model is that the number of arcs which are possible to eliminate during the pre

solve implemented in Xpress IVE and through the reduction techniques discussed in Section 4 are reduced considerably. This is because as long as the distance from a location $(i, m) \in \mathcal{N}$ to another location $(j, n) \in \mathcal{N}$ is less than the maximum walking distance for the request, it is not possible to force a vehicle to visit the location. Furthermore, considering the minimum number of visits to a subset of locations the model has to exclude all locations where a request can walk in or out from the location. Thus, many of the reduction techniques have less effect when incorporating the possibility of walking and the complexity of the problem increases. As an example, look at the Table 15 and instance $S3.5.5$, including the possibility of walking provides a saving in the objective value but increases the size of the problem. Furthermore, since walking takes longer compared to driving the effects of this possibility is better if one allows for a longer service time. The possibility of walking also creates the greatest economical savings in combination with one or both of the other possibilities. This is because the possibility of walking is often used to walk to a nearby location to be picked up together with another requests, or to walk to or from a nearby fixed network. This can for example be seen in $S1.4.5$ where allowing for using all features in combination creates greater economical savings than only including one option.

The Three Problem Features:
Including the possibility of transfer increases the size of the problem considerably, while including the possibility of walking increases the size slightly. Including both possibilities of course increases the size of the problem considerably. This is reasonably, since the features provide flexibility from a practical point of view. This flexibility increases the possible solutions and thus the possible variables and constraints, and the resulting problem is bigger. A bigger problem size means an increased size of the solution space, which makes the search for the optimal solution more exhaustive, and thus the resulting solution process more complicated. The possibility of transfer and walking also makes it more difficult to reduce the size of the problem through valid inequalities and Xpress IVEs own pre solving. For example, consider the instances with two node visits. If neither transfers to other vehicles or a fixed network is allowed, the pre solving manages to reduce the total number of rows and columns by 90 percent. Including one or both of these options the reduction from original rows and columns to pre solved rows and columns decreases to about 60 percent for the rows and less than 10 percent for the columns. Thus, including the problem features increases the original problem by about 10 percent for the rows and 4 percent for the columns. However, considering the size of the problem after pre solve, the number of rows are increased by 77 percent and the number of columns are increased by about 90 percent.

The number of node visits that are considered when solving the problem also affects the size of the problem. In the Table 16, considering the instance not activating the option of transfers and walking (010), increasing the number of node visits from 2 to 3 increases the original size of the problem by about 20 and 40 percent in rows and columns, respectively. However, the size after pre solving is the same as with two node visits. This is quite understandably considering that the node visits affect the number of transfer nodes, and without transfer nodes

the number of node visits should therefore not affect the size of the problem. For the case with both transfers and walking (111), the size of the original problem increases similarly when increasing the number of node visits from two to three. However, after the pre solving, the number of rows increases by 36 percent, while the number of columns increases by about 43 percent.

6.2.2 The Maximum number of Node Visits

One important aspect when formulating the IDARP model is how to model the transfer locations, since the number of visits is not necessary restricted to one at the locations. The formulation implemented in this thesis creates a set of artificial nodes for each transfer location. The set of nodes contains the maximum number of visits at the locations and restricts each node to only be visited once. The issue is to specify the size of this set (the number of node visits). In the practical sense of the problem, there is no limit on the number of times a location can be visited, and thus the solution equal to the solution found when relaxing the number of node visits is the optimal practical solution. It is then beneficial to implement a model that allow for the necessary visits and where the optimal practical solution is feasible. For example, if the user set the number of node visits to maximum two visits, and the practical optimal solution specify that three of the requests should visit transfer location 1, this solution is not feasible in the implemented model. The implementation of the problem can allow for different number of maximum node visits at the transfer locations, and in this study two different options can be chosen between. First of all, the user can specify a number of node visits per location for all the transfer locations, or the user can set the number of node visits per transfer location on an individual basis for the transfer locations.

An upper bound for the set of node visits which works well on the smallest instances is to set the maximum number of visits equal to the number of requests. Then each request can visit the transfer location once. However, when increasing the size of the problem the chance that each request visits every transfer location, or that a transfer location is visited by all requests, is minimal. Thus, it might be possible to reduce the number of node visits per transfer location without cutting away the optimal solution. Reducing the number of node visits, while holding the number of transfer locations fixed, reduces the size of the problem. For a bigger fixed network the effect of changing the number of node visits are increased since changing the number of node visits from n to $n+1$ increases the number of nodes by $nTransferLocations$. That is, when increasing the number of node visits the size of the problem increases considerably because a whole new set of transfer locations is created with their own sets of arcs to and from the locations. As a consequence, it is beneficial to keep the maximum number of node visits as low as possible without cutting away the optimal practical solution. As an example Table 17 and Table 18 display details for the instance $S1.4.5$ with four transfer locations and five requests. For these solutions all problem features (transfer, multiple requests, walking) are activated. The solution time is set to 30 minutes (1800) and the number of node visits are varied from one to five. The option of flexible node visit is off so that

all transfer locations have the same number of nodes per location. Table 17 shows the number of rows and columns for the original problem and the problem after pre solve for instance $S1.4.5$ for different number of node visits. In addition, the percent reduction after pre solve of the problem is stated in the last column. For example, for the first row in Table 17, the instance run is $S1.4.5$ with one node visit. The original number of rows is 1978 and during the pre solve the number of rows are reduced by 68 percent to 631 rows after pre solve. Table 18 gives the best solution, best bound and resulting gap after 30 minutes solution time. The number of rows and columns increases considerably as the number of node visits increases. Furthermore, the reduction in the problem after pre solve decreases as the number of node visits increases. The increased problem size increases the solution time of the problem. From Table 18 it is reasonable to assume that instance $S1.4.5$ does not require more than one visit per location for finding the best possible practical solution. Thus, even though the instance have five requests, only one node visit per transfer location is necessary. Furthermore, with one node visit per transfer location the problem is solved in neglectable time. Increasing the number of node visits to two increases the solution time to five minutes. Furthermore, increasing the number of node visits to three result in nine percent gap in 30 minutes. For the instance run with five node visits per transfer location, Xpress does not manage to find a feasible solution within the time provided (30 minutes). To find a feasible solution to instance $S1.4.5$ with five node visits, the software needs almost two and a halve hour solution time, and the gap for the first feasible solution is between 15 and 20 percent. Thus, it is great benefits to be achieved by restricting the number of node visits on some or all locations. However, bigger instances or structurally different instances might require a higher number of node visits for all or some of the transfer locations.

Table 17: Varying the Number of NodeVisits 1

Instance	Org. Rows/ Cols	Pre. Rows/ Cols	Percent Reduction
S1.4.5 (1)	1978/ 936	631/ 760	68 / 19
S1.4.5 (2)	2773/ 1997	1116/ 1817	60 / 9
S1.4.5 (3)	3792/ 3356	1750/ 3160	54 / 6
S1.4.5 (5)	6502/ 6968	3371/ 6656	48 / 4

Table 18: Varying the Number of NodeVisits 2

Instance	Best Solution	Best Bound	Percentage Gap
S1.4.5 (1)	2659.20	2659.20	0
S1.4.5 (2)	2659.20	2659.20	0
S1.4.5 (3)	2659.20	2414.00	9
S1.4.5 (5)	-	2155.00	-

The details of instances $S2.4.5r$ and $S3.5.5r$, where the $ErrorRate$ is set to 1.8 instead of 0.8 and the solution time is set equal to four hours (14400), is shown in Table 19. Again the instances are solved with all problem features, and the number

of node visits is stated in brackets. For instance $S2.4.5r$, increasing the number of node visits from two to three node visits makes the problem too difficult to find a feasible solution within the four hours provided. With instance $S3.5.5r$ increasing the number of node visits from one to two allows for request 1 to travel from location 13 to 12 with the fixed route, and thus manages to walk to its destination at location 6. With only one node visit per location this is not possible because request 2 visits location 12 on its trip from pickup to drop off location. Thus, in the case with one node visit request 1 has to be transported by vehicle from location 13 to its destination at location 6. Including another node visits thus create a saving off 284 in fleet operation cost and an increase of 132 in penalties, for a total saving of 152 in total. However, as mentioned earlier, increasing the number of node visits considerably complicates the problem, so it is beneficial to find and use the least number of node visits which does not restrict the practical problem too much.

Table 19: Varying the Number of NodeVisits 3

Instance	Best Solution	Best Bound	Percentage Gap
S2.4.5r (1)	2358.00	2358.00	0
S2.4.5r (2)	2358.00	2358.00	0
S2.4.5r (3)	-	1621.37	-
S3.5.5r (1)	2296.00	2296.00	0
S3.5.5r (2)	2144.00	2144.00	0
S3.5.5r (3)	2170.00	1484.08	32

In the instances mentioned above the number of transfer locations are four or five, but some instances with bigger fixed networks have also been tested. The bigger fixed network consisting of seven to nine transfer locations respectively and are illustrated in Appendix A.3. For the instances $B1.7.4$, $B2.8.4$ and $B3.9.4$, with four requests, setting the number of node visits to one allows for the problems to be solved within about 75 minutes. On the other hand, increasing the number of node visits to two, only the instance with seven transfer locations reaches the 20 percent gap in four hours. Thus, increasing the number of transfer locations makes the problem considerably more complicated to solve and increases the growth in complexity when increasing the number of node visits with one. In detail, the number of rows and columns in the problems with seven to nine transfer locations increases considerably (to about twice the size) when increasing the number of node visits from n to $n+1$. Furthermore, the reduction in rows and columns decreases when increasing the maximum number of node visits. As a consequence, when increasing the number of node visits the process of solving the instance becomes considerably more complicated.

That is, increasing the number of node visits from one to two nodes per location might increase the solution space and make it possible to find better solutions. This is because all feasible solutions for maximum n visits per location is also feasible for maximum $n+1$ visits per location. In addition, some solutions with $n+1$ visits per location is also included in the solution space. However, the improved solutions

are considerably more complicated to find and might not be possible to find in the time provided, and the resulting solution after the time runs out might be worse and maybe even unfeasible. Thus, it might be too difficult to find a better or even an as good solution as with one node visit if two node visits are allowed for the instance. In this case then, the importance is to limit the size of the problem so an acceptable solution can be found, instead of searching for the optimal practical solution and when the solution time limit run out being left with no solution at all.

The IDARP implementation considers not only changing the number of node visits for all transfer locations but also the option of setting the maximum number of node visits independently for the transfer locations. With this option it might be possible to keep the size of the problem small while not reduce the flexibility. For example, consider a setting where one or a few of the transfer locations are visited by all the requests while the rest of the transfer locations are visited at most one time. Instead of setting the number of visits to the same number as the number of requests, it is possible to set the number of maximum node visits to one for all transfer locations except for the one or few locations which needs the node visits to be at least the number of requests, and the resulting problem is smaller. Details of instance $R1.4.6$ run with node visit sets of [1111], [1133] and [2222] is provided in Table 20. The total time in seconds (and percentage gap) together with the rows and columns of the problem is stated. In addition, the best solution found in all cases where 2983.6. The number of original rows and columns are greatest with the individually based node visits, while after pre solving the number of rows and columns are greatest for the problem with two node visits per transfer location. This is because in this case the locations that have the highest number of node visits are used in more constraints creating a higher number of initial rows. On the other hand, it looks like these locations are included in more reduction techniques resulting in a greater reduction in set of constraints and thus eliminates a greater number of rows. The instance run with two node visits per transfer location does not manage to solve the problem to optimality and it is clear that the complexity of solving this problem is greater than for the two other options.

Table 20: Varying the Number of NodeVisits 4

Instance	Total Time (% Gap) (seconds)	Org Rows/Cols	Pre Rows/Cols
R1.4.6 [1111]	117.48	3952/1286	1422/1173
R1.4.6 [1133]	4966.78	5278/2688	2022/2511
R1.4.6 [2222]	14400 (3)	4782/2671	2053/2520

Thus, there is a trade-off when increasing or reducing the number of visits per location: Increasing the number of visits per transfer location might provide a better solution and the possibility of finding the optimal practical solution. However, increasing the number of node visits considerable complicate the problem of finding both feasible and improved solutions.

6.2.3 Varying Characteristics of Requests and the Vehicle Fleet

In addition to the problem features and the number of node visits, a few characteristics of the requests and the vehicle fleet might affect the complexity of solving the problem. The most important factors are the maximum service time for the request, the number of vehicles in the fleet and the time windows for pickup and drop off for the requests. This section discusses some important technical aspects of varying the maximum service time, the fleet size and the time windows for the requests.

First of all, increasing the *ErrorRate*, and thus the maximum service time of the requests, increases the complexity of the problem, especially if it allows for a greater use of the problem features. This is because additional arcs and thus additional solutions are feasible. For instance, the number of possible arcs which can be used increases and the number of arcs that are eliminated considering time window restrictions are reduced. Thus, several solutions have to be search through during the solution process and this complicates the solution process. As an example the details of the number of rows and columns in $S3.5.5$ with two node visits and *ErrorRate* 0.8 and 1.8 are given in Table 21. The increase in the problem size increases slightly, but since the pre solving manages to reduce the problem size less with an *ErrorRate* of 1.8 the resulting problem size is larger. In this example the number of rows are increased by 13 percent, while the number of columns increase by nine percent.

Table 21: Varying the Maximum Service Time of the Requests

Instance	Org. Rows/ Cols	Pre. Rows/ Cols	Percent Reduction
S3.5.5 (2)	3068/ 2286	1274/ 2140	58 / 6
S3.5.5r (2)	3127/ 2463	1435/ 2328	54 / 5
Percent Increase	*2/8*	*13/9*	

As evident in Table 15 (rows five and six, and rows eleven and 12), with instances $S1.4.5$ and $S3.5.5$, the number of rows and columns and the total solution time increases as the maximum service time is increased (the r represents that the *ErrorRate* is set to 1.8, instead of 0.8). This is because additional arcs and thus additional solutions are feasible. Thus several solutions have to be search through during the solution process and this complicates the solution process. On the other hand, increasing the maximum service time for the requests (and thus probably lowering the user convenience) might result in an optimal solution with a lower objective value as for example $S3.5.5$ (rows eleven and 12).

Increasing the size of the time windows for the requests increases the number of solutions in the solution space and makes it more complicated to find the optimal solution, but probably easier to find a solution. On the other hand, narrowing the time windows makes it possibly more difficult to find the feasible solutions, but the solution space is smaller so the number of solutions to search through before finding the optimal solution is reduced. Table 22 and 23 provide details for instance $H1.4.6$

incorporating all problem features. *stw* is added to the name of the instance if the width of the time windows are reduced to half their size, while *btw* is added to the name if the size of the time windows are doubled. The first column gives the best feasible solution, the second column gives the total solution time in seconds, while the two last columns provides the original rows and columns and the number of rows and columns after the pre solve, together with the resulting reduction during pre solve. It is evident from the table that narrowing the time windows actually reduces the time to completion, but that the resulting objective function is higher. For example, instance $H1.4.6$ with narrow time windows makes it possible to find the optimal solution in less than half the time than for the instance with regular time windows. Considering the rows and columns of the problem as well, the number of rows and columns decreases when narrowing the time windows. Furthermore, the pre solve manages to reduce the size of the problem to a greater extend than with regular time windows. On the other hand, increasing the size of the time window increases the solution time and the problem is not possible to solve to optimality within four hours. In addition, the effect of the pre solve is significantly reduced for the rows, from 71 percent with regular time windows to 48 percent with big time windows.

