SYDDANSK UNIVERSITET
UNIVERSITY OF SOUTHERN DENMARK

Algorithms for the Truck and Trailer Routing Problem

Master's Thesis by

Ralph Zitz

`ralph@imada.sdu.dk`

Advisor: Professor, Ph.D., Dr. Scient, Jørgen Bang-Jensen

Department of Mathematics and Computer Science
University of Southern Denmark, Odense

Abstract

In the present thesis an extended version of the Truck and Trailer Routing Problem (TTRP) is considered, in which the main objective is to minimize the total length of all constructed routes.

The TTRP is an interesting variant of the well known Vehicle Routing Problem (VRP). However, the TTRP differs from the classic VRP since a subset of the vehicles are allowed pull trailers for the benefit of increased capacity. Doing so, however, is not without its merits since not all customers may be serviced by a vehicle pulling a trailer, and therefore the vehicle may have to find a suitable parking place for the trailer before visiting such customers.

Several variants of the TTRP have been described in the literature. This thesis expands on the model initially defined by Chao [Cha02] by introducing several additional constraints, which mimic problems that could arise in a real-world application. Thus, both time window constraints as well as load constraints are considered.

Algorithms for constructing the initial solutions as well as methods for improving these are presented, and several experiments were conducted on newly generated problem instances in order to determine the best strategies employed by the implemented algorithms.

Finally, it is argued that the use of trailers is beneficial in problem settings which include time windows for customers, while respecting the load constraints of the employed vehicles.

Preface

During a whole year with an interesting and challenging project, it is surprisingly difficult to summarize all the work into a single document. Looking back, the entire process was allowed to go in its own direction, which consequently also allowed many blind alleys and several bumps in the road to be discovered.

It has been a year with many learning experiences some of which are now contained within these pages, while others may have led to a more rapidly receding hairline.

But all in all it has been challenging and fun.

> *A learning experience is one of those things that say, "You know that thing you just did? Don't do that."*
>
> - Douglas Adams.

Acknowledgements

Without the help, support and guidance I have received, writing this thesis would not have been possible and it is therefore a pleasure to thank all the people who made a contribution. However, a small number of people deserve to be mentioned individually. Hence my special thanks go to

Professor, Ph.D., Dr. Scient, Jørgen Bang-Jensen, for accepting me as his student, providing me with an interesting basis for a master's thesis, and for devoting his precious time to discuss problems along the way.

Assistant Professor, Ph.D. Marco Chiarandini, for sparking my interest in combinatorial optimization as well as creative discussions regarding this thesis.

M.Sc., Ph.D. Student Steffen Elberg Godskesen, for countless good advice and discussions.

My Wife, for being **exceptionally** loving, understanding, and supportive during this long year.

My Son, for pulling me back into the real world after a long days work, by allowing me to change his diaper.

My family, for supporting me during the course of my many years of study.

M.Sc. Jacob Aae Mikkelsen and Stud.Sc. Magnus Gausdal Find, for proof reading and providing valuable suggestions.

Balkonen, for simply being Balkonen - and making everything a little more fun.

Contents

Chapter 1

Introduction

The efficient routing of objects is of significant importance in our modern society. Several aspects of routing affect us on a daily basis regardless if we think about it or not. Whether it is browsing the internet, sending mail, transporting consumer goods, or simply travelling, routing is involved.

Recent developments in the world economy, as well as increased interest in environmental concerns, justify the application of efficient techniques to solve complicated routing problems. Benefits include lowering cost expenditures as well as possibly reducing pollution from vehicles, but also introducing time wise gains since routes may potentially be shortened.

Algorithms for solving the Vehicle Routing Problem (VRP) address the issue of finding an efficient routing plan for the logistic problems mentioned above. Informally, the VRP is defined as having a pool of customers that have to be visited once. A fleet of vehicles is available and the problem is to visit all customers while minimizing some objective such as for example total distance travelled. In the real world, the fleet of vehicles, or a subset hereof, often make use of trailers when making deliveries or transporting goods etc., but due to national regulations, road conditions etc., some of the customers cannot be visited with a trailer.

An example of this problem involved the distribution of dairy products by the Dutch dairy industry [Ger96]. Many customers were located in cities with heavy traffic and limited parking places. Therefore, the truck had to park the trailer before it could service some customers along a certain route and then return to pick up the trailer again. Another case involved the delivery of compound animal feed to farmers. Because of narrow roads and/or small bridges on the delivery routes, various types of vehicles called double bottoms, consisting of a truck and a trailer were commonly used.

It is clear that the efficient routing of trucks and trailers is highly relevant, but in spite of this, the problem has received little attention in the literature.

To grab the details of this thesis, it is expected that the reader has a level of knowledge in the field of computer science at the level of masters degree or above.

1.1 Motivation

This thesis is based, and motivated by a formal problem description by the Danish company Transvision A/S [A/S75]. For approximately 30 years the company has specialized in advanced transportation and distribution planning systems. A brief summary of the description follows (for a more detailed description please see section 2.4). The original description in Danish is available in appendix A.