Table 22: Varying the Time Windows of the Requests 1

Instance	Best Solution	Total Time (% Gap) (seconds)
H1.4.6 (1)	3817.80	13383.43
H1.4.6stw (1)	4332.20	5056.80
H1.4.6btw (1)	3662.80	14400 (24

Table 23: Varying the Time Windows of the Requests 2

Instance	Org. Rows/Cols	Pre. Rows/Cols	Percentage Reduction
H1.4.6 (1)	4021/1546	1182/1430	71/8
H1.4.6stw (1)	3998/1498	979/1394	76/7
H1.4.6btw (1)	4083/1715	2110/1581	48/8

Varying the number of vehicles in the fleet affect the size of the problem, and by increasing the fleet size the problem size increases as well. For each vehicle in the fleet a set of binary variables are created specifying which arcs the vehicle are able to travel on. Thus, when increasing the number of vehicles in the fleet, the number of variables increases. In addition, the number of constraints increases, especially since some constraints are created for each vehicle in the fleet. Furthermore, the number of set of routes which fulfill all request and the number of symmetric solutions increases, thus increasing the solution space.

Increasing the load capacity of the vehicles might also affect the complexity of the problem. With increased capacity the number of arcs (binary variables) that

are possible to use for the vehicle and thus have to be created increases, and this increases the size of the problem. Additionally, the number of feasible routes increases with a higher load capacity, and the resulting solution time for the problem is often increased. The increase in rows and columns of the problem is evident for instance $R1.4.6$ (activating all problem features) in Table 24. The term m is added to the instance when the vehicle load capacity is increased from four to eight. The table states the best feasible solution found, the total solution time in seconds, and the original and pre solved rows and columns of the problem. Using vehicles with capacity of eight people create the need to increase the solution time with four to five times the solution time of the problem with regular capacity of four vehicles. On the other and, increasing the load capacity might lead to economical savings for the vehicle fleet.

Table 24: Varying the Load Capacity of the Vehicles

Instance	Best Solution	Total Time (seconds)	Org. Rows/Cols	Pre. Rows/Cols
R1.4.6 (1)	2983	117.48	3852/1286	1422/1173
R1.4.6m (1)	2915	502.83	3991/1635	1900/1586

To conclude, increasing the $ErrorRate$, and thus maximum service time, allow for a greater use of different problem features of transferring, multiple requests and walking, and thus increases the complexity of the problem and makes it more difficult to solve. Furthermore, varying the time windows might affect the complexity of the problem and narrower time windows decreases the solution space and makes it possible to solve the problem faster to optimality, but might make it more difficult to find the feasible solutions. Increasing the number of vehicles in the fleet and increasing the load capacity of the vehicles increases the size of the problem and as a consequence increases the time of finding the optimal solution.

6.2.4 The Size of the Test Instances

For the IDARP, the size of the fixed network and the maximum number of node visits at the transfer locations significantly affect the size of the solvable instances. These factors, as discussed earlier, considerably complicate the process of solving the problem, and this section considers the size of the problem the implementation manages to solve.

For the smaller fixed network with four or five transfer locations instances between three and 15 requests are solved. Details of the instances with five requests are provided in the Table 25. The table states the best solution, total solution time in seconds and rows and columns after pre solve. The maximum solution time is set to four hours (14400), and the optimal solution is found for all instances except for $S3.5.5(2)$ which has a resulting gap of nine percent after four hours. Table 25 illustrates that when increasing the number of node visits from one visit to two visits the solution time of the problem increases significantly. The number of rows

and columns are almost doubled when increasing the number of node visits from one to two. For example, $S3.5.5r$ increase from 759 rows to 1435 when increasing the number of node visits from one to two. Similarly, the solution time increases from 23 seconds to three hours and 15 minutes for instance $S3.5.5r$. Notice that the size of instance $S3.5.5$ which is the biggest of the small instances increases slightly more than the other instances when increasing the number of node visits from one to two. On the other hand, increasing the number of node visits from two to three a feasible solution is only found for instance $S3.5.5(3)$ within four hours.

Table 25: The Size of the Problem 1

Instance	Best Solution (% Gap)	Total Time (seconds)	Pre Rows/Cols
S1.4.5 (1)	2659.20	5.13	631/760
S1.4.5 (2)	2659.20	263.23	1116/1817
S1.4.5r (1)	2659.20	20.66	792/992
S1.4.5r (2)	2659.20	1298.90	1337/2057
S2.4.5 (1)	2358.00	9.65	618/582
S2.4.5 (2)	2358.00	2380.31	1188/1692
S2.4.5r (1)	2358.00	33.68	883/813
S2.4.5r (2)	2358.00	13190.16	1408/1856
S3.5.5 (1)	2435.00	4.33	482/519
S3.5.5 (2)	2435.00 (9)	14400.00	1274/2140
S3.5.5r (1)	2296.00	22.90	759/940
S3.5.5r (2)	2144.00	11523.94	1435/2328

The best feasible solution, bound and the resulting gap for the instances with seven and nine requests are given in Table 26. The instances with seven requests and one node visits are solved within 30 minutes, while if the instances are run with two node visits the resulting gaps are 16, 16 and 36 percent after four hours, respectively. For the smaller fixed network with four or five transfer locations Xpress manages to find feasible solutions for the instances with up to nine requests and one node visit within reasonable time. However, only one of these instances are within ten percent gap after four hours. If the number of node visits are increased from one visit to two visits, the software only manages to find a feasible solution to one of the instances $S3.5.9r$, and the resulting gap is at 40 percent after 12 hours. Increasing the number of requests to eleven and using fixed networks of four to five transfer locations ($S1.4$, $S2.4$ and $S3.5$) the software only managed to find a feasible solution for the instance with fixed network $S2.4$ after 12 hours. The resulting gaps are then 40 percent for instance $S2.4.11r$ and 50 percent for instance $S2.4.11$.

Considering the bigger fixed network with seven to nine transfer locations, one node visit, and four requests the implemented problem manages to solve the instances within 20 minutes. However, increasing the number of node visits from one to two for the transfer locations considerably complicates the solution process and the resulting gap is 17 percent gap for $B1.7(2)$ having seven transfer locations and

Table 26: The Size of the Problem (14400) 2

Instance	Best Solution	Best bound	Percent Gap (Time seconds)
S1.4.7 (1)	3052.00	3052.00	0 (164.22)
S1.4.7 (2)	3106.00	2603.85	16
S2.4.7 (1)	3105.20	3105.20	0 (197.21)
S2.4.7 (2)	3111.20	2610.16	16
S3.5.7 (1)	2538.00	2538.00	0 (1377.44)
S3.5.7 (2)	2665.00	1702.75	36
S1.4.9 (1)	4037.80	3291.25	18
S1.4.9r (1)	3709.60	3062.04	17
S2.4.9 (1)	3620.00	2733.36	24
S2.4.9r (1)	3450.00	2509.49	27
S3.5.9 (1)	2950.00	2659.03	10
S3.5.9r (1)	3086.00	2234.98	28

29 percent for instance $B3.9(2)$ with nine transfer locations. The solution time for the instances are still acceptable, and it is reasonable to assume that a few more requests can be added in the instances.

As mentioned earlier the instances with nine requests are the biggest instances which are solvable in acceptable time, if all problem features are activated, as they have 18, 24 and ten percent gap after 12 hours solution time. However, without these options a lot larger instances can be solved, and even the instances with 15 requests can be solved within 15 minutes without activating any of the three options. Thus, when including the options instances with up to nine request and five transfer locations (one node visit) can be solved in acceptable solution time, while without the problem features instances with a considerably greater number of visits can be solved.

To sum up, the implementation manages to solve instances with nine requests and five transfer locations (one node visit) if all problem features are activated. When increasing the number of node visits from one to two solvable instances can have up to seven requests. On the other hand, without activating problem features Xpress manages to solve problems with 15 requests in less than 15 minutes. Thus, including the problem features considerable complicate the problem, and makes it possible to solve small instances of the problem.

6.3 Economic Analysis

As already mentioned, including the problem features tend to create possible cost savings for operating the service, but at the same time usually lower the user convenience. In addition, including the possibilities of transferring to fixed network, transporting multiple requests in the same vehicles and letting the requests walk some part of their trip, considerably complicates the solution process, which might make it necessary for the operator to invest in more advanced decision support systems. This section looks at economical advantages and disadvantages for including the problem features. The analysis is based on results from solving several sets of instances and the different fixed networks and details of some of the instances are presented in Appendix A. The details consists of the different costs and penalties, and routes of the vehicles and trips for the requests are provided in Appendix A for the instances discussed in this section.

If only one of the problem features (transferring, multiple requests and walking) is to be incorporated into the problem, the possibility of transferring to other networks provides the greatest economical savings for the fleet. Thought, this possibility also creates a considerably drop in user convenience. The possibility of transporting multiple requests at the same time can also lead to reduction in the fleet costs since routes can be scheduled more efficiently. However, the penalties in this model do not consider the loss of convenience for the users of having to share the vehicle with other requests. The last problem characteristic, allowing the requests to walk some part of the trip, has minimal effect on the problem if only this possibility is activated, but can provide further savings in combination with other problem features. Since the distance the requests are able to walk is set on an individual basis, it is reasonable to assume that the drop in convenience is not significantly, because then the request can set the distance to zero. Activating all problem features in combination might create further savings, especially if the $ErrorRate$ is increased. However, combining several problem features, and especially if the $ErrorRate$ is increased, the drop in user convenience is similarly increased. Varying the characteristics of the requests and vehicle fleet might also affect the practical solution. For example, increasing the capacity of the vehicle might make it possible to take greater advantage of transporting several requests at the same time. However, most of the problem features and possibilities that creates flexibility for the operator and possible inconvenience for the users increase the solution space, and the operator might need more advanced software and knowledge to find the efficient solutions.

6.3.1 Transferring between Transportation Modes

Opening up for the possibility for the requests to transfer between vehicles or to a fixed network considerably complicate the process of finding the optimal solution, but might create possible lower objective values. In this section considerations are done both from the operator and drivers point of view and from the point of view of the users.

From the Operators and the Drivers Point of View:
As already mention the possibility of transferring between modes of transport can increase the flexibility of the service in that a vehicle does not have to transport a request all the way from its pickup to its drop off location. Thus, the operator has greater flexibility to create efficient routes for the vehicles. Considering the different problem features studied in this thesis, the possibility of transferring to different modes of transport clearly provides the greatest economical savings for the problem. This is evident from the results in Appendix A.2. For the instance $S1.4.5$ (Table 41) the percentage savings for the operation fleet is 0 including only walking, 16 including transporting multiple requests, 22 including transfers and 39 including all options in combination. Thus, considering including single problem features, the possibility of transfer provides the best economical savings of 22 percent. For the case with five requests and network $S3.5$ (Appendix A.2.3) the economical savings for the fleet is 28.6 percent, while including the penalties for user inconvenience the savings are reduced to 24 percent. Considering the trips of the vehicles, including the possibility of using a different mode of transport allow for request 2 to be transported by vehicle to transfer location 14, where the request is able to use the fixed route to its final destination. Furthermore, request 5 is able to use the fixed route to transfer location 14 where it is picked up by vehicle 2. However, notice that location 5 and location 14 are located at the same physical location. Thus, the benefits of traveling to location 14 is that the request can leave its pickup location without making a vehicle do a detour to visit location 5 within the time window for pickup. As a result it is possible to schedule more efficient routes for the vehicles.

The increased flexibility makes it possible for the operator to divide the area into zones and let each vehicle focus on one zone. The driver can increase their control of the characteristics in their zone. When the vehicles focus on smaller geographic zones the distance between the request and thus the time the vehicles drive empty should be less than if the requests are spread in bigger geographic area. If a request is suppose to travel across zones, it can be set off at a transfer location where another vehicle can pickup the request. Furthermore, if the zones are connected with a fixed route the vehicles does not have to drive all the way to the end of the zone but can set of the request at a transfer location and the request can use the fixed network into the other zone. As an example, consider instance $R1.4.6$ (Table 37 and Table 38 in Appendix A.1.3), where the network is quite sparse. Incorporating the problem characteristic, vehicle 1 can transport request 1 and request 2 which is located far from the other locations to a central location near drop off for location 1 and the transfer location 17. From there, request 2 is able to use the fixed route to another part of town and to its drop off location. Similarly, vehicle 2 can focus on the more populated area with several requests. Thus, vehicle 1 does not have to travel to the zone for vehicle 2 even though request 2 are going into this area.

The possibility of transfer makes it possible to let the demand responsive vehicles operate in dense and centralized areas while the consumers who are traveling to the less populated areas can travel by a fixed route. For example, $C1.7.3$ (Table

35 and Table 36 in Appendix A.1.2) illustrates a well established fixed network. Here the economical benefits for the operator and fleet is quite great, especially if the $ErrorRate$ is relaxed from 0.8 to 1.8 and the other problem features are incorporated as well. Then, the vehicles focus on transporting the requests in the city center and the requests going in or out of the city center use the fixed route. In detail, request 2 and request 3 walk to the fixed route and take the fixed route into the city center, at location 11. Here a vehicle picks up the requests and transport them to their final location within the city center. Furthermore, request 1 are driven to transfer location 9 where it takes the fixed route to its final destination. Equally, if a fixed network can transport the request into and within a central area, the vehicle fleet can focus on the more rural areas where requests are spread and it does not exists a well-established fixed network. If a request is going into the central area, a vehicle can transport the request to the nearest transfer location where they can use a fixed route into the central area.

From the drivers point of view the zone might create less variable work. In addition, since the drivers does not necessary transport the requests all the way, and probably transport a higher number of passengers per day, they might get less contact with their customers. The limited contact might be beneficial for some drivers, but might create a less satisfactory working environment for the other drivers.

From the Users Point of View:
For the point of view of the users transferring between mode of transport lowers their user convenience since they have to switch vehicles during the service. Some requests might even have difficulty when changing between modes of transport and will require assistance. Some might even be able to switch between modes of transport. For the instances with five requests and the network $S1.4$ (Table 41) including the penalties in the objective value the objective value is higher for the possibility with only transfers compared to the possibility with only transporting multiple requests at the same time, because of the high cost of penalties. For the instance $S3.5.5$ the penalties are also highest including the possibilities of transfers, but the cost reduction from operating the fleet is so high that they are neglectable in comparison.

On the other hand, some requests might not find the transfers that inconvenient. Especially, price sensitive customers might prefer the drop in convenience caused by transfers if this result in a lower overall cost of the service. Thus, the possibility of transfer lowers the convenience for the users. However, it is difficult to say to what extent since if the problem characteristic makes it possible to charge a lower price for the service, some customers might prefer the loss of convenience if the reduction in price is sufficient.

6.3.2 Transporting Multiple Requests at the Same Time

In a regular taxi service the vehicles transport one request at the time, except for a few cases when the requests themselves take the initiative to travel together. However, in the problem considered in this thesis the possibility of transporting multiple requests at the same time is incorporated. This section considers the advantages and the disadvantages for the operator, drivers and the users.

From the Operators and the Drivers Point of View:
As mentioned earlier transporting multiple requests at the same time can provide cost savings when scheduling the fleet. This is since the vehicle can pickup request located in the same area at the same time instead of going back and forth. Furthermore, it might reduce the time the vehicle has to drive empty between transporting requests. The instances with five requests and network $S1.4$ and $S3.5$ (Appendix A.2) illustrates these benefits. With network $S1.4$ vehicle 3 is able to pickup request 5 and then pickup and drop off request 4, before dropping off request 5 at its destination. For the instance with network $S3.5$, vehicle 3 manages to pickup request 4, then request 1, then dropping off request 4 before request 1. Thus, the vehicle is able to visit the locations in a more efficient route than if the vehicle has to transport the requests in turn.

Consider the case where there are two requests located at a similar pickup location (or drop off location). Using a regular taxi service the requests have to be transported in sequence, either by two separate vehicles or one vehicle traveling back and forth. The IDARP formulation allows one vehicle to pickup both requests and delivers them if the combined load does not exceed the capacity of the vehicle. For example, for instance $C1.7.3$ (Table 35 and Table 36 in Appendix A.1.2) these benefits are evident. With the possibility of transporting multiple requests at the same time, the vehicle can pickup request 2 and request 3 located at the same physical location, and transport it to their respective drop off locations. However, without this option two vehicles have to pickup the requests at the same physical location and transport them into the same area for drop off.