The "Trailer Problem" is described as a classical distribution problem given a number of customers which have to be serviced from a terminal or depot. The vehicles consist of a truck pulling a trailer each with their own capacity. Some of the customers cannot be serviced by a vehicle pulling a trailer, hence it has to be parked somewhere in advance. De-coupling and re-coupling the trailer is associated with a cost, and an additional cost is associated with every section of a route on which a truck pulls a trailer. A number of locations at which a trailer can be parked is given beforehand. The problem thus consists of determining when and where to park a trailer in order to minimize the cost associated with the routing plan. Here are a number of possible complications which may be considered:

1. A parked trailer may be picked up by a different truck.

2. Load constraints impose rules upon the load of the trailer in relation to the truck.

3. Multiple terminals or depots from which the vehicles start. Trailers have to return to the depot of their origin (see 1).

4. Time windows associated with customers. The speed of a vehicle depends on whether it has a trailer coupled or not.

Although simply stated, the above problem description, is in fact quite problematic to solve - even when none of the possible complications are considered. In addition, the fact that the problem is commonly encountered in real-life applications makes solving it a highly motivating factor.

As mentioned previously, problems of this kind have not received much attention in the literature, however, the work of Chao [Cha02] and the Ph.D. thesis by Scheuerer [Sch04] shed some light on a possible solution method which has served as initial inspiration.

1.2 Aim and Contribution of Thesis

The aim and goal of this thesis is to develop an algorithm capable of solving the Truck and Trailer Routing Problem (TTRP), or as shall be described in the following sections, a variation hereof called "The Extended Truck and Trailer Routing Problem" (ETTRP), which takes some of the additional constraints posed by Transvision A/S into account.

The contribution of this thesis can briefly be summarized as:

- The description of a new model capable of handling a specific subset of the possible complications proposed by Transvision A/S.
- The implementation of a new construction heuristic for the ETTRP.
- The implementation of a metaheuristic not previously used for solving this type of routing problem (The Attribute Based Hill Climber) (ABHC).

Besides attempting to solve the ETTRP in general, this thesis is also to be considered a study of the ABHC's ability to solve the problem at hand.

To evaluate the implemented algorithms, the results obtained are compared to those listed in the literature for which known benchmark problems exist.

Finally a summary of the results on newly developed benchmark instances along with ideas and discussions for future extensions and work is given.

Chapter 2

Preliminary Models

The Vehicle Routing Problem (VRP) is a well known and well studied combinatorial optimization problem. The need to handle more realistic constraints than those present in the classic VRP, has given rise to many variations of the original model. The variations include (but are not limited to) capacity constraints on vehicles, pick-up and delivery problems, split delivery, time window constraints on customers, multiple depots, and a wide variety of objective functions. In order to avoid confusion it is necessary to state the exact problem studied in this thesis, and specifically where it deviates from earlier works.

In the following section the most basic VRP model is introduced, and in sections 2.2-2.4 these definitions are extended.

2.1 The Vehicle Routing Problem (VRP)

One of the most classic combinatorial optimization problems is the Travelling Salesman Problem (TSP). The objective of the problem is that a travelling salesman has to visit a set of customers or cities while minimizing the total distance travelled. Doing so, the salesman has to return to the origin or depot to complete the tour.

A generalization of the TSP is the Vehicle Routing Problem (VRP), in which several vehicles starting from the same city set out to service the same set of customers as in the TSP. Again, the vehicles complete their tours by returning to the depot. In the context of the TSP the vehicles correspond to salesmen, thus in the case of a single vehicle the VRP equals the TSP.

2.2 The Capacitated Vehicle Routing Problem (CVRP)

A further generalization of the VRP is the Capacitated Vehicle Routing Problem (CVRP). Here the delivery of goods to customers is involved using a fleet of homogeneous vehicles. Naturally, these vehicles have a limited capacity and thus in a normal setting, only a subset of the customers can be serviced by each vehicle. Postal offices, Fuel delivery trucks etc. are all examples where the CVRP is applicable. Since the

CVRP resembles the routing problem treated in this thesis, a more formal definition from Toth and Vigo [TV02] of the problem follows.

The CVRP can be described as a graph theoretic problem. Let $G = (V, A)$ be a complete directed graph, where $V = \{0, \ldots, n\}$ is the vertex set and A is the arc set. Vertices $i = 1, \ldots, n$ correspond to the customers, whereas vertex 0 corresponds to the depot. Sometimes the depot is associated with vertex $n + 1$.

A non-negative cost, c_{ij} is associated with each arc $(i, j) \in A$ and represents the travel cost spent when going from vertex i to vertex j. We do not allow loop arcs such as (i, i), by imposing the cost, $c_{ii} = +\infty$, $\forall i \in V$. If the graph is directed, the problem is said to be asymmetric, however, in the context of this thesis all graphs are assumed to be symmetric i.e. $c_{ij} = c_{ji}$, $\forall\ (i, j) \in A$, and the arc set A is generally replaced by a set of undirected edges E. Given an edge $e \in E$, we let $\alpha(e)$ and $\beta(e)$ denote its endpoint vertices. In the following we denote the edge set of the undirected graph G by A when edges are indicated by means of their endpoints (i, j), $i, j \in V$, and by E when edges are indicated through a single index e. Following the above definitions we let $|e|$ respectively $d(i, j)$ denote the length of an edge $e \in E$ or $(i, j) \in A$, depending on the context in which it appears. For all graphs it is assumed that the triangle inequality is satisfied:

$$c_{ij} \leq c_{ik} + c_{kj}, \forall i, j, k \in V \tag{2.2.1}$$

The graphs treated in this thesis all have vertices associated with points in the plane, and as such the cost c_{ij}, for each arc $(i, j) \in A$, is defined as the Euclidean distance between two points corresponding to the vertices i and j.

Each customer $i = 1, \ldots, n$ is associated with a non-negative demand d_i to be delivered, and the depot has a fictitious demand $d_0 = 0$. Given a vertex set $S \subseteq V$, we define the total demand of the set by:

$$\mathcal{D}(S) = \sum_{i \in S} d_i \tag{2.2.2}$$

A set of K identical vehicles, each with capacity C, is available at the depot. Feasibility is ensured by assuming that $d_i \leq C, \forall i \in V$. Each vehicle may perform at most one route, and we assume that K is larger than K_{min}, where K_{min} is the minimum number of vehicles needed to serve all customers. The value of K_{min} may be determined by solving a bin packing problem having bins of capacity C, and items of weight d_i, $i = 1, \ldots, n$.

Combinatorially, a solution to the routing problem consists of a partition of V into K routes $R = \{R_1, \ldots, R_K\}$ each satisfying:

$$\mathcal{D}(R_k) \leq C,\ k = 1, \ldots, K \tag{2.2.3}$$

Corresponding to $k = 1, \ldots, K$ ordered sequences or Hamiltonian cycles σ_k defined as:

$$\sigma_k = (0_s, i, \ldots, j, 0_e) \text{ where } i, j \in R_k, i \neq j, \forall i, j \tag{2.2.4}$$

To avoid ambiguity whenever a vertex is listed twice within a sequence σ_k or similar, we let i_s and i_e denote the "start" respectively the "end" vertex in the context where it appears. Thus a vertex i_s *must* be visited before i_e thereby imposing an ordering. Furthermore we let i^+ and i^- denote the customer immediately following customer i (in a routing plan) and the customer immediately preceding customer i respectively.

A solution is the union of K cycles (routes) whose only intersection is the depot node:

$$\bigcap_{k=1}^{K} R_k = \{0\} \tag{2.2.5}$$

Each cycle corresponds to the route serviced by one of the K vehicles.

The customers $\mathcal{C} \subseteq V$ of a route or sequence σ_k is given by:

$$\mathcal{C}(R_k) = \{q \in R_k \mid q \neq 0\} \tag{2.2.6}$$

The edges $\mathcal{E} \subseteq E$ of a route R_k or sequence σ_k is given by:

$$\mathcal{E}(R_k) = \big\{(i,j) \mid j = i^+,\ i \in R_k \setminus \{0_e\}\big\} \tag{2.2.7}$$

Similarly the total length of a route is given by

$$L(R_k) = \sum_{e \in \mathcal{E}(R_k)} |e|,\ k \in [1,\dots,K] \tag{2.2.8}$$

And finally the length of a routing plan is defined by the sum of all route lengths:

$$L(R) = \sum_{k=1}^{K} L(R_k) \tag{2.2.9}$$

As mentioned above the objective is typically a minimization of some objective function ω over the routing plan, subject to the constraints (2.2.3), (2.2.4), and (2.2.5).

$$\min \omega(R) \tag{2.2.10}$$

It is easy to see that the CVRP generalizes the VRP since setting each customers demand: $d_i = 0, \forall i \in V$ reduces the VRP to the CVRP.

For more information on the CVRP see Toth & Vigo [TV02].

2.3 The Vehicle Routing Problem with Time Windows (VRPTW)

The Vehicle Routing Problem with Time Windows is an extension to the classic CVRP. The additional constraint imposed by this model is the introduction of a time window for each customer including the depot. For each customer the time window specifies a time span in which the customer should be serviced by a vehicle. Using the notation from Solomon [Sol87] each customer has a service time s_i associated with it, which denotes the time the vehicle has to spend servicing customer i before continuing its tour. The service time of the depot is $s_0 = 0$. The time windows $[e_i, l_i]$ are defined by an earliest start of service e_i and a latest start of service l_i, $\forall i \in V$. A vehicle is allowed to arrive earlier than e_i at customer i, but as a consequence has to wait until e_i before service can commence. The latest a vehicle is allowed to arrive at a customer i is l_i. The time window associated with the depot imposes additional constraints. First, a vehicle is not allowed to start its tour before e_0. Secondly, a vehicle is not allowed to end its tour later than l_0. Thus the depots time window can be used as total time limit for each tour, allowing the routing plan R to effectively model a normal working day (in Denmark 8 hours).

Since every customer has a time window $[e_i, l_i]$ associated with it we may also introduce the concepts of waiting time w_i, arrival time a_i, beginning of service b_i, as well as earliest departure δ_i and latest arrival ψ_i for every customer $i \in V$, of which the last two are described in greater detail in section 5.2.