In the $R1.4.6$ (Table 37 and Table 38 in Appendix A.1.3), where the network is quite sparse, there are possible cost reductions to be made by transporting multiple requests in combination. In this case two of the requests, request 1 and request 2, are located far away from all the other requests and most of the fixed route. Then, a vehicle can pickup both requests at the same time and transport them into a more central area. This decreases the distance travelled and usage of vehicles compared to if the vehicles have to take two trips to the area. In addition, another vehicle can pickup request 3 and then pickup request 5 before dropping off request 3 and then request 5. Notice, that by using vehicles with greater load capacities the advantages of transporting multiple requests at the same time increases. For example, vehicle 2 uses a time of 296 time units with a capacity of four, but only 274 with a capacity of eight. This is because the vehicle can pickup request 3, then request 5, deliver request 3, then pickup request 6 and drop off the requests in turn. In this way the vehicle is able to finish up in one area before moving on to another area, and further savings occur. However, it is important to make sure

that the cost per head in the vehicle is lower using the multi vehicles compared to regular vehicles.

From the drivers point of view, some drivers might find it stressful to coordinate the seating and differing preferences for the different requests traveling together. For example, some requests might prefer a higher temperature than other requests. Thus, the driver might find it difficult to satisfy all users at the same time, and this can create a stressfull working environment for the drivers.

From the Users Point of View:
For the users, having to share a vehicle with another request might possible lower their satisfaction of the service. While a few requests might welcome the opportunity to meet new people, most users probably prefer the privacy of traveling alone. However, the loss in user convenience for sharing the vehicle with another request is not included in the objective value of this model. In addition, if they are going to different places, at least some of the requests experiences a longer service time if one allow for the possibility to transport requests together. This is because if two requests are to be transported together, the request that is picked up first has to travel to the other request's pickup location, and similarly, the last request to be dropped off has to visit the drop off location of the other request. For example, in instance $C1.7.3$ (Table 35 and Table 36 in Appendix A.1.2), request 2 has to sit in the vehicle while request 3 is picked up and dropped off, before request 2 arrive at their drop off location. Similarly, for instance $R1.4.6$ (Table 37 and Table 38 in Appendix A.1.3) and with a vehicle capacity of eight, request 5 have to sit in the vehicle while request 3 is dropped off and request 6 is picked up. Thus, request 5 have to travel with two other requests during their trip.

Some requests might prefer a lower service cost and sharing vehicle, than traveling alone at a higher cost. Similarly, if the different requests are going from or to the same location and know each other, they might prefer to be able to go together, and especially if this is a cheaper transportation alternative per head.

6.3.3 Walking some Part of the Trip

The integrated dial-a-ride formulation allow for the requests to decide for themselves have far they are able to walk. This allows for advantages and disadvantages for the service, which are discussed in this section.

From the Operators and the Drivers Point of View:
For the operator the possibility for the requests to walk some part of the trip allows for the scheduling of more efficient solutions. For example, a request can walk to a neighbor where they both can be picked up by a demand responsive vehicle, or walk from the buss stop to their final location.

The possibility of walking by itself has minimal effect on the problem because even though the request can walk to a location in the area nearby, it cannot be

picked up from this location as long as none of the other problem features are incorporated in the implementation. This is since the implementation does not allow a vehicle to visit a location without picking up or delivering the request associated with the location. As a consequence, the possibility of walking is only practical if one or both of the other problem features are activated. For example, by including all problem features request 1 in instance $S3.5.5$ is able to take itself from pickup to drop off with the help of the fixed route and by walking. Another example is given in instance $S3.5.7$ where request 3 is able to walk to pickup location of request 7, where both requests are picked up by vehicle 1.

In the instance $R1.4.6$ (Table 37 and Table 38 in Appendix A.1.3) two of the requests, request 1 and request 2, are located far away from all the other requests and most of the fixed route. Including the possibility of walking in combination with the other problem features allow for request 1 to walk to pickup location of request 2 where both request can be picked up by a vehicle and thus the vehicle does not have to pickup both request at their respective locations. Furthermore, in the instance $C1.7.3r$ (Table 35 and Table 36 in Appendix A.1.2) request 2 and 3 is located away from the rest of the location but close to a fixed route. Including the possibility of walking in combination with the other problem features and relaxed $ErrorRate$ allow for the requests to walk to the nearby transfer location 10, where they can take the fixed route into a more central location and be picked up by a vehicle located nearby. As a consequence, a vehicle does not have to travel to the area just to pick up the requests, and the fleet cost is reduced by 36 percent, from 1201 to 772.

From the Users Point of View:
From the point of view of the users having to walk some part of the trip lowers the user convenience. However, since the requests are allowed to specify their own maximum walking distance those who consider walking inconvenient can set the value to zero. Thus, the overall reduction of the user convenience should not be excessive, but it allows for the price sensitive customers or those who wants to walk for some other reason, to help to create efficient routes for the vehicles. For example, in the instance $C1.7.3r$ mentioned above. Here the savings in the operational fleet is 36 percent (from 1201 to 772) by letting request 2 and request 3 walk to their transfer location instead of having to be driven by a vehicle, but the savings in total is only one percent because the increase in penalties increases from 55.6 to 474.8.

6.3.4 Including all Three problem features in Combination

Further economical savings and synergy effects for the fleet can occur if all problem features are included when solving the problem. For example, a vehicle can pickup several request and drop them off at a transfer location, where they can use the fixed route to get as close as possible to their drop off location. Furthermore, if one of the drop off locations are within walking distance for the requests they are able to walk by themselves and a vehicle does not have to pick it up from the drop off location. This section considers the advantages and disadvantages for the operator,

driver and user of including all problem features.

From the Operators and the Drivers Point of View:
In an economical setting, considering either including none or all problem features, incorporating the features of transferring to a fixed network or between vehicles, transporting several requests at the same time and the possibility of walking, have great potential economical savings for the fleet. In fact, considering the two practical instances $C1.7.3$ (Table 35 and Table 36) and $R1.4.6$ (Table 37 and Table 38), some of the benefits discussed in the individual analysis only occur when combining the different options. Furthermore, for the instances with five requests $S1.4.5$ and $S3.5.5$ (Appendix A.2) including all problem features creates further savings compared to the economical benefits discussed in the individual sections. For $S1.4.5$ request 1 and request 2 are able to take advantage of using the fixed network while vehicle 2 is able to transport request 4 and request 5 in combination and thus reduce the distance travelled. For instance $S3.5.5$ including only the possibility of transfers (100) provide the same result as all three problem features (111). However, additional savings occur if the $ErrorRate$ is increased from 0.8 to 1.8. For example, relaxing the service time allow request 1 to transport itself from pickup to drop off by using the fixed route and walking, and the problem is then solvable with two vehicles. This decreases the fleet cost by 533 (23 percent) and increases the penalties by 242, for a total savings of 291 (12 percent).

Table 27 shows result from instance with seven requests and the networks $S1.4$, $S2.4$ and $S3.5$. Furthermore, the number in brackets represents the number of node visits, while the three last binary numbers represents which problem features are activated. The best solution and the total solution time in seconds are given in Table 27. For example, for instance $S1.4.7(1)$ the optimal solution is given both including no problem features (4700.00) and all problem features (3105.20), and the reduction in row three state the improvement in the objective value for including the problem features (in this case a saving of 34 percent).

Table 27: Economic Results 1

Instance	Best Solution	Total Time (seconds)
S1.4.7 (1) 000	4700.00	2.93
S1.4.7 (1) 111	3105.20	297.21
Percent Reduction	*34*	
S2.4.7 (1) 000	3454.00	1.04
S2.4.7 (1) 111	3052.00	264.22
Percent Reduction	*12*	
S3.5.7 (1) 000	3747.00	0.83
S3.5.7 (1) 111	2538.00	1377.44
Percent Reduction	*32*	

For the three instances with seven requests ($S1.4.7$, $S2.4.7$ and $S3.5.7$), the

economical savings including the problem features were 34, 12 and 32 percent in total, respectively (see Table 27). Furthermore, the details for the different cost and penalties for the instances with and without the problem features, are given in Appendix A.2. Without any of the problem features the vehicle fleet have to pickup the request and transport it to its drop off location right away. For the first case $S1.4.7$ activating the problem features makes it possible to use two vehicles instead of the original four vehicles which has to be used without the problem features. For example, request 5 are able to walk to request 4, and then both requests are picked up at location 4, transported to requests 4's drop off location before the vehicles travels to drop off request 5 at it destination location. Similarly, request 3 walks to pickup location for request 7 where a vehicle picks up both requests and deliver them at their respective drop off locations. Furthermore, the possibility of using the fixed route lets request 1 be able to take the fixed route from its pickup location and to a nearby transfer location in proximity of the drop off location for request 1. More importantly, one vehicle is able to pickup request 2 transport it to transfer location 17, set it off while picking up request 1, and then deliver request 1 at its drop off location, while request 2 uses the fixed route between location 17 and 16, where another vehicle picks it up and drops it off at its drop off location. Including all problem features creates economical saving of almost 40 percent for the fleet costs. However, an increase in penalties lowers the savings to 34 percent in total. Similar effects are seen for the other instances as well. For example instance $S2.4.7$, where a vehicle can pickup request 7 and then pickup and drop off request 3, before the vehicle picks up request 5 and deliver request 7 before it delivers request 5. In instance $S3.5.7$, request 3 walks to request 7, and at location 7 a vehicle picks up both requests and transport them to their respective drop off locations. Furthermore, a vehicle picks up request 2 transport it to transfer location 18, where it sets of request 2 and picks up request 5 and deliver request 5 at its drop off location, while request 2 are able to use the fixed route the last part of the trip. This causes economical savings for the operator when scheduling the fleet. The examples illustrates the possible economical savings from including the problem features, and the tables in Appendix A.2 show the detailed costs of the different model runs. On the other hand, the complexity of the problem increases considerably when incorporating the problem features. Thus, greater investments in software and expertise might be necessary for the operator to manage to find feasible and acceptable solutions. Similarly, the routes for the drivers will be more divided between different requests, and the driver will not necessary transport the request all the way to the final destination.

Similar effects are seen for the bigger instances with nine requests, and Table 28 shows the best solution found, the best bound and the resulting gap after 12 hours. $-$ represents that Xpress does not manage to find a feasible solution in the time provided (12 hours). The values are given for the three instances without problem features and including all problem features, and with one or two node visits for the transfer locations. For example, for $S3.5.9$, the best solution found without the possibility to transfer, transport multiple requests and for the requests to walk, is 4189. However, including the problem features, and setting the number of node visits to one, reduces the cost of the fleet to 3107. Notice that Xpress has

not managed to solve the instance to optimality within the time limit of 12 hours so even better solutions than 3107 might exist. Similarly, increasing the number of node visits to two allows for a greater use of the transfer locations and reduces the objective value to at least 3061. As evident from the table, the economical savings when increasing the number of requests are higher than for the smaller instances because a higher number of requests makes it possible to take advantage of the problem features and in this way schedule the vehicle routes more efficiently. For example, instead of just dropping of a request at a transfer location, with a higher number of requests, there might be a request waiting for pickup at the same transfer location or in the area nearby which can be picked up. Thus, more of the routes for the vehicles are used efficiently to transport requests. At the same time, the complexity of finding these efficient routes increases, thus increasing the investments in software and makes the work for the operator more difficult.

Table 28: Economic Results (43000) 2

Instance	Best Solution	Best Bound	Percent Gap
S1.4.9 (1) 000	5138.00	5138.00	0
S1.4.9 (1) 111	3887.40	3319.94	15
S1.4.9 (2) 111	-	2850.42	-
S2.4.9 (1) 000	4034.00	4034.00	0
S2.4.9 (1) 111	3620.00	2597.57	28
S2.4.9 (2) 111	-	2541.15	-
S3.5.9 (1) 000	4189.00	4189.00	0
S3.5.9 (1) 111	3107.00	2425.06	22
S3.5.9 (2) 111	3061.44	2008.20	34

From the Users Point of View:

It is however necessary to point out that even though the total costs decreases in all cases, the convenience for the user is decreased as well. In detail, they have to transfer between modes of transport, use fixed route networks, and travel together with other passengers and maybe walk some part of the trip. Furthermore, if a vehicle transport a request from pickup to drop off location the request can be sure that the vehicle has the responsibility of getting the request to its drop off location. However, when the trips are split into different parts the requests feel more of this responsibility themselves since they actively take part during transferring and walking. The $ErrorRate$ and penalties set an upper bound on the inconvenience for the user, and it is assumed that this level of convenience is acceptable for all users. However, in real cases some users are quite sensitive when it comes to a drop in user convenience and find the level unacceptable, while other users do not mind the drop in convenience and could possible even accept a lower service quality.

6.3.5 Varying the Load Capacity of the Vehicle

The load capacity of the vehicle is the number of passengers, bags and equipment the vehicle is able to transport. The vehicles considered in this problem are homogeneous and can transport four or eight passengers at the time. Here, the advantages and disadvantages of varying the load capacity for the transportation service and the requests are considered.

From the Operators and the Drivers Point of View:
For the operators point of view, it might be difficult to decide on the combination of vehicles in the fleet. Using a vehicle with a low load capacity can be beneficial when the requests are sparsely located, are small in load size and when it is not possible to take advantage of transporting multiple requests at the same time. This is because the fixed and variable costs of these vehicles are low. However, if the loads of the requests often exceeds the vehicle load capacity, or several trips have to be made to the same area, it might be economical benefits from using bigger vehicles, both considering the number of vehicles used and the total distance travelled by the vehicles. The bigger vehicles will probably have a higher fixed and variable cost, but if the vehicles are filled mostly to capacity the cost per head might be lower than for the smaller vehicles. Then again, if the big vehicles travels the same route and transport the same number of requests as the small vehicles, the cost per head is higher for the bigger vehicles.

Instances $R1.4.6$ is run both with a regular vehicle capacity of four passengers ($R1.4.6$) and with a vehicle capacity of eight passengers ($R1.4.6m$) (Table 37 and Table 38). Increasing the capacity of the vehicles allow for a vehicle to pickup request 6 when it is in the neighborhood of the request, instead of going back for the request after dropping off request 5. Thus, it is possible to lower the cost of the fleet since vehicle 2 are able to drive a shorter route. The decrease in fleet costs are 3.7 percent, with a resulting total objective value saving of 2.3 percent.

From the Users Point of View:
If multiple requests are transported in the same vehicle the issues for the user convenience considered in Section 6.3.2 apply and those users who prefer to travel alone might experience a lower user convenience. Increasing the load capacity of the vehicle increases the probability of transporting several requests together, which is beneficial for the fleet but probably not for the users. Consider again the case with $R1.4.6$ in Table 37 and Table 38 in Appendix A.1.3, both request 5 and request 6 experience a longer service time using vehicles with capacity of eight compared to the regular capacity of four passengers. In fact, the penalty for service time is increased by 14.4 percent. In detail, the service time for request 5 increases from 55 to 68, while the service time for request 6 increases from 46 to 49. However, for requests consisting of bigger parties traveling to and/or from similar or the same places the convenience might increase since they prefer to travel together. For price sensitive users, if it is possible to take the advantage of the extra room in the bigger vehicles so the per passenger cost is reduced, the use of multi taxies might be beneficial.

6.3.6 Varying the Time Windows of the Requests

Narrowing and relaxing the time windows for pickup and drop off of the requests affect the problem since with relaxed time windows the operator has more flexibility to schedule the requests as they wish and thus possible create more efficient and cheaper routes, while narrow time windows provide little flexibility when deciding the schedule of the requests. In the IDAR implementation the time windows for pickup is given in the instance and from this data the time window for drop off is calculated. Thus, if the initial time window is wide for the request the resulting final time window is wide. This section considers the effects of narrowing and relaxing the time windows for the operator, drivers and the users.