If we assume a vehicle travels directly from customer i to customer j where c_{ij} is the direct travel time between i and j, then we may express the arrival time a_j at customer j as follows:

$$a_j = b_i + s_i + c_{ij} \tag{2.3.1}$$

From this it follows that the start of service at customer j can be calculated from:

$$b_j = \max\{e_j, a_j\} \tag{2.3.2}$$

Therefore the waiting time at customer j can be expressed as:

$$w_j = \max\{b_j - a_j, 0\} \tag{2.3.3}$$

Hence the total waiting time of a route k is:

$$\mathcal{W}(R_k) = \sum_{j \in \mathcal{C}(R_k)} w_j \tag{2.3.4}$$

The equations (2.3.1), (2.3.2), and (2.3.3) are standard definitions found in the literature but are included here for completeness.

In general, time windows are either considered to be hard of soft constraints. When viewed as hard constraints the time windows may not be violated. In the case of soft time windows, this constraint is relaxed and violations are tolerated although at a cost in the evaluation function.

In the context of this thesis all time window constraints are considered hard, therefore, a feasible routing plan has to ensure that the start of service b_i for all customers including the depot $i \in V$ is satisfied, i.e.:

$$b_i \in [e_i, l_i], \ \forall i \in V$$

Properties of the VRPTW: The VRPTW is easily seen as a generalization of the CVRP, since we can set the time windows $e_i = 0$ and $l_i = \infty$, $\forall i \in V$.

2.4 The Truck and Trailer Routing Problem (TTRP)

In the literature the use of trailers in vehicle routing problems has so far received little attention, even though it has a high practical application. In the TTRP, the use of trailers (a commonly neglected feature in the CVRP) is considered where customers are serviced by a truck pulling a trailer. However, due to practical constraints including government regulations, limited manoeuvring space at a customer site, road conditions etc. a subset of the customers may only be serviced by a truck.

Gerdessen gave two examples in [Ger96] of real-world applications. The first involved the distribution of dairy products by the Dutch dairy industry. Many customers were located in cities with heavy traffic and limited parking places, therefore the trailer pulled by a truck had to first be parked after which the truck serviced some customers along a certain route before returning to pick up the trailer again. Another case involved the delivery of compound animal feed to farmers. Because of narrow roads and/or small bridges on the delivery routes, various types of vehicles called double bottoms, consisting of a truck and a trailer, were commonly used.

Semet and Taillard give another application in [ST93]. Here 45 food chain stores in Switzerland were serviced by a fleet of 21 trucks and 7 trailers. The combined use of both trucks and trailers were therefore of great interest.

In the following, the model by Chao [Cha02] is presented since it is used by Scheuerer [Sch04], Yu et al. [YLC08], and as a foundation for this thesis.

We consider a fleet of m vehicles, of which m_1 consist of a truck coupled with a trailer, and $m - m_1$ trucks without trailers ($1 \leq m_1 \leq m$). A vehicle with a trailer is henceforth called a Complete Vehicle, whereas a truck without a trailer remains a truck. In this context, we consider trucks and trailers to be homogeneous. We let C_{truck} and $C_{trailer}$ denote the truck and trailer capacities respectively, thus the capacity of a Complete Vehicle is: $C_{truck}+C_{trailer}$, and the capacity of a truck is C_{truck}. Furthermore we let C^i_{truck} denote the capacity of the truck immediately before visiting customer i. Equally we define $C^i_{trailer}$ as the capacity of the trailer before visiting customer i. The cost structure described in section 2.2 still applies unchanged to this model, even though a homogeneous fleet is considered, hence the cost of a vehicle travelling with or without a trailer is the same.

Although every customer has a service time s_i associated with it as was the case in the VRPTW, no time window constraints are considered for the customers. The depot on the other hand does have a time window constraint, thus imposing a total distance/duration constraint on the constructed routes.

The TTRP considers two different kinds of customers: Customers which can be serviced by a complete vehicle or a truck alone are referred to as *Vehicle Customers* (VC), otherwise we refer to the customers as *Truck Customers* (TC) if it has to be serviced by a truck. We let $\epsilon(i)$ denote the customer type of customer i. Formally we have a partition of $V \setminus \{0\}$ into subsets $V_V \subseteq V$ and $V_T \subseteq V$ consisting of all vehicle customers and all truck customers respectively.

$$V = V_T \cup V_V \cup \{0\}$$

$$V_T \cap V_V = \emptyset$$

Since we have to consider different vehicle types in the TTRP, it makes sense to classify a route according to the vehicle serving it. A route is defined as a "Vehicle Route" (VR) if the assigned vehicle is a complete vehicle, otherwise the route is referred to as a "Truck Route" since it is serviced by a truck. We now consider the structure of a VR: The route may begin with a complete vehicle leaving the depot (i.e. the trailer is coupled), next the vehicle proceeds to service a number of VC customers, we refer to this part of the route as the *Maintour*. Thus the maintour of a VR represents a tour starting and ending at the depot in which strictly VC customers are serviced. However, during the VR it is possible for the vehicle to park its trailer at a parking place, and then proceed to service customers along a *Subtour*. On a subtour, both VC and TC customers can be serviced, since the vehicle does not have its trailer coupled. A subtour starts and ends at the parking place where the trailer was de-coupled. When the vehicle ends the subtour it re-couples the trailer and proceeds along the maintour. The edges comprising the maintour of a route are referred to as maintour edges, and similarly the edges in subtours are referred to as subtour edges.