From the Operators and the Drivers Point of View:
With narrow time windows the vehicle have to pickup the request in the limited time window and this creates little flexibility for the operator when scheduling the routes. For example, if two requests are located at a similar area but they have narrow non-overlapping time windows it might not be possible to pickup the requests at the same time and as a result the vehicles might have to transport the requests separately which will increase the usage of the vehicles and the total distance traveled. However, if the time windows are relaxed the vehicle might be able to pick the requests up together and thus only have to visit the area once. Similarly, with wider time windows, the vehicles might be able to pickup requests in the same zone and thus finish up with this zone before moving on to another zone. However, with tighter time windows the vehicles will have to pinball across the different zones to fulfill the requests within the narrow time windows, and the resulting routes are often more expensive. As an example, consider instance $C1.7.3$ run with regular time windows $C1.7.3$ and narrow time windows $C1.7.3stw$. Details for these instances are given in Table 35 and Table 36 in Appendix A.1.2. For the instance $C1.7.3$ one vehicle is able to transport all three request, while narrowing the time windows require two vehicles to fulfill the same three requests. This increases the fixed and variable costs of the fleet with a total of 5 percent. Thus, it is evident that the operator benefits from relaxed time windows. Furthermore, instance $H1.4.6$, which details of the objective value and the practical solution is given in Appendix A.1.1, is run with regular, narrow and big time windows for the requests. For the instances run without the problem features and varying the size of the time windows, narrowing the time windows forces the use of one more vehicle to find a feasible solution. On the other hand, relaxing the time windows for the requests makes it possible to schedule routes for the vehicles that are more efficient and thus travels a shorter total distance than with regular time windows. If the problem features are incorporated when solving the problem, varying the size of the time windows has an increased effect on the economical solution. Narrowing the size of the time windows increases the distance traveled by the vehicles because they have to make detours so that every request is picked up (dropped off) during the time window for pickup (drop off). On the other hand, increasing the size of the time windows makes it possible for the operator to schedule more efficient routes for the vehicles. In fact, even though the problem with big time windows are not solved to optimality, the upper bound (3662.80) is better than the optimal solution (3817.80) for the instance with regular time windows, and only two vehi-

cles are needed to fulfill all requests. However, the gap between upper and lower bound for the instance with big time windows is 24 percent after four hours and it is possible that the optimal solution has a lower cost than the upper bound of 3662.80.

From the drivers point of view, loose time windows allow for drivers to be less concerned about delays, since the requests do not know when they are being picked up during the request anyway. On the other hand, if the time windows are narrow, this allows little room for delay during the route.

From the Users Point of View:
The factor of waiting for pickup is not considered in the objective value and it is therefore not possible to quantify the loss of user convenience. However, it is quite understandable that the user prefers a more narrow time windows. The requests can be picked up at any time during the time windows and have to be ready during the whole interval. If this interval is wide, say for example from midmorning to midday, they have to be home and ready during the period, since they do not known when they are going to be picked up. On the other hand, if they know that they will be picked up midday, they can use the morning as they wish to do other activities. Thus, wider time windows lower the convenience for the users. Thus, for instance $H1.4.6$ the penalties decrease from 342.80 with regular time windows to 220.80 for the instance with narrow time windows (Table 31 and Table 32 in Appendix A.1.1). However, if the driver is able to call in advance, say half an hour before picking up the request, the loss of convenience can be reduced since the passengers know that they will not be picked up before they get the call.

Furthermore, some requests might prefer to be picked up as fast as possible because they have to leave their pickup location. Others might need to be at their drop off location at certain time because they for example have an appointment they have to make. These considerations need to be maintained and in the implementation version of the problem it can only be considered by calculating the possible pickup times. Thus, in sum the requests need to be able to communicate their preferred time windows and it might be preferable to update to a more specified time when the schedule is set. Since the IDAR formulation does not allow the drivers and requests to communicate, narrowing the time windows increase the quality of the service compared to looser time windows.

6.3.7 Varying the Maximum Service Time

Increasing the $ErrorRate$, and thus allowing a longer service time for the requests, the operator has more flexibility to schedule the requests as they wish and thus possibly create more efficient and cheaper routes for the vehicles. In the implementation an $ErrorRate$, E, is specified in the data instance and this scalar is used to calculate the maximum service time $S_r^{max} = (1 + E)T_{r,r+\overline{r}}$ for request r. Thus, a larger $ErrorRate$ results in a larger maximum service time for the requests. This section considers the effects of varying the maximum service time for the operator, drivers and the users.

From the Operators and the Drivers Point of View:
Allowing for a larger $ErrorRate$, and thus service time, creates flexibility for the operator when scheduling the routes of the vehicles and makes it possible to use slower modes of transport, as for example using the fixed routes or walking. This might create savings for the vehicle fleet because they will need to transport the request a shorter part of their trip.

Traveling by vehicle is the fastest mode of transport, using a fixed network is a slightly slower mode of transport, while walking is the slowest mode of transport. Furthermore, some deviations from the direct travel route usually occur if a request walks or uses a fixed route. Thus, if the $ErrorRate$ is set low, many of the possible solutions considering using a fixed route or the walking violates the maximum service time of a request and is thus infeasible. If the maximum service time is increased it is possible to take greater advantage of the possibilities of the problem features, and with it lowering the objective value. Though, this will come at a cost of decreased user convenience since the service time is increased. Details of the instances $S2.4.5$ and $S3.5.5$ are provided in Table 29 and Table 30. With an $ErrorRate$ of 0.8 the optimum solution value for $S3.5.5(1)$ is higher (2435.00) than the optimum solution value with an $ErrorRate$ of 1.8 (2296.00). Increasing the $ErrorRate$ thus allow for a economic saving of six percent for the objective value. For instance $S1.4.5$ the effects of increasing the $ErrorRate$ is only evident when allowing for two node visits at the transfer locations. For a solution time of four hours the problems have not been solved to optimality, but since the optimistic bound on the solution with $ErrorRate = 0.8$ is higher than the pessimistic bound on the solution with $ErrorRate = 1.8$ it is reasonably to assume that increasing the $ErrorRate$ creates economical benefits for the fleet.

Table 29: Varying the Maximum Service Time 1

Instance	Best Solution	Best Bound	Percent Gap
S2.4.5 (1)	2358.00	2358.00	0
S2.4.5r (1)	2358.00	2358.00	0
S2.4.5 (2)	2106.33	1704.07	19
S2.4.5r (2)	1659.66	1621.37	2

Table 30: Varying the Maximum Service Time 2

Instance	Best Solution	Best Bound	Percent Gap
S3.5.5 (1)	2435.00	2435.00	0
S3.5.5r (1)	2296.00	2296.00	0
S3.5.5 (2)	2435.00	2213.38	9
S3.5.5r (2)	2144.00	2144.00	0

Considering the details off the instances provided in Table 47 to Table 50 in Appendix A.2, with an $ErrorRate$ of 0.8 the operator has to use three vehicles to

fulfill all requests, while if the $ErrorRate$ is relaxed only two vehicles are needed to fulfill all requests since requests.

From the Users Point of View:
On the other hand, relaxing the service time of the requests reduces the convenience for the users. A longer maximum service time increase the average service time for the requests, as they are able to use slower modes of transport or travel longer detours. However, price sensitive requests might prefer a longer service time and a cheaper service price, and then relaxing the service time might be appropriate. To illustrate the effect of relaxing the service time the result from running instance $S3.5.5$ with an $ErrorRate = 0.8$ ($S3.5.5$) and an $ErrorRate = 1.8$ ($S3.5.5r$) are provided in Table 47 to Table 50 in Appendix A.2. Here it is evident that increasing the maximum service time allow for reducing the objective value and finding more efficient trips for the operation fleet. However, the penalties paid as a cause of loss in convenience for the user increases significantly. For example, in instance $S3.5.5$ increasing the $ErrorRate$ increased the penalties from 144 to 386. With an $ErrorRate = 1.8$ request 1 manages to use the fixed network the first part of the trip and walk the final part of the trip.

For some requests and especially those who have problems when transferring between modes of transport the loss in convenience might be so great that they chose to take their business elsewhere. However, some requests are more price sensitive and might prefer a few transfers and a longer service time if this makes the cost of the service is lower.

6.3.8 The Process of Finding an Acceptable Solution

In the previous sections different characteristics for the problem and how they affect the operator, drivers and users are considered. These sections assume that it is possible to find the solutions within reasonable time and that the cost of finding the different solutions are equal. However, this might not always be the case and this section considers the process of finding the solutions. As mentioned in the Section 6.2 the process of finding a solution to the problem is considerably complicated by adding;

- The possibility of transferring to fixed network and between vehicles
- The possibility for the requests to walk some part of the trip
- Increasing the maximum number of node visits

The problem is further complicated by increasing the $ErrorRate$, and thus maximum service time, and slightly increased when increasing the number of vehicles in the fleet. Furthermore, increasing the size of the time windows for the request increases the number of solutions in the solution space and makes it more complicated to find the optimal solution, but probably easier to find a solution. On the other hand, narrowing the time windows makes it more difficult to find feasible solution, but the solution space is reduced. Thus, in addition to the trade

off between cost and convenience there is a trade off between cost savings for the operational fleet and an increase in cost of finding an acceptable solution.

In detail, by incorporating the complicating options and characteristics in the problem the process of finding the optimal solution get more complicated, and as a consequence the operator will require a more advanced decision support system. Thus, there is a need for a greater investment in IT structure, and the operators work is more advanced and expensive to perform. This is because it has to be possible to find an acceptable solution within the time from the last request has been made and before the vehicles have to leave the depot and start their route.

7 Conclusion

In this thesis an integrated dial-a-ride model has been presented. The integrated dial-a-ride problem is a version of the pickup and delivery problem where the load represents a number of people. The model has to fulfill a set of requests by scheduling a vehicle fleet. In this case, some part of the request can be performed by a fixed route service and the requests which are able to can walk some part of the service. At the same time the goal of the model is to minimize fleet operating costs and user inconvenience.

Different solution methods have been proposed to solve the problem in the literature. The main exact solution methods are branch-and-bound and column generation methods. Metaheuristics have also been proposed to solve bigger instances of the problem, where a spectrum of solution methods are tried: Different types of construction and insertion based method, local search based methods (for instance simulated annealing and tabu search), and other types as for instance genetic algorithms and combinational methods. In addition, several different problem features have been discussed in the literature. The model presented in this thesis is a static problem considering minimizing the cost of the fleet and the penalties for user convenience. The vehicle fleet is homogeneous, centered at a depot and contains a fixed number of vehicles that can be used. Each vehicle has a specific usage and load capacity. The requests specifies a pickup and drop off location, a time window for pickup, and a load to be transported together with a maximum walking distance. The requests can be transported by vehicles, transfer between vehicles and to a fixed network, and walk some part of the trip.

7.1 Technical Findings

An arc-based formulation for the IDARP that minimizes the costs of operating the vehicles and the user inconvenience is given, and several ways of reducing the solution space is discussed. The effect of the reduction techniques are evaluated and the arc elimination constraints, symmetry breaking constraints and subtour elimination constraints seem to reduce the complexity of the problem the most. In addition, different problem features (transferring between modes of transport, transport multiple requests at the same time, allowing for the requests to walk some part of the trip) are incorporated into the problem and considerably complicates the solution process. Some other characteristics of the problem might also have an effect on the solution complexity. For example, increasing the $ErrorRate$ and thus the maximum service time allow for a greater use of the option of transfers, multiple requests and walking and thus increases the size of the problem. The problem is formulated so the operator can set the maximum number of node visits at each transfer location. Each node can be visited once, so increasing the number of node visits per transfer location can allow for better economical solutions. However, increasing the number of node visits from n to $n + 1$ increases the size of the problem considerably. Thus, setting the number of node visits too small might cut off a feasible solution, while setting the number too large might make

the problem too difficult to solve. Since often a few node visits are enough and the operator might end up with a worse solution with too many node visits since there is not enough time to search through the entire solution space, setting the number of node visits to two seems to work well in the instances tested in this thesis.

The NP-hard problem has been solved using Xpress IVE optimization suite, and manages to solve small sized problem. Problems with one node visit, five transfer locations and nine requests are solved within reasonable time. Considering two node visits, problems with five transfer locations and seven requests are solvable within reasonable time.

7.2 Economical Findings

The problem features of transferring between modes of transport, transporting multiple requests at the same time and walking create possible economical savings regarding the cost of operating the fleet but might increase the difficulty of finding good solutions and decrease the user convenience. Considering the problem features individually, the possibility of transferring clearly creates the best cost savings. However, this possibility often also lead to the greatest loss in user convenience. The possibility of transporting multiple requests at the same time can also lead to reduction in the fleet costs since routes can be scheduled more efficiently. However, the penalties do not consider the loss of convenience for the users of having to share the vehicle with other requests. Alone, the possibility of walking does not significantly affect the solution of the problem. Including the problem features in combination might create synergy effects and cost reductions as the operator can take advantage of different options in combination. Relaxing the restriction of the service time and increasing the time windows of the requests, might increase the use of the problem features (transferring, multiple requests, walking) which lowers the operation fleet costs and at the same time the convenience for the user. In addition, increasing the capacity of the vehicle might make it possible to take greater advantage of transporting several requests at the same time. However, most of the characteristics which creates flexibility for the operator and possible inconvenience for the users, increases the solution space, and the operator might need more advanced software and knowledge to find the efficient solutions.

To sum up, the possibilities of transferring between modes of transport, transporting multiple requests at the same time and allowing for the requests to walk some part of the trip, can create savings for the demand responsive vehicle fleet. However, these problem features complicates the solution process of the problem and might require more investments and expertise to find good solutions. In addition, the user convenience is reduced as a consequence of transfers, increased service time, less privacy during the trip and even walking some part of their trip. The goal is therefore to find a service where it is possible to schedule the routes within the time and cost provided, which is as efficient for the vehicle fleet as possible while still maintaining a certain level of user convenience.

8 Future Research

The main technical and implementation issues to consider further can be divided into (1) creating a more efficient formulation and a more sophisticated solution approach to be able to solve bigger instances in reasonable time, and (2) making the problem more realistic and applicable for real life.

8.1 Technical Issues

Several aspects can be considered in future work on the technical and implementation issues. These can be divided into (1) how to reduce the size and complexity of the model and (2) implementing an efficient algorithm (exact or heuristic) to solve the problem.

8.1.1 Reducing the Size of the Problem

This thesis have focused on reducing the solution space by including among others valid inequalities. However, it might still be possible to reduce the gap between the LP and IP solution. For example, the cost of operating the fleet in the LP solution is lower than the fleet cost in the IP solution because of the use of fractional arcs in the LP solution. Another factor to consider is limiting the number of node visits at the transfer locations while at the same time trying not to cut away the optimal solution. For example, it is possible to create an algorithm to set the maximum number of node visits at the transfer locations. As already discussed, as the instance size increases the probability that a transfer location is visited by each request is small, while the significant increase in nodes complicates the process of finding good solutions. In the model presented in this thesis the maximum number of visits is given explicitly in the problem formulation which makes it easy to change the number. Furthermore, the implementation allow for individually based node visits for those locations which are more central and need several visits. However, at a later stage an algorithm can be created that calculate the appropriate number of maximum visits given the number of requests, the size of the fixed network, and the density of requests located around a transfer location.

8.1.2 Applying a more Advanced Solution Method

When solving the model in this thesis the Xpress IVE optimization suite has been used, however many studies have been done on more custom-made solution approaches, both exact methods and heuristics. An example on an exact method could be a column generation approach. The model can be reformulated into a set partitioning problem consisting of feasible routes which are chosen to satisfy all request. Instead of creating all routes and solving the problem, the problem can be solved with a subset of routes and new routes can be added when needed. The columns (routes) can be created by a dynamic programming method and be added during a column generation approach. Another option is to create the routes by an

approximated method. The model can also be solved by metaheuristics. Several methods, for example local search methods (tabu search and simulated annealing) seem to provide quite good results. One of these methods can be modified to solve the IDARP presented in this thesis.

8.2 Economical Issues

To make the model more applicable for real life setting several factors can be studied and a few of these are mentioned below. First of all, the model presented considers three factors of user inconvenience and assumes that all passengers value these factors equally. The model would be more realistic if the customers themselves where able to set a value for the penalties corresponding to the factors of user convenience. For example, all customers have a certain amount of points (for instance 100) to divide among the factors, and can thus give higher penalty to those factors that are important for them.

For price sensitive customers it should be possible to consider the value of lower service cost for the requests, and this factor can be valued similar as the three existing factors. The cost of the service can be set in many different ways and one is provided here. Firstly, a base cost for providing the service is considered. Next, a cost corresponding to the direct distance between pickup and drop off location is added. Then, a fixed cost for each transfer during the trip and/or a part of the cost corresponding to the distance saved by taking a fixed route can be subtracted. Furthermore, a cost can be subtracted if there is another request in the vehicle during a part of the service. Determining the costs corresponding to the different elements might be difficult to do, and as a minimum it is necessary to make sure that the demand responsive service has enough income to cover all costs of operating the fleet and providing the fixed route service.

To balance costs and convenience an algorithm comparing cost savings and loss of user convenience can be used. This can be done by creating evaluation functions and add these as restrictions in the problem instead of penalizing them in the objective function. Some examples of factors to consider are: (1) Allow an increase in service time if this lowers the cost in a special manner, (2) if there are more transfers than two a certain cost saving should be achieved, and (3) if the request has to share the vehicle with another request there should be a certain amount of cost savings. The details of these evaluation functions might be difficult to find and estimate, and is therefore not considered here. Furthermore, there might also be appropriate to have an upper bound on the factors measuring the user convenience so to make sure that the quality of the service cannot fall below a lower bound.

The fixed route is assumed to be the same mode of transport and it is assumed that the fixed route always is a slower way of traveling than using a vehicle. However, a more realistic approach could be to consider different fixed network with different velocities. For example, there can be great benefits from using a fast velocity train between different cities. It can also be assume that the cost of trav-

eling with the fixed route network is covered by the demand responsive service, so the cost of using the networks has to be considered in the objective function. Furthermore, these costs can be individual for the different networks. In addition, the fixed route networks are assumed to have so frequent departures that a request is able to depart as soon as it arrives at the transfer location. A more realistic assumption is to include a time table for the departure of the fixed network, or assume an expected waiting time at the transfer location. The time table and/or expected waiting time could also be individual for the respective networks.

The vehicle fleet is assumed homogeneous and it is assumed a constant velocity on the trips. However, it could have been interesting to implement a heterogeneous fleet, and maybe several depots for the fleet. If the fleet is heterogeneous some of the vehicles can have special facilities, as for instance extra room for baggage, special access for wheelchairs, or just room for large groups of passengers. If the vehicles are located in several areas it can be taken advantage of the fixed route to connect the areas (for example dense areas with less dense areas). Furthermore, it can be included request loads which are higher than the capacity of a vehicle. This can for example be modeled by splitting the load in two new requests until the new requests have a load less than the capacity of a vehicle. Another possibility is to allow for split loads so the operator can schedule more efficient routes without considering being restricted to pickup the whole load at one stop. The operator has to make sure that the whole load of the request is transported from their pickup to drop off location.

References

M. Aldaihani and M. Dessouky. Hybrid scheduling methods for paratransit operations. *Computers and Industrial Engineering*, 45:73–96, 2003.

A. Alshamrani, K. Mathur, and R.H. Ballou. Reverse logistics: simultaneous design of delivery routes and return strategies. *Computer and Operation Research*, 34: 595–619, 2007.

D. Applegate, R. Bixby, V. Chvátal, and W. Cook. Finding cuts in the tsp (a preliminary report). *DIMACS Technical Report*, 95-05, 1995.

D. Applegate, R. Bixby, V. Chvátal, and W. Cook. On the solution of traveling salesman problems. *Documenta Mathematica, Extra volume ICM*, 3:645–656, 1998.

D.L. Applegate, R.E. Bixby, V. Chvátal, and W.J. Cook. *The traveling salesman problem: A computational study.* Princeton University Press, 2006.

A. Attanasio, J.-F. Cordeau, G. Ghiani, and G. Laporte. Parallel tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem. *Parallel computing*, 30(3):377–387, 2004.

E.K. Baker. An exact algorithm for the time-constrained traveling salesman problem. *Operations Research*, 31(5):938–945, 1983.

R. Bent and P. Van Hentenryck. A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows. *Computer and Operation Research*, 33:875–893, 2006.

G. Berbeglina, J.F. Cordeau, I. Gribkovskaia, and G. Laporte. Static pickup and delivery problems: A classification scheme and survey. *TOP (2007)*, 15:1–31, 2007.

K. Bergvinsdottir, J. Larsen, and R. Jørgensen. Solving the dial-a-ride problem using genetic algorithms. Imm-technical report-2004-20, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, 2004.

L.-P. Bigras, M. Gamache, and G. Savard. The time-dependent traveling salesman problem and single machine scheduling problems with sequence dependent setup times. *Discrete Optimization*, 5:685–699, 2008.

O. Bräysy. A reactive variable neighborhood search for the vehicle routing problem with time windows. *INFORMS Journal on Computing*, 15:347–368, 2003.

O. Bräysy and M. Gendreau. Tabu search heuristic for the vehicle routing problem with time windows. *TOP*, 10(2):211–237, 2002.

A. Chabrier. Vehicle routing problem with elementary shortest path based column generation. *Computers and Operations Research*, 33(10):2972–2990, 2006.

S. Chan. *Metaheuristics for solving the dial-a-ride problem.* PhD thesis, North Carolina State University, USA, 2004.

E. Choi and D.-W. Tcha. A column generation approach to the heterogeneous fleet vehicle routing problem. *Computers and Operations Research*, 34(7):2080–2095, 2007.

C. Christiansen and J. Lysgaard. A column generation approach to the capacitated vehicle routing problem with stochastic demands. Technical report, Aarhus School of Business, Department of Business Studies, 2006.

M. Christiansen. *Inventory and time constrained ship routing: A mathematical programming approach*. PhD thesis, NTNU, Norway, 1996.

J.-F. Cordeau. A branch-and-cut algorithm for the dial-a-ride problem. *Operational Research*, 54:573–586, 2003.

J.-F. Cordeau and Salazar González J.J. Iori M., Laporte G. A branch-and-cut algorithm for the pickup and delivery traveling salesman problem with lifo loading. *Networks*, 51(1):46–59, 2010.

J.-F. Cordeau and G. Laporte. A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research Part B*, 37:579–594, 2002.

J.-F. Cordeau, G. Desaulniers, J. Desrosiers, M.M. Solomon, and F. Soumis. The vrp with time windows. In P. Toth and D. Vigo, editors, *The Vehicle Routing Problem*, volume 9, chapter 7, pages 157–193. Society for Industrial and Applied Mathematics., 2002a.

J.-F. Cordeau, M. Gendreau, G. Laporte, J.-Y. Potvin, and F. Semet. A guide to vehicle routing heuristics. *Journal of the Operational Research Society*, 53: 512–522, 2002b.

C.E. Cortès, M. Matamala, and C. Contardo. The pickup and delivery problem with transfers: Formulation and a branch-and-cut solution method. *European Journal of Operational Research*, 200(3):711–724, 2010.

L. Coslovich, R. Pesenti, and W. Ukovich. A two-phase insertion technique of unexpected customers for a dynamic dial-a-ride problem. *European Journal of Operation Research*, 175:1605–1615, 2006.

T.G. Crainic, F. Malucelli, M. Nonato, and F.C. Guertin. Meta-heuristics for a class of demand responsive transit systems. *INFORMS Journal on Computing*, 17(1):10–24, 2005.

M. Cremers, W. Haneveld, and M. van der Vlerk. A two-stage model for a day-ahead paratransit planning problem. *Mathematical Methods of Operations Research*, 69:323–341, 2008.

Fausto Errico, Teodor Gabriel Crainic, Federico Malucelli, and Maddalena Nonato. A survey on planning semi-flexible transit systems: Methodological issues and a unifying framework. *Transportation Research Part C*, 36:324–338, 2013.

M. Gendreau, F. Guertin, J.-Y. Potvin, and R. Séguin. Neighborhood search heuristics for a dynamic vehicle dispatching problem with pickups and deliveries. *Transportation Research Part C*, 14(157-174), 2006.

B. Golden, S. Raghavan, and E. Wasil. *The vehicle routing problem: Last advantages and new challenges*. Operations Research/Computer Science Interfaces, 2008.

I. Gribkovskaia, Ø. Halskau, G. Laporte, and M. Vlcek. General solutions to the single vehicle routing problem with pickups and deliveries. *European Journal of Operation Research*, 180:568–584, 2007.

C. Häll, H. Anderson, J. Lundgren, and P. Värbrand. The integrated dial-a-ride problem. *Public Transport*, 1(1):39–54, 2009.

H. Hernándes-Pérez and J.-J. Salazar-González. A branch-and-cut algorithm for a traveling salesman problem with pickup and delivery. *Discrete Applied Mathematics*, 145:126–139, 2004a.

H. Hernándes-Pérez and J.-J. Salazar-González. Heuristics for the one-commodity pickup-and-delivery traveling salesman problem. *Transportation Science*, 38(2): 245–255, 2004b.

H. Hernándes-Pérez and J.-J. Salazar-González. The one-commodity pickup-and-delivery traveling salesman problem: Inequalities and algorithms. *Networks*, 50 (4):258–272, 2007.

M. Hickman and K. Blume. Modeling cost and passenger level of service for integrated transit service. In Stefan Voß and Joachim R. Daduna, editors, *Computer-Aided Scheduling of Public Transport*, volume 505, chapter 3, pages 233–251. Lecture Notes in Economics and Mathematical System, 2001.

M.E.T. Horn. Fleet scheduling and dispatching for demand-responsive passenger services. *Transportation Research Part C*, 10:35–63, 2002.

M.E.T. Horn. Procedures for planning multi-leg journeys with fixed-route and demand-responsive passenger transport services. *Transportation Research Part C*, 12:33–55, 2004.

M.I. Hosny and C.L. Munford. The single vehicle pickup and delivery problem with time windows: Intelligent operators for heuristic and metaheuristic algorithms. *Journal of Heuristics*, 16(3):417–439, 2010.

T.-Y. Hu and C.-P. Chang. Exact algorithm for dial-a-ride problems with time-dependent travel cost. *Eastern Asia Society for Transportation Studies*, 9:1–18, 2013.

J.-J. Jaw, A.R. Odoni, H.N. Psaraftis, and N.H.M. Wilson. A heuristic algorithm for the multi-vehicle advance-request dial-a-ride problem with time windows. *Transportation Research*, 8 20B(3):243–257, 1984.

H.L.M. Kerivin, M. Lacroix, A.R. Mahjoub, and A. Quilliot. The splittable pickup and delivery problem with reloads. *European Journal of Industrial Engineering*, 2(2):112–133, 2008.

G. Laporte. The traveling salesman problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(2):231–247, 1992.

C.F. Liaw, C.C. White, and J.L. Bander. A decision support system for the bimodal dial-a-ride problem. *IEEE Transportations on Systems, Man, and Cybernetics*, 26(5):552–565, 1996.

R. Masson, F. Lehuédé, and O. Péton. The dial-a-ride problem with transfers. *Computers and Operations Research*, 41:12–23, 2014.

G.R. Mauri and L.A.N. Lorena. A multiobjective model and simulated annealing approach for a dial-a-ride problem. *Analysis, Design, and Evaluation of Human-Machine Systems*, 11(1):256–268, 2006.

C. Mues and S. Pickl. Transshipment and time windows in vehicle routing. *International Symposium on Parallel Architectures, Algorithms and Networks*, pages 113–119, 2005.

Y. Nakao and H. Nagamochi. Worst case analysis for pickup and delivery problems with transfer. *Communications and Computer Sciences*, E91-A(9):2328–2334, 2010.

P. Oertel. *Routing with reloads*. PhD thesis, University of Cologne, 2000.

G. Pankratz. A grouping genetic algorithm for the pickup and delivery problem with time windows. *OR Spectrum*, 27:21–41, 2005.

Julie Paquette, Jean-Francois Cordeau, and Gilbert Laporte. Quality of service in dial-a-ride operations. *Transportation*, 39(3):539–564, 2012.

S. N. Parragh, K.F. Doerner, and R.F. Hartl. A survey on pickup and delivery problems: Part i: Transportation between customers and depot. *Journal fuer Betriebswirtschaft*, 58:21–51, 2008a.

S. N. Parragh, K.F. Doerner, and R.F. Hartl. A survey on pickup and delivery problems: Part ii: Transportation between pickup and delivery locations. *Journal fuer Betriebswirtschaft*, 58:81–117, 2008b.

S. N. Parragh, K.F. Doerner, and R.F. Hartl. Variable neighborhood search for the dial-a-ride problem. *Computers and Operations Research*, 37(1129-1138), 2010.

D. Pisiger and S. Ropke. A general heuristic for vehicle routing problems. *Computers and Operations Research*, 34(8):2403–2435, 2007.

A. Rais, F. Alvelos, and M.S. Carvalho. New mixed integer-programming model for the pickup-and-delivery problem with transshipment. *European Journal of Operational Research*, 235:530–539, 2013.

J. Renaud, F.F. Boctor, and G. Laporte. Perturbation heuristics for the pickup and delivery traveling salesman problem. *Computers Operational Research*, 29: 1129–1141, 2002.

S. Ropke and J.-F. Cordeau. Branch-and-cut-and-price for the pickup and delivery problem with time windows. *Transportation Science*, 40(5):455–472, 2008.

S. Ropke, J.-F. Cordeau, and G. Laporte. Models and branch-and-cut algorithms for pickup and delivery problems with time windows. *Networks*, 49:258–272, 2007.

M. W. P. Savelsbergh and M. Sol. The general pickup and delivery problem. *Transportation Science*, 29(1):17–29, 1995.

Stefan Steinerberger. New bounds for the traveling salesman problem. *Advances in Applied Probability*, 47, 2015.

State of California Public Utilities Commission Transportation License Section. Basic information for transportation network companies and appliactions.

X. Wang, M.M. Dessouky, and F. Ordonez. A pickup and delivery problem for ridesharing considering congestion. submitted for publication, 2015.

A Instances and Networks

This section provides the fixed networks for the important instances run. Furthermore, some details and results of some specific instances are given. First, a few case specific instances are discussed in Section A.1, then the small fixed network with four to five transfer locations are illustrated in Section A.2, before the bigger instances of seven to nine transfer locations are illustrated in Section A.3.

A.1 Case Specific Networks

A few specific case instances are run to study the advantages and disadvantages of incorporating the different problem features and some additional characteristics of the fleet and requests. The details of these networks and some of the results are given in this section.

A.1.1 Hospital Case

The hospital case $H1.4.6$ illustrates situations where the majority of the passengers are traveling to a common area as for instance a hospital or a shopping mall, and in these cases the hospital or shopping mall has a fixed network connected to the route. The instance are solved with one and two node visits and with the option of flexible node visits. With the flexible node visit option, all transfer locations have two node visits except for transfer location 1 which has five node visits. Furthermore, the instance is run with using halve the size of the time windows ($H1.4.6stw$), and double the time windows ($H1.4.6btw$).

The detailed cost and penalties in the objective value for the instance $H1.4.6$ are given in Table 31 and Table 32. First, the total objective value is given, then the value corresponding to the fleet costs are given, before the individual elements are given. Similarly the total penalty is given before the individual penalty values. These values are given for the regular instance $H1.4.6$ and for the instances $H1.4.6stw$ and $H1.4.6btw$.