It should be noted, that there is no restriction on the amount of subtours originating from a maintour, thus a VR may consist of exactly one maintour, no subtours, or any number of subtours. Naturally, the violation of the capacity constraints $C_{truck} + C_{trailer}$ of a VR is not allowed. Likewise, the capacity constraint C_{truck} of a subtour may not be violated. It is assumed that the transfer of cargo from truck to trailer and vice versa is possible, and that the cargo is homogeneous i.e. fluids etc. In addition, the transfer of cargo imposes no additional cost in the objective function.

Next we consider the structure of a TR: A Truck Route starts and ends at the depot. Since the vehicle does not include a trailer, both VC and TC customers may be serviced as long as the C_{truck} capacity constraint is not violated. A TR does not contain any subtours.

Whenever the vehicle visits a new customer it will service the customer from the truck alone if it is to be serviced as part of a subtour, or from the trailer or a combination of trailer and truck if the customer is serviced on the maintour.

The start and end node of a tour is defined as the *Root* of the tour. For a VR the root is the depot. For a subtour the root is the parking place where the trailer was de-coupled. It should thus be clear that truck routes and subtours only differ by their respective start and end nodes.

According to the Chao's model [Cha02], any VC customer as well as the depot may be chosen as a possible parking place. However, it should be noted that multiple uses of the same VC customer as a parking place in different routes is prohibited (this restriction does not include the depot). However, a parking place may be used multiple times within the same VR, provided the subtours originating from the parking place are serviced consecutively.

An exemplary TTRP routing plan can be seen in figure 2.1.

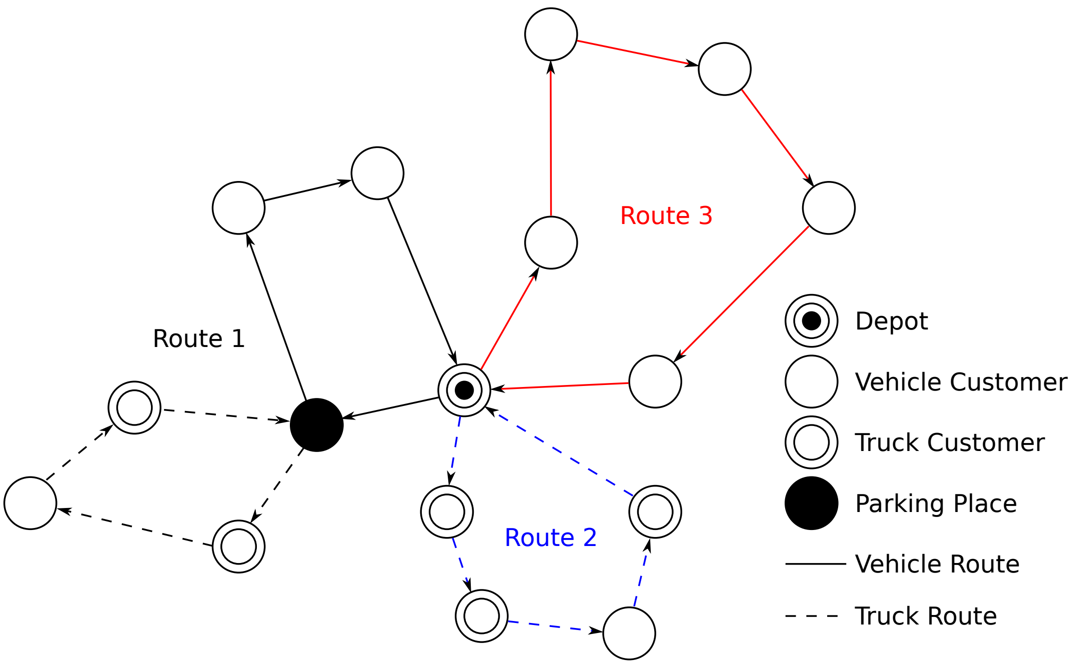


Figure 2.1: Exemplary TTRP Routing Plan

The objective of the TTRP is to construct a feasible routing plan which minimizes the total distance travelled of all vehicles given the constraints mentioned. In the problem instances created by Chao (see section 6.1), an upper bound was given for the number of available vehicles and trailers respectively. Violating this bound would result in an infeasible routing plan.

The TTRP expands on the complexity of the CVRP due to the following problems which have to be considered:

- Determining the optimal number of subtours per route.
- Determining the optimal parking places for the trailers.
- The assignment of customers to tours in which they can feasibly be serviced.

A precise mathematical model for the TTRP was developed by Scheuerer in his Ph.D. thesis [Sch04], but since the formulation is quite extensive it is not included in this thesis.

2.4.1 Properties of the TTRP

The TTRP generalizes the CVRP if we set all customer types to: $\epsilon(i) = TC$, and relax the time window for the depot to: $[0, \infty]$. In this case the TTRP can be solved as a regular CVRP without the use of trailers. Otherwise we can choose to set $\epsilon(i) = VC$, $\forall i \in V \setminus \{0\}$, and relax the time window for the depot as before and solve the problem as a regular CVRP using Complete Vehicles since we no longer need to park the trailers.

Like the Periodic Vehicle Routing Problem (PVRP) and the Multi-depot Vehicle Routing Problem (MDVRP), the TTRP can be seen as a multi-level optimization problem [Cha02]. At the first level every customer has to be assigned to a vehicle. At the second level every customer has to be assigned to a tour depending on its type. At the third level the optimal parking places have to be determined. Lastly the order or sequence the customers have to be serviced in has to be determined in a cost-minimal way according to the chosen objective function.

The TTRP is an interesting object of study since it also generalizes other well known optimization problems besides those listed earlier. If we let the parking places represent *locations*, then the TTRP can be seen as a Location Routing Problem. The location component therefore consists in determining the optimal location of the parking places. The routing component consists of determining the optimal subtours from parking places as well as the routes leaving the depot. The two components are strongly coupled.

We may also view the TTRP as a Multi-Trip Vehicle Routing Problem (MTVRP) [BM98] since a maintour may include several tours. Assume the vehicle fleet is limited to only consist of complete vehicles with limited truck capacity and unlimited trailer capacity. In addition, all customers are assumed to be truck customers (thus only the depot is a valid parking place). In this case every subtour starts and ends in the depot and the resulting vehicle route represents a corresponding MTVRP route.

Finally the TTRP includes elements from the Site Dependent Vehicle Routing Problem (SDVRP) [Nik09] since truck customers limit the allowed vehicle type and the Heterogeneous Fleet Vehicle Routing Problem (HFVRP or HVRP) [TV87], since vehicles have different capacities depending on the truck pulling a trailer or not, as well as different maximum capacities in maintours and subtours.

Chapter 3

The Extended TTRP (ETTRP)

In this chapter the new model treated in this thesis is presented. In section 3.2 the definitions and constraints unique to the new model is given, while section 3.3 discusses and states the main objectives valid for this thesis. Finally a short note on the hardness of the problem is given in section 3.4.

3.1 New Model Considerations

Although the model of Chao [Cha02] has certain practical relevance, it lacks a few real-life considerations. For instance, it is assumed that a vehicle is allowed to park its trailer at a customer before proceeding to service other customers along a subtour. In a real application however, this may not be acceptable for any number of reasons including the fact that a customer may simply not be interested in acting as temporary parking place. Furthermore, parking places are always considered to be located at an existing customer serviceable with a trailer, again one might also want to include the possibility of using parking places which physical location differs from any customer.

Naturally the limitations of the TTRP model is not to be considered a mistake, they simply serve as a means to keep the model simple. Keeping this in mind the term "Extended TTRP" (ETTRP) can be seen as a model which removes some of the limitations/simplifications of the TTRP model.

We now proceed to clarify precisely where the ETTRP deviates from the TTRP:

1. VC-customers are no longer considered feasible parking places[1].

2. A parking place may be used any number of times within the same or different routes.

It is now helpful to let P denote the set of available parking places, and let $P_{used} \subseteq P$ be the set of parking places used in the current routing plan, as well as let $P_{\cap used}$ denote the set of parking places used in several routes.

As was the case for the TTRP, the ordered sequence σ_i now signifies an ordered sequence $\hat{\sigma_i}$ in which every used parking place may appear an even number of times

[1] It is important to note, that a parking place may still be located physically at the same location as a vehicle customer, but that the model clearly distinguishes between parking places and customers.

within the same or different routes. Vehicle and Truck customers only appear once within any sequence.

As a consequence the constraint (2.2.5) of section 2.2 has to take this into account. An example of an exemplary ETTRP routing plan can be seen in figure 3.1.

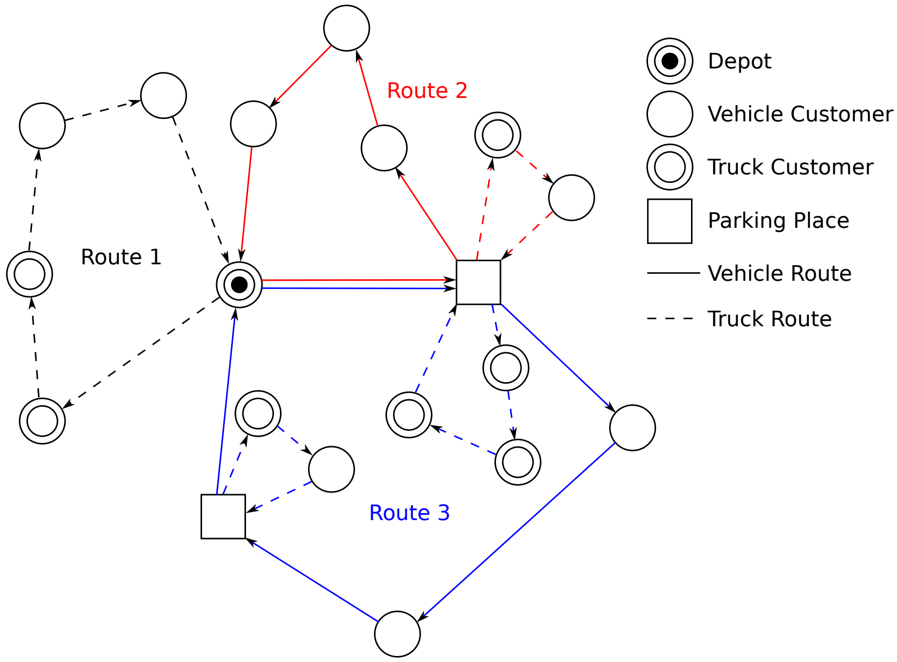


Figure 3.1: Exemplary ETTRP Routing Plan

It is still easy to see that the ETTRP is a generalization of the original TTRP. To see this, we add parking places at the same physical location as every vehicle customer $i \in V_V$ and restrict its usage to routes in which the customer i is also located.