Table 31: Details of Costs and Penalties H1.4.6 1

Cost Parameter	H1.4.6 (1) 000	H1.4.6stw (1) 000	H1.4.6btw (1) 000
Total Cost	5746.00	5767.00	5564.00
Fleet Cost	5746.00	5767.00	5564.00
Vehicles	20.00	25 .00	20.00
Distance	3172.00	3180.00	3070.00
Usage	2554.00	2562.00	2472.00
Penalties	0	0	0
Service Time	0	0	0
Walking	0	0	0
Transfer	0	0	0

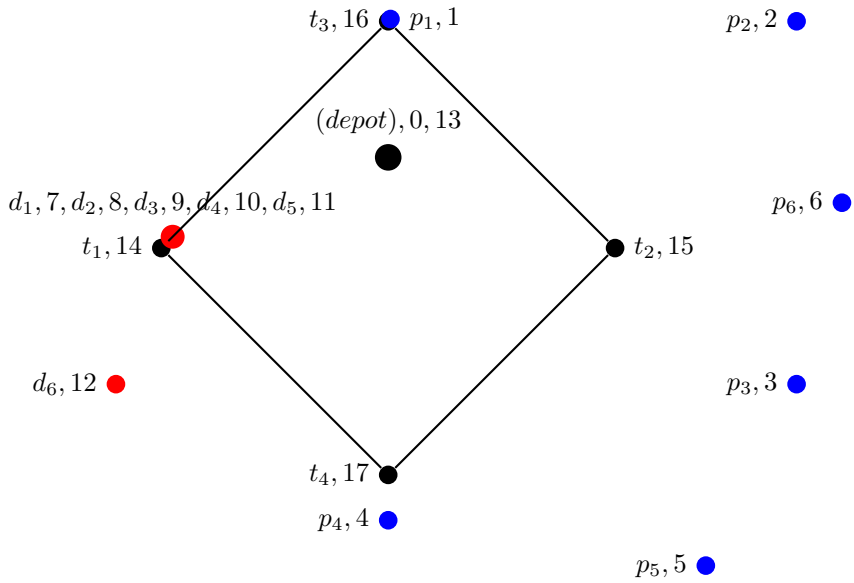

Figure 20: Case H1.4.6

Table 32: Details of Costs and Penalties H1.4.6 2

Cost Parameter	H1.4.6 (1) 111	H1.4.6stw (1) 111	H1.4.6btw (1) 111
Total Cost	3817.80	4332.20	3662.80
Fleet Cost	3475.00	4111.40	3320.00
Vehicles	15.00	15.00	10.00
Distance	1916.00	2268.00	1810.00
Usage	1544.00	1828.40	1500.00
Penalties	342.80	220.80	342.80
Service Time	304.80	212.80	304.80
Walking	30.00	0	30.00
Transfer	8.00	8.00	8.00

The results for the instances are provided in the Table 33 and Table 34. That is, the routes for the vehicles and the trips for the vehicles are given by stating the sequence the nodes are visited in. The sign $-$ illustrate that the arc is driven by a vehicle, $=$ illustrates that the arc is used by a fixed route, while $--$ illustrate that the request walks the arc. For example, request 6 in Table 34 first travel by vehicle from its pickup location (location 6), to a transfer location 15, where it uses the fixed route to location 17 and is transported by vehicle to its drop off location (location 12).

Table 33: Details of Results H1.4.6 1

Vehicle/Request	H1.4.6 (1) 000	H1.4.6stw (1) 000	H1.4.6btw (1) 000
Vehicle 1	0-4-10-6-12-13	0-1-7-5-11-13	0-2-8-5-11-13
Vehicle 2	0-1-7-5-11-13	0-6-12-13	0-6-12-13
Vehicle 3	0-3-9-13	0-3-9-13	0-1-7-4-10-13
Vehicle 4	0-2-8-13	0-2-8-13	0-3-9-13
Vehicle 5		0-4-10-13	
Request 1	1-7	1-7	1-7
Request 2	2-8	2-8	2-8
Request 3	3-9	3-9	3-9
Request 4	4-10	4-10	4-10
Request 5	5-11	5-11	5-11
Request 6	6-12	6-12	6-12

Table 34: Details of Results H1.4.6 2

Vehicle/Request	H1.4.6 (1) 111	H1.4.6stw (1) 111	H1.4.6btw (1) 111
Vehicle 1	0-6-15-3-5-11-9-13	0-6-15-2-8-13	0-6-15-3-5-9-13
Vehicle 2	0-2-8-13	0-4-10-17-12-13	0-2-8-14-17-12-10-13
Vehicle 3	0-17-12-10-13	0-3-5-9-13	
Vehicle 4			
Vehicle 5			
Request 1	1=16=14=7	1=16=14=7	1=16=14=7
Request 2	2-8	2-8	2-8
Request 3	3-5-11-9	3-5-9	3-5-9
Request 4	4- -17-12-10	4-10	4- -17-12-10
Request 5	5-11	5-9- -11	5-9- -11
Request 6	6-15=17-12	6-15=17-12	6-15=17-12

A.1.2 City Case

The city case $C1.7.3$ illustrates areas with a well established fixed network which creates great possibilities for the requests to use the fixed network at least some part of the trip. Instance $C1.7.3$ is solved with one, two and three node visits respectively. In addition, the instances are solved with an $ErrorRate$ of 0.8 ($C1.7.3$) and an $ErrorRate$ of 1.8 ($C1.7.3r$). Furthermore, the instance is run using half the size of the time windows ($C1.7.3stw$).

The detailed cost and penalties in the objective value for the instance $C1.7.3$ are given in Table 35. First, the total objective value is given, then the value corresponding to the fleet costs are given, before the individual elements are given. Similarly the total penalty is given before the individual penalty values. These values are given for the regular instance $C1.7.3$ and for the instance $C1.7.3stw$.

The results for the instance ($C1.7.3$ and $C1.7.3stw$) are provided in the Table 36. That is, the routes for the vehicles and the trips for the vehicles are given by

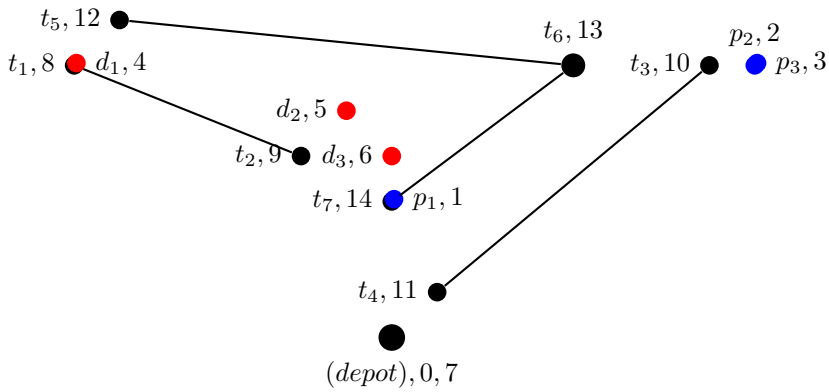

Figure 21: Case C1.7.3

Table 35: Details of Costs and Penalties C1.7.3

Cost Parameter	C1.7.3	C1.7.3r	C1.7.3stw
Total Cost	1256.6	1246.8	1317.6
Fleet Cost	1201	772	1262
Vehicles	5	10	10
Distance	608	422	694
Usage	588	340	558
Penalties	55.6	474.8	55.6
Service Time	55.6	398.8	55.6
Walking	0	60	0
Transfer	0	16	0

stating the sequence the nodes are visited in. The sign $-$ illustrate that the arc is driven by a vehicle, $=$ illustrates that the arc is used by a fixed route, while $--$ illustrate that the request walks the arc. For example, request 1 first use the fixed route to location 14, is picked up by a vehicle and transported to location 9. Here, the request use the fixed route to location 8 and then to its drop off location (location 4).

Table 36: Details of Results C1.7.3

Vehicle/Request	C1.7.3	C1.7.3r	C1.7.3stw
Vehicle 1	0-14-9-2-3-6-5-7	0-11-6-5-7	0-2-3-6-5-7
Vehicle 2		0-14-9-7	0-14-9-7
Request 1	1=14-9=8=4	1=14-9=8=4	1=14-9=8=4
Request 2	2-3-6-5	2- -10=11-6-5	2-3-6-5
Request 3	3-6	3- -10=11-6	3-6

A.1.3 Rural Case

The rural case $R1.4.6$ illustrates situations where the fixed route is less developed. $R1.4.6$ represents a couple of requests which are located far away from the rest of the requests and the fixed route. These requests have a capacity less than the total vehicle capacity. This instance is run both with a vehicle capacity of four, and with vehicle capacity of eight passengers.

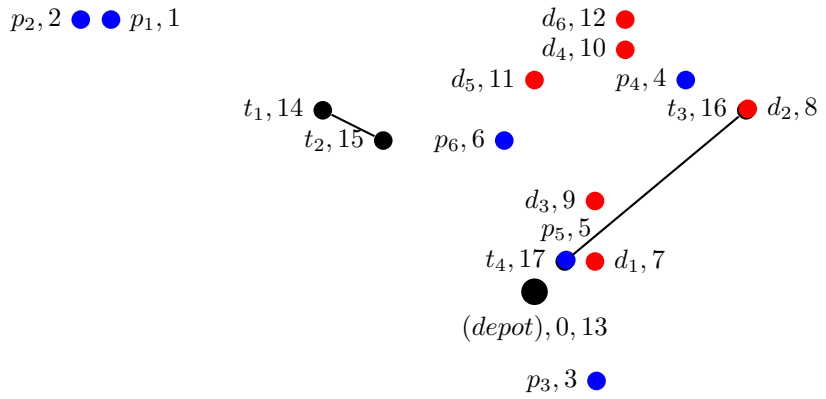

Figure 22: Case R1.4.6

The detailed cost and penalties in the objective value for the instance $R1.4.6$ are given in Table 37. First, the total objective value is given, then the value corresponding to the fleet costs are given, before the individual elements are given. Similarly the total penalty is given before the individual penalty values. These values are given for the regular instance $R1.4.6$ and for the instance $R1.4.6m$. $R1.4.6m$ is equal to $R1.4.6$ except that $R1.4.6m$ has a vehicle fleet where the vehicles have load capacity of eight instead of the regular capacity of four passengers.

Table 37: Details of Costs and Penalties R1.4.6

Cost Parameter	R1.4.6 111	R1.4.6m 111
Total Cost	2983.6	2915.6
Fleet Cost	2728	2628
Vehicles	10	10
Distance	1504	1448
Usage	1214	1170
Penalties	255.6	287.6
Service Time	221.6	253.6
Walking	30	30
Transfer	4	4

The results for the instance ($R1.4.6$ and $R1.4.6m$) are provided in the Table

38. That is, the routes for the vehicles and the trips for the vehicles are given by stating the sequence the nodes are visited in. The sign $-$ illustrate that the arc is driven by a vehicle, $=$ illustrates that the arc is used by a fixed route, while $--$ illustrate that the request walks the arc. For example, request 1 first walk from its pickup location (location 1) to request 2's pickup location (location 2) where it is picked up by a vehicle, transported to location 17 where request 2 is set off, and then to its drop off location (location 7).

Table 38: Details of Results R1.4.6

Vehicle/Request	R1.4.64 111	R1.4.6m 111
Vehicle 1	0-2-17-7-13	0-2-17-7-13
Vehicle 2	0-3-5-9-11-6-12-4-10-13	0-3-5-9-6-11-12-4-10-13
Request 1	1- -2-17-7	1- -2-17-7
Request 2	2-17=16=8	2-17=16=8
Request 3	3-5-9	3-5-9
Request 4	4-10	4-10
Request 5	5-9-11	5-9-6-11
Request 6	6-12	6-11-12

A.2 Small Fixed Networks

In this section the small fixed network with four to five transfer locations are illustrated for the instance with seven requests. Some details of the results for the instances with seven and five requests are also given.

A.2.1 Case 1 Small

The network in case 1 is a square where it is possible to travel from every transfer location to another transfer location. The network can for example illustrate a route through a rural area where one of the transfer locations for example represents a hospital or a shopping centre so the fixed network transport people to or from this location. In addition, the pickup locations and the drop off locations for the instance with seven request $S1.4.7$ is illustrated in the figure.

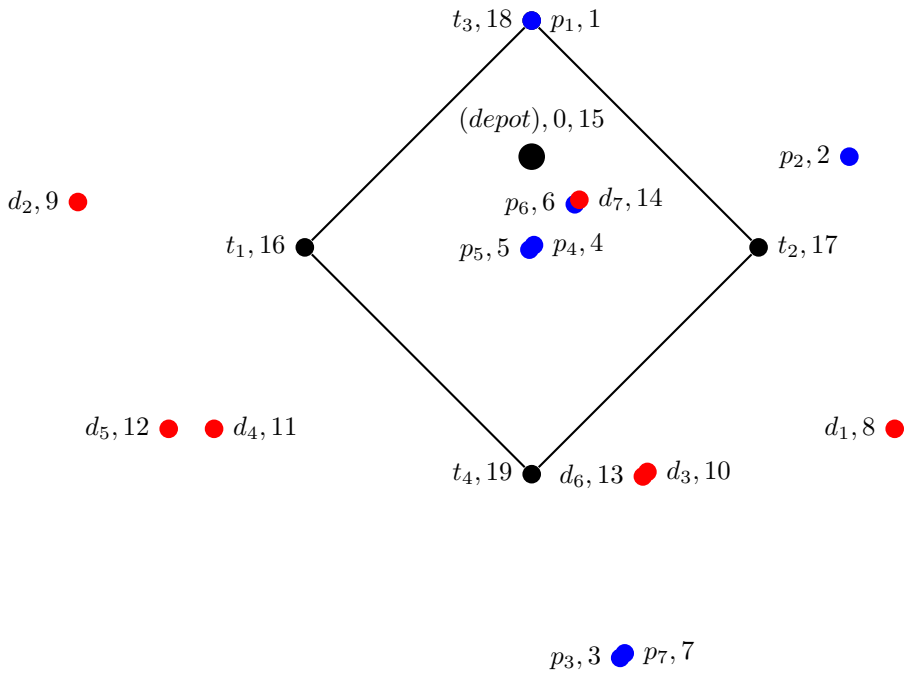

Figure 23: Case S1.4.7

Table 39 gives detailed results for instance $S1.4.7(1)$ with one node visits. 000 says that no problem features are included, while 111 state that all problem features are used. The detailed cost and penalties in the objective value for the instance $S1.4.7$ is provided in the tables. First, the total objective value is given, then the value corresponding to the fleet costs, before the individual elements are given. Similarly, the total penalty is stated before the individual penalty values.

Table 39: Details of Costs and Penalties S1.4.7

Cost Parameter	S1.4.7 (1) 000	S1.4.7 (1) 111
Total Cost	4700	3105
Fleet Cost	4700	2833.80
Vehicles	20	10
Distance	2596	1564
Usage	2084	1260
Penalties	0	271.20
Service Time	0	259.20
Walking	0	0
Transfer	0	12

The results for the problem without activating the options of transfers, multiple requests and walking, and with the options are given below. The sign $-$ illustrate that the arc is driven by a vehicle, $=$ illustrates that the arc is used by a fixed route, while $--$ illustrate that the request walks the arc.

Table 40: Details of Results S1.4.7

Vehicle/Request	S1.4.7 (1) 000	S1.4.7 (1) 111
Vehicle 1	0-4-11-5-12-15	0-6-13-4-11-12-16-9-15
Vehicle 2	0-2-9-15	0-2-17-8-7-10-14-15
Vehicle 3	0-6-13-3-10-7-14-15	
Vehicle 4	0-1-8-15	
Request 1	1-8	1=18=17-8
Request 2	2-9	2-17=16-9
Request 3	3-10	3- -7-10
Request 4	4-11	4-11
Request 5	5-12	5- -4-11-12
Request 6	6-13	6-13
Request 7	7-14	7-10-14

Table 41 and Table 42 the details for the results from the instance with five request and the network from $S1.4$ is given. The requests are located as the five first requests in the Illustration 23, however the ordering of the locations are different for the instance with five locations. Here, initial depot is 0, the pickup locations of the requests go from 1 to 5, the drop off locations go from 6 to 10, final depot is numbered 11, while the transfer locations range from 12 to 16. Table 41 gives detailed results for instance $S1.4.5(2)$ with two node visits and both $ErrorRate$ of 0.8 and 1.8 ($1.4.5r$). The three binary numbers in the first row represents which of the problem features that are activated. The three last number represents which of the problem features (transfer, multiple requests, walking) which are activated. That is, if the first number is 1 the possibility of transfer is included, the second 1 represents that the possibility of multiple requests is activated, while the last num-

ber is 1 when the possibility of walking is used. The detailed cost and penalties in the objective value for the instance $S1.4.5(2)$ is provided in the tables. First, the total objective value is given, then the value corresponding to the fleet costs, before the individual elements are given. Similarly, the total penalty is stated before the individual penalty values. Furthermore, Table 42 shows the routes for the vehicles and the trips for the requests.

Table 41: Details of Costs and Penalties S1.4.5

Cost Parameter	S1.4.5 (2) 000/001	S1.4.5 (2) 010/011	S1.4.5 (2) 100/101	S1.4.5(r) (2) 111/110
Total Cost	3899.00	3269.00	3289.20	2659.20
Fleet Cost	3899.00	3267.00	3024.00	2392.00
Vehicles	15.00	15.00	10.0	10.00
Distance	2154.00	1804.00	1670.00	1320.00
Usage	1730.00	1448.00	1344.00	1062.00
Penalties	0.00	2.00	265.20	267.20
Service Time	0.00	2.00	253.20	255.20
Walking	0.00	0.00	0.00	0.00
Transfer	0.00	0.00	12.00	12.00

Table 42: Details of Results S1.4.5

Vehicle/ Request	S1.4.5 (2) 000/001	S1.4.5 (2) 010/011	S1.4.5 (2) 100/101	S1.4.5(r) (2) 111/110
Vehicle 1	0-4-9-5-10-11	0-1-6-3-8-11	0-4-9-5-10-12-7-11	0-2-13-6-3-8-11
Vehicle 2	0-1-6-3-8-11	0-2-7-11	0-2-13-6-3-8-11	0-4-5-9-10-12-7-11
Vehicle 3	0-2-7-11	0-5-4-9-10-11		
Request 1	1-6	1-6	1=14=13-6	1=14=13-6
Request 2	2-7	2-7	2-13=12-7	2-13=12-7
Request 3	3-8	3-8	3-8	3-8
Request 4	4-9	4-9	4-9	4-5-9
Request 5	5-10	5-4-9-10	5-10	5-9-10

A.2.2 Case 2 Small

The small network for case 2 consist of two lines, one bigger line and a smaller line. The dataset can for example illustrate a small city or countryside with poor fixed route network and where private vehicles and more flexible traffic systems are the primary modes of transport. In addition, the pickup locations and the drop off locations for the instance with seven request $S2.4.7$ is illustrated in the figure.

Table 43 gives detailed results for instance $S2.4.7(1)$ with one node visits. 000 says that no problem features are included, while 111 state that all problem features are used. The detailed cost and penalties in the objective value for the instance

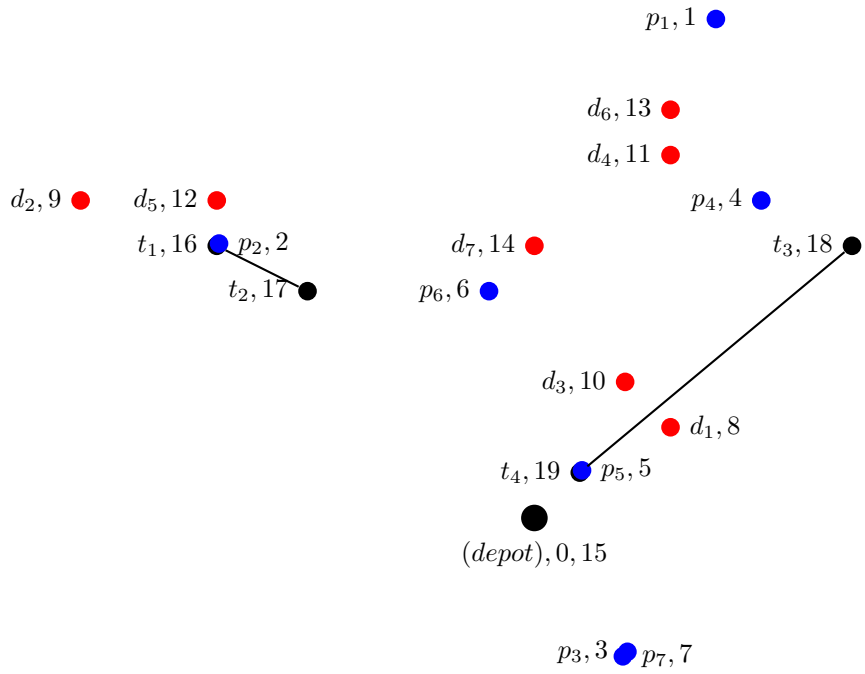

Figure 24: Case S2.4.7

$S2.4.7$ is provided in the tables. First, the total objective value is given, then the value corresponding to the fleet costs, before the individual elements are given. Similarly, the total penalty is stated before the individual penalty values.

Table 43: Details of Costs and Penalties S2.4.7

Cost Parameter	S2.4.7 (1) 000	S2.4.7 (1) 111
Total Cost	3454	3052
Fleet Cost	3454	2932
Vehicles	10	10
Distance	1908	1618
Usage	1536	1304
Penalties	0	120
Service Time	0	120
Walking	0	0
Transfer	0	0

The results for the problem without activating the options of transfers, multiple requests and walking, and with the options are given below. The sign $-$ illustrate that the arc is driven by a vehicle, $=$ illustrates that the arc is used by a fixed route, while $--$ illustrate that the request walks the arc.

Table 44: Details of Results S2.4.7

Vehicle/Request	S2.4.7 (1) 000	S2.4.7 (1) 111
Vehicle 1	0-3-10-5-12-2-9-6-13-15	0-7-3-10-19-14-12-16-9-6-13-15
Vehicle 2	0-4-11-1-8-7-14-15	0-4-11-1-8-15
Vehicle 3		
Vehicle 4		
Request 1	1-8	1-8
Request 2	2-9	2- -16-9
Request 3	3-10	3-10
Request 4	4-11	4-11
Request 5	5-12	5=19-14-12
Request 6	6-13	6-13
Request 7	7-14	7-3-10-19-14

A.2.3 Case 3 Small

The small network in case 3, which is also the example network, consists of two lines connected at the middle. This can for example represent two lines going through a city center and out to less centralized part of town, or into other smaller towns. For a passenger to go from transfer location 1 to transfer location 4 the passenger have to transfer to the other line at transfer location 2 and a penalty is added to the objective value. In addition, the pickup locations and the drop off locations for the instance with seven request $S3.5.7$ is illustrated in the figure.

Table 45 gives detailed results for instance $S3.5.7(1)$ with one node visits. 000 says that no problem features are included, while 111 state that all problem features are used. The detailed cost and penalties in the objective value for the instance $S3.5.7$ is provided in the tables. First, the total objective value is given, then the value corresponding to the fleet costs, before the individual elements are given. Similarly, the total penalty is stated before the individual penalty values.

The results for the problem without activating the options of transfers, multiple requests and walking, and with the options are given below. The sign $-$ illustrate that the arc is driven by a vehicle, $=$ illustrates that the arc is used by a fixed route, while $--$ illustrate that the request walks the arc.

Table 47 to Table 50 the details for the results from the instance with five request and the network from $S3.5$ is given. The requests are located as the five first requests in the Illustration 25, however the ordering of the locations are different for the instance with five locations. Here, initial depot is 0, the pickup locations of the requests go from 1 to 5, the drop off locations go from 6 to 10, final depot is numbered 11, while the transfer locations range from 12 to 16. Table 47 to Table 48 give detailed results for instance $S3.5.5(2)$ with two node visits and both $ErrorRate$ of 0.8 and 1.8 ($3.5.5r$). The three binary numbers in the first row

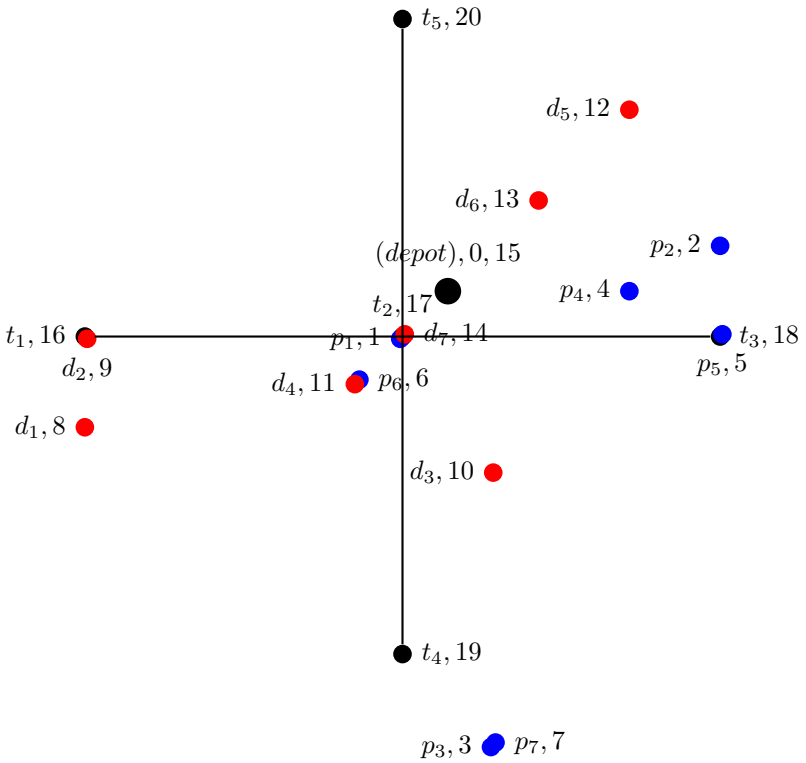

Figure 25: Case S3.5.7

Table 45: Details of Costs and Penalties S3.5.7

Cost Parameter	S3.5.7 (1) 000	S3.5.7 (1) 111
Total Cost	3747	2538
Fleet Cost	3747	2352
Vehicles	15	10
Distance	2068	1296
Usage	1664	1046
Penalties	0	186
Service Time	0	182
Walking	0	0
Transfer	0	4

represents which of the problem features that are activated. The three last number represents which of the problem features (transfer, multiple requests, walking) which are activated. That is, if the first number is 1 the possibility of transfer is included, the second 1 represents that the possibility of multiple requests is activated, while the last number is 1 when the possibility of walking is used. The detailed cost and penalties in the objective value for the instance $S3.5.5(2)$ is provided in

Table 46: Details of Results S3.5.7

Vehicle/Request	S3.5.7 (1) 000	S3.5.7 (1) 111
Vehicle 1	0-1-8-3-10-7-14-15	0-4-17-11-8-7-10-14-15
Vehicle 2	0-4-11-6-13-5-12-15	0-6-13-2-18-12-15
Vehicle 3	0-2-9-15	
Vehicle 4		
Request 1	1-8	1=17-11-8
Request 2	2-9	2-18=16=9
Request 3	3-10	3- -7-10
Request 4	4-11	4-17-11
Request 5	5-12	5- -18-12
Request 6	6-13	6-13
Request 7	7-14	7-10-14

the tables. First, the total objective value is given, then the value corresponding to the fleet costs, before the individual elements are given. Similarly, the total penalty is stated before the individual penalty values. Table 49 to Table 50 illustrates the routes for the vehicles and the trips for the requests.

Table 47: Details of Costs and Penalties S3.5.5 1

Cost Parameter	S3.5.5 (2) 000	S3.5.5 (1)(2) 111	S3.5.5r (1) 111	S3.5.5r (2) 111
Total Cost	3209	2435	2296	2144
Fleet Cost	3209	2291	2042	1758
Vehicles	15	15	10	10
Distance	1770	1260	1126	968
Usage	1424	1016	906	780
Penalties	0	144	254	386
Service Time	0	140	250	318
Walking	0	0	0	60
Transfer	0	4	4	8

Table 48: Details of Costs and Penalties S3.5.5 2

Cost Parameter	S3.5.5 (2) 001/011	S3.5.5 (2) 010	S3.5.5 (2) 100/110/101
Total Cost	3123	3153	2435
Fleet Cost	3085	3147	2291
Vehicles	15	15	15
Distance	1700	1736	1260
Usage	1370	1396	1016
Penalties	38	6	144
Service Time	38	6	140
Walking	0	0	0
Transfer	0	0	4

Table 49: Details of Results S3.5.5 1

Vehicle/Request	S3.5.5 (2) 000	S3.5.5 (1)(2) 111	S3.5.5r (1) 111	S3.5.5r (2) 111
Vehicle 1	0-1-6-3-8-11	0-1-13-6-3-8-11	0-13-6-3-8-11	0-4-9-3-8-11
Vehicle 2	0-4-9-5-10-11	0-2-14-5-10-11	0-2-14-4-10-9-11	0-2-14-5-10-11
Vehicle 3	0-2-7-11	0-4-9-11		
Request 1	1-6	1-13-6	1=13-6	1=13=12- -6
Request 2	2-7	2-14=12=7	2-14=12=7	2-14=12=7
Request 3	3-8	3-8	3-8	3-8
Request 4	4-9	4-9	4-10-9	4-9
Request 5	5-10	5-10	5=14-4-10	5-10

Table 50: Details of Results S3.5.5 2

Vehicle/Request	S3.5.5 (2) 001/011	S3.5.5 (2) 010	S3.5.5 (2) 100/110/101
Vehicle 1	0-13-6-3-8-14-10-11	0-3-8-5-10-11	0-1-6-3-8-11
Vehicle 2	0-2-7-11	0-2-7-11	0-2-14-10-11
Vehicle 3	0-4-9-11	0-4-1-9-6-11	0-4-9-11
Request 1	1- -13-6	1-9-6	1-6
Request 2	2-7	2-7	2-14=12=7
Request 3	3-8	3-8	3-8
Request 4	4-9	4-1-9	4-9
Request 5	5- -14-10	5-10	5=14-10

A.3 Big Fixed Networks

Some bigger fixed network have also been considered consisting between seven to nine transfer locations, and their networks are illustrated in Figure 26 to Figure 28. Furthermore, the pickup and drop off locations for the instances with five requests are included in the illustrations. The structure of the fixed networks are similar as for the smaller networks.