In the initial problem description by Transvision A/S, presented in section 1.1 a number of possible complications might be considered. A subset of the possible complications were chosen to be included in this thesis and are presented in the following.

3.2 Additional Constraints

Time Windows: Time windows in the context of the VRP have many real-life applications, and therefore it also makes sense to apply the definitions to the ETTRP as well. As a consequence the model treated in this thesis also takes time windows and route duration as defined in section 2.3 into account. In addition to the normal time windows $[e_i, l_i]$ associated with every customer $i \in N$, we also define time windows $[e_p, l_p]$ for every parking place $p \in P$. These are needed in the context of local search (see chapter 5) for feasibility checks.

Load Constraints: Load constraints impose an interesting restriction upon the routes within a routing plan. The reasoning behind the constraint is that the cargo or load of a vehicle has to be distributed between the truck and the trailer being pulled, in such a way that the vehicle is able to safely utilize its brakes. In other words, it has to be ensured that the load of a truck always exceeds that of its trailer during the entire route. Furthermore, we prohibit the transfer of cargo from truck to trailer and vice versa, since this is not always applicable in a real-life situation.

The addition of load constraints, as well as time windows, still lets the ETTRP be seen as a generalization of the CVRP. By setting the time windows appropriately (see section 2.3), as well as setting the customer types $\epsilon(i)$ as was done in section 2.4, we effectively transform the CVRP into the ETTRP. Load constraints have no meaning in such a setting since a vehicle will either never have a trailer coupled or always have a trailer coupled.

3.3 Objectives

A few examples of vehicle routing problems have been presented in the previous sections, and since these may be applicable in many different scenarios, the objective of the optimization might also vary.

Minimizing total route length: A common objective is the minimization of the total length of the routing plan. In this case the main focus is to reduce the total travel costs without considering vehicle employment. Using the previous definition (2.2.9) of the length of a routing plan, the objective can be defined as follows:

$$\omega(R) = \min \sum_{k=1}^{K} L(R_k) \tag{3.3.1}$$

Minimizing number of employed vehicles, thereafter total route length: It might be the case that driver salaries, vehicle maintenance, as well as the purchase of additional vehicles contribute a significant cost, when considering the expenses of a company. Therefore, an objective in which we also consider the number of utilized vehicles might be considered. The objective thus becomes twofold: the primary objective is to reduce the number of vehicles used, the secondary objective is to minimize the total length of the routing plan. We define the objective as follows:

$$\omega(R) = \lambda \cdot K + \min \sum_{k=1}^{K} L(R_k) \tag{3.3.2}$$

The constant λ denotes the cost of employing one of the k vehicles. If λ is set sufficiently large, the focus of the optimization will be on minimizing the number of vehicles. Only when two routing plans employ the same amount of vehicles will the total length be relevant. Otherwise λ could be set to a value so that the decrease in overall route length makes up for the use of an additional vehicle.

The ETTRP objective: The objective considered in this thesis is that of minimizing the total route length (3.3.1). Therefore it is assumed that a sufficiently large fleet of vehicles is available, and that the extra cost associated with its maintenance is of negligible concern. In chapter 8, however, we may compare the results obtained by the implemented algorithms to those considering the objective 3.3.2 if the number of vehicles employed is the same.

The motivating factor for choosing the minimization of the total route length as an objective was the fact that both Chao, Scheuerer as well as Yu et al. used this objective in their models.

3.4 Hardness

In previous sections it was shown that the ETTRP could be seen as a generalization of the CVRP. The CVRP in turn could be seen as a generalization of the VRP which was also a generalization of the well known TSP. The decision version of the TSP has been proven to be NP-complete [CLRS01], and its optimization version is thus per definition NP-hard, hence by restriction the ETTRP is also NP-hard. In [Sol87] Solomon claims, that in 1984 Savelsbergh showed, that finding a feasible solution to the VRPTW with a fixed amount of vehicles was in itself NP-complete, although no reference to Savelsberghs proof could be found.

I 2008 a gap was closed by Jepsen et al. [JPSP] when they solved a few more of the previously unsolved VRPTW benchmark problems proposed by Marius M. Solomon [Sol84]. Very recently four more benchmark problems have been solved to optimality by Baldacci et al. [BBMR] leaving only a single instance unsolved at the time of this writing. However, considering the fact that the problems have been around for more than 25 years might indicate that these types of problems are in fact very hard to solve.

3.5 Summary of Model Characteristics

A brief overview of the characteristics of the previously presented models is shown in table 3.1

Model	Capacity constraints	Time windows	Heterogeneous customers	Parking places	Load constraints
CVRP	X				
VRPTW	X	X			
TTRP	X		X	X	
ETTRP	X	X	X	X	X

Table 3.1: The characteristics of each previously presented model.

Chapter 4

Known Algorithms for the TTRP and Related Problems

The purpose of this chapter is to describe various procedures developed in the literature for handling the construction of initial solutions to the TTRP. In addition some procedures for improving these solutions by the use of metaheuristics or otherwise will be presented. It is important to note, that some of the procedures are applied to TTRP definitions which might slightly deviate from the definition presented in this thesis. In short the purpose of the chapter is to include a brief survey of how the problem has been previously treated, and how it might possibly serve as an inspiration.

The chapter is concluded by a brief discussion about the various techniques described and a possible approach to solving the ETTRP.

4.1 Semet and Taillard

Semet and Taillard [ST93] describe a three step procedure to solve their definition of the TTRP. In the first step all VC customers ("trailer-stores") are assigned to maintours constructed by the Fisher and Jaikumar-Heuristic [FJ81], using all available trailers for the vehicles (which are to be interpreted as vehicles in this step). If the number of trailers is insufficient to service all the VC customers, additional trucks are allocated and assigned to trailers. In the next step every TC customer is assigned to its nearest VC customer, followed by a subsidiary local search procedure which tries to resolve violated capacity constraints without making existing feasible subtours infeasible. In the last step those remaining TC customers which have not yet been assigned, a VRP is solved with the remaining trucks available. The routes from the last step consist of trucks only. The authors claim that their procedure successfully constructs feasible routes when time windows are not taken into account [ST93, p. 472]. This claim, however, is questioned by Scheuerer [Sch04, p. 77].

4.2 Semet

Semet [Sem95] proposes a two step procedure based upon the method developed by Fisher and Jaikumar [FJ81]. In the first step customers and trailers are assigned simultaneously to vehicles, of which it is assumed that all are employed. In the second step a vehicle tour is constructed for every complete vehicle as well as a truck tour for every truck. The method focuses on the assignment problem of the first step, and proposes the use of a *Branch-and-Bound* algorithm where lower bounds are obtained by solving a *Lagrangian Relaxation*. For a complete overview please refer to [Sem95]. We limit the description to how the initial seed customers are chosen, and to how the cost-approximation of the assignment of customers is devised.

First the VC customer which is the furthest away from the depot is initially chosen as seed customer. Next those VC customers that maximize the equally weighted sum of the distance to all other seed customers as well as the distance to the nearest seed customer are successively added to the set S of seed customers. Every seed customer is assigned a distinct truck, and the number of chosen seed customers equals the number of available trucks. The cost of assigning a customer to a truck depends on the customers type. For VC customers the cost equals the cost of a fictitious tour from the depot to the customer. The cost of TC customers assigned to vehicles without trailers is calculated in the same manner. If, however, a TC customer is to be assigned to a truck pulling a trailer, then the cost is determined from the sum of the cost of a fictitious tour from the customer to any possible *root*, and the cheapest insertion of the *root* customer into the fictitious route from depot to the seed customer of this route. All possible root candidates are evaluated (with exception of the seed customer of this route) and the cheapest chosen. As a consequence the insertion cost of a TC customer once trailers are employed is higher, thus trying to limit the number of used trailers in general.

4.3 Gerdessen

Gerdessen describes three multi-step procedures for constructing an initial solution to what is referred to as the Vehicle Routing Problem with Trailers (VRPT) [Ger96]. Gerdessen also describes the application of several neighbourhoods usable for local search, but does not discuss any means of escape from a local optimum. The model by Gerdessen assumes that all complete vehicles park their trailers once. In addition a *manoeuvring time* is introduced as a measure associated with the inconvenience of visiting a customer. “Difficult” customers have a higher manoeuvring time.

Heuristic I: The first procedure initially solves a CVRP by ignoring the possibility of parking the trailer. In step one, the number of resulting routes is determined by selecting a number of seed customers equalling the total demand divided by the capacity of a vehicle (rounded up). In the next step the first seed customer is chosen as the one having the greatest distance from the depot. The remaining seed customers are chosen successively as the ones which maximize the distance to previously selected seeds. After this step a number of routes equal to the number of seed customers have been created, each consisting of two vertices, a seed customer and the depot. In the next steps every unassigned customer is added to a route in a maximum savings fashion. Now that the CVRP has been solved, the next step consists of solving the Travelling Salesman Problem with a Trailer (TSPT). This step essentially partitions each of the routes constructed from the CVRP into two separate routes according to a criteria based on cheapest insertion and manoeuvring time for each customer. One route corresponds to the vehicle route, and the other to a subtour. Finally the cheapest insertion of a customer from the vehicle tour into the subtour determines the parking place to use for connecting the tour routes. The TSPT is solved for each of the original CVRP routes.

Gerdessen points out that some of the constructed routes may have high manoeuvring times, and thus proposes the following heuristics as well:

Heuristic II: First, routes for the trucks are constructed by assigning the customers to trucks according to a criteria based on the customers distance to the depot and its manoeuvring time. Next, a parking place for each of the routes is determined by means of the cheapest insertion of one of the unrouted customers. The parking place then serves as a seed for a construction step equal to the one described in the first heuristic.

Heuristic III: The final heuristic is identical to the second heuristic except that before the parking places are determined both the truck routes and the vehicle routes have been constructed and improved by means of a local search procedure.

4.4 Chao

Chao [Cha