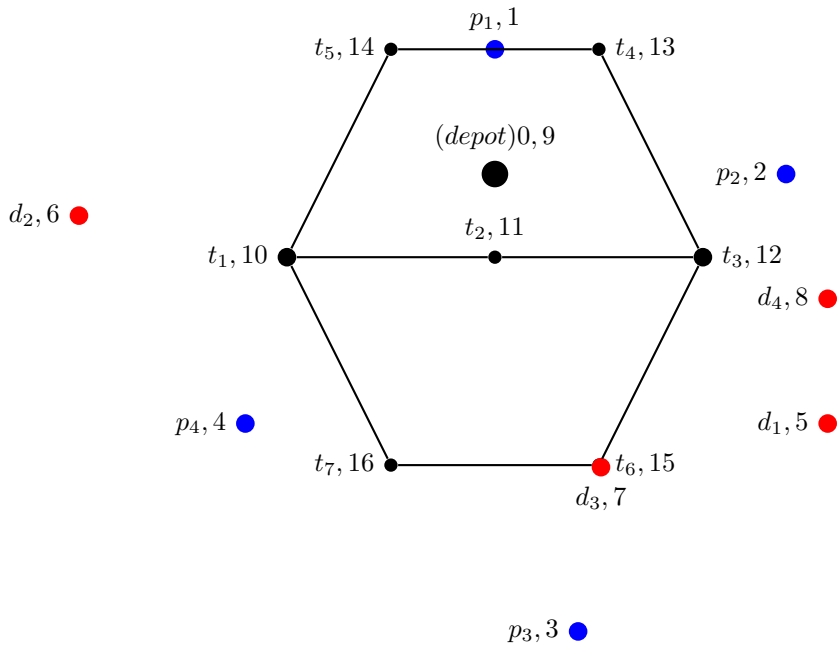

Figure 26: Case B1.7.4

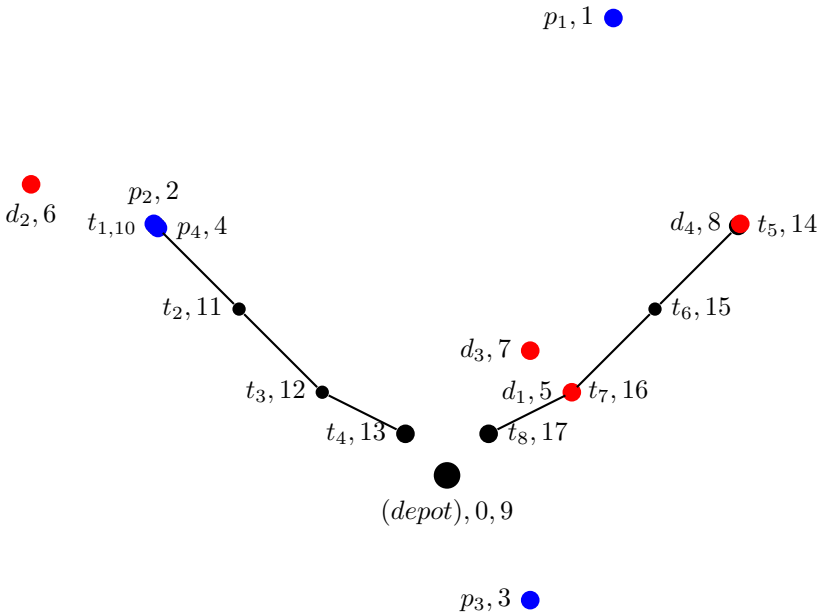

Figure 27: Case B2.8.4

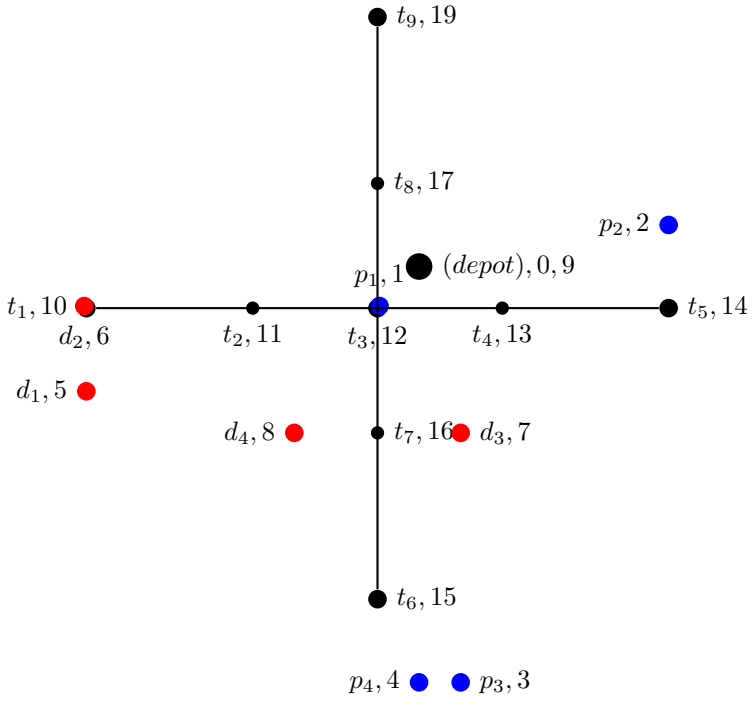

Figure 28: Case B3.9.4

B Mathematical Model

B.1 Sets

$\mathcal{P}$	Set of pickup locations
$\mathcal{D}$	Set of drop off locations
$\mathcal{G}$	Set of transfer locations
$\mathcal{N}$	Set of all nodes
$\mathcal{N}^{\mathcal{P}}$	Set of pickup nodes
$\mathcal{N}^{\mathcal{D}}$	Set of drop off nodes
$\mathcal{N}_g^G$	Set of transfer nodes corresponding to transfer location g
$\mathcal{R}$	Set of requests
$\mathcal{K}$	Set of vehicles

B.2 Parameters

P^S	Penalty cost for service time above direct travel time for each request
P^R	Penalty cost for each request walking a certain distance
P^T	Penalty per transfer for each request
T_{ij}^K	Travel time when using a vehicle from location i to location j
T_{ij}^F	Travel time when using a fixed network from location i to location j
T_{ij}^W	Travel time when walking from location i to location j
D_{ij}	Distance of traveling from location i to location j
C_{ij}	Cost of traveling from location i to location j using a vehicle
C^U	Cost per time unit of using each vehicle
Q	Capacity of a vehicle
U	Maximum usage time of a vehicle
D_r^R	Allowed walking distance by request r
L_r	Load of request r
$\underline{T}_i$	Earliest time at which service may begin at location i
$\overline{T}_i$	Latest time at which service may begin at location i
Z_{ij}	$\begin{cases} 1 & \text{if there exists a fixed route from transfer location } i \text{ to transfer location } j \\ 0 & \text{otherwise} \end{cases}$
F_{ir}	$\begin{cases} 1 & \text{if node } i \text{ is the pickup node of request } r \\ -1 & \text{if node } i \text{ is the drop off node of request } r \\ 0 & \text{otherwise} \end{cases}$
E	Fraction of allowed service time

B.3 Variables

x_{imjnk} $\begin{cases} 1 & \text{if vehicle } k \text{ travels from node } (i,m) \text{ to node } (j,n) \\ 0 & \text{otherwise} \end{cases}$

y_{imjnr} $\begin{cases} 1 & \text{if request } r \text{ travels from node } (i,m) \text{ to node } (j,n) \text{ on a vehicle} \\ 0 & \text{otherwise} \end{cases}$

w_{imjnr} $\begin{cases} 1 & \text{if request } r \text{ travels from node } (i,m) \text{ to node } (j,n) \text{ using a fixed network} \\ 0 & \text{otherwise} \end{cases}$

v_{imjnr} $\begin{cases} 1 & \text{if request } r \text{ walks from node } (i,m) \text{ to node } (j,n) \\ 0 & \text{otherwise} \end{cases}$

t^A_{im} arrival time at node (i,m)

t^D_{im} departure time at node (i,m)

t^A_k arrival time at final depot for vehicle k

t^D_k departure time at initial depot for vehicle k

t_r number of transfers of request r

B.4 IDARP formulation

$$Minimize\Big(\sum_{(i,m)\in\mathcal{N}} \sum_{(j,n)\in\mathcal{N}} \sum_{k\in\mathcal{K}} C_{ij} x_{imjnk} + C^U \sum_{k\in\mathcal{K}} (t^A_k - t^D_k) + P^T \sum_{r\in\mathcal{R}} t_r$$
$$+P^S \sum_{r\in\mathcal{R}} ((t^A_{r+\overline{r}} - t^D_r) - T_{r,r+\overline{r}}) + P^R \sum_{(i,m)\in\mathcal{N}} \sum_{(j,n)\in\mathcal{N}} \sum_{r\in\mathcal{R}} D_{ij} v_{imjnr} \Big) \quad (90)$$

Subject to

$$\sum_{(j,n)\in\mathcal{N}} x_{01jnk} \le 1 \qquad k \in \mathcal{K} \quad (91)$$

$$\sum_{(j,n)\in\mathcal{N}} x_{jn,2\overline{r}+1,1k} - \sum_{(j,n)\in\mathcal{N}} x_{01jnk} = 0 \qquad k \in \mathcal{K} \quad (92)$$

$$\sum_{(j,n)\in\mathcal{N}} x_{jnimk} - \sum_{(j,n)\in\mathcal{N}} x_{imjnk} = 0 \qquad (i,m) \in \mathcal{N}, k \in \mathcal{K} \quad (93)$$

$$\sum_{(j,n)\in\mathcal{N}} \sum_{k\in\mathcal{K}} x_{imjnk} \le 1 \qquad (i,m) \in \mathcal{N} \quad (94)$$

$$\sum_{(j,n)\in\mathcal{N}} y_{r1jnr} - \sum_{(j,n)\in\mathcal{N}} \sum_{k\in\mathcal{K}} x_{r1jnk} = 0 \qquad r \in \mathcal{R} \quad (95)$$

$$\sum_{(j,n)\in\mathcal{N}} y_{r1jnr} - \sum_{(j,n)\in\mathcal{N}} \sum_{k\in\mathcal{K}} x_{r1jnk} = 0 \qquad r \in \mathcal{R} \quad (96)$$

$$\sum_{(j,n)\in\mathcal{N}}(y_{i1jnr}-y_{jni1r})$$
$$+\sum_{(j,n)\in\mathcal{N}}(v_{i1jnr}-v_{jni1r})$$
$$+\sum_{(g(i),n)\in\mathcal{N}^G_{g(i)}} w_{i1g(i)nr}=F_{ir} \quad (i,1)\in\mathcal{N}^P, r\in\mathcal{R} \tag{97}$$

$$\sum_{(j,n)\in\mathcal{N}}(y_{jni1r}-y_{i1jnr})$$
$$+\sum_{(j,n)\in\mathcal{N}}(v_{jni1r}-v_{i1jnr})$$
$$+\sum_{(g(i),n)\in\mathcal{N}^G_{g(i)}} w_{g(i)ni1r}=F_{ir} \quad (i,1)\in\mathcal{N}^D, r\in\mathcal{R} \tag{98}$$

$$\sum_{(j,n)\in\mathcal{N}} y_{imjnr}+\sum_{j\in\mathcal{G}}\sum_{(j,n)\in\mathcal{N}^G_j} w_{imjnr}$$
$$+\sum_{(j,n)\in\mathcal{N}} v_{imjnr}-\sum_{(j,n)\in\mathcal{N}} v_{imjnr}$$
$$-\sum_{(j,n)\in\mathcal{N}} y_{jnimr}-\sum_{j\in\mathcal{G}}\sum_{(j,n)\in\mathcal{N}^G_j} w_{jnimr}=0 \qquad i\in G,(i,m)\in\mathcal{N}^G_i, r\in\mathcal{R} \tag{99}$$

$$\sum_{k\in\mathcal{K}} x_{imjnk}(t^D_{im}+T^K_{ij}-t^A_{jn})\le 0 \quad (i,m)\in\mathcal{N},(j,n)\in\mathcal{N} \tag{100}$$

$$\sum_{r\in\mathcal{R}} w_{imjnr}(t^A_{im}+T^F_{ij}-t^D_{jn})\le 0 \quad i,j\in\mathcal{G},(i,m)\in\mathcal{N}^G_i,(j,n)\in\mathcal{N}^G_j \tag{101}$$

$$\sum_{r\in\mathcal{R}} w_{imjnr}(t^D_{im}-t^D_{jn})\le 0 \quad (i,m)\in\mathcal{N}^P, j\in\mathcal{G},(j,n)\in\mathcal{N}^G_j \tag{102}$$

$$\sum_{r\in\mathcal{R}} w_{jnimr}(t^D_{jn}-t^A_{im})\le 0 \quad (i,m)\in\mathcal{N}^D, j\in\mathcal{G},(j,n)\in\mathcal{N}^G_j \tag{103}$$

$$\sum_{r\in\mathcal{R}} v_{imjnr}(t^D_{im}+T^R_{ij}-t^A_{jn})\le 0 \quad (i,m)\in\mathcal{N},(j,n)\in\mathcal{N} \tag{104}$$

$$t^D_{i1}+T^K_{i,\overline{r}+i}-t^A_{\overline{r}+i,1}\le 0 \quad (i,1)\in\mathcal{N}^P \tag{105}$$

$$t^A_{im}-t^D_{im}\le 0 \quad (i,m)\in\mathcal{N} \tag{106}$$

$$x_{01imk}(t^D_k+T^K_{0i}-t^A_{im})\le 0 \quad (i,m)\in\mathcal{N},k\in\mathcal{K} \tag{107}$$

$$x_{im,2\overline{r}+1,1k}(t^D_{im}+T^K_{i,2\overline{r}+1}-t^A_k)\le 0 \quad (i,m)\in\mathcal{N},k\in\mathcal{K} \tag{108}$$

$$t_k^A - t_k^D \leq U \qquad k \in \mathcal{K} \tag{109}$$

$$t_{\overline{r}+r,1}^A - t_{r1}^D \leq T_{r,\overline{r}+r}^K (1+E) \qquad r \in \mathcal{R} \tag{110}$$

$$\underline{T}_i \leq t_{im}^D \leq \overline{T}_i \qquad (i,m) \in \mathcal{N} \tag{111}$$

$$\begin{aligned} &2 \cdot \sum_{i \in \mathcal{G}} \sum_{(i,m) \in \mathcal{N}_i^G} \sum_{j \in \mathcal{G}} \sum_{(j,n) \in \mathcal{N}_j^G} w_{imjnr} \\ &- \sum_{i \in \mathcal{G}} \sum_{(i,m) \in \mathcal{N}_i^G} \sum_{(g(i),n) \in \mathcal{N}_{g(i)}^G} w_{img(i)nr} \\ &- \sum_{i \in \mathcal{G}} \sum_{(i,m) \in \mathcal{N}_i^G} \sum_{(g(i),n) \in \mathcal{N}_{g(i)}^G} w_{g(i)nimr} \leq t_r \qquad r \in \mathcal{R} \end{aligned} \tag{112}$$

$$0 \leq \sum_{r \in \mathcal{R}} L_r y_{imjnr} \leq \sum_{k \in \mathcal{K}} Q x_{imjnk} \qquad (i,m) \in \mathcal{N}, (j,n) \in \mathcal{N} \tag{113}$$

$$x_{imjnk} \in [0,1] \qquad (i,m) \in \mathcal{N}, (j,n) \in \mathcal{N}, k \in \mathcal{K} \tag{114}$$

$$y_{imjnr} \in [0,1] \qquad (i,m) \in \mathcal{N}, (j,n) \in \mathcal{N}, r \in \mathcal{R} \tag{115}$$

$$w_{imjnr} \in [0,1] \qquad i,j \in \mathcal{G}, (i,m) \in \mathcal{N}_i^G, (j,n) \in \mathcal{N}_j^G, r \in \mathcal{R}, Z_{ij} = 1 \tag{116}$$

$$v_{imjnr} \in [0,1] \qquad (i,m) \in \mathcal{N}, (j,n) \in \mathcal{N}, r \in \mathcal{R}, D_r^R \geq D_{ij} \tag{117}$$

$$t_r \in [0,1,...] \qquad r \in \mathcal{R} \tag{118}$$

$$t_{im}^A \geq 0 \qquad (i,m) \in \mathcal{N} \tag{119}$$

$$t_{im}^D \geq 0 \qquad (i,m) \in \mathcal{N} \tag{120}$$

$$t_k^A \geq 0 \qquad k \in \mathcal{K} \tag{121}$$

$$t_k^D \geq 0 \qquad k \in \mathcal{K} \tag{122}$$

C Digital Attachements

Files attached are:

- Mosel implementation: $IDARP.mos$
- Data instances
- The thesis in PDF: $IDARPrapport.pdf$
- Instructions for how to use the files: $ReadMe.txt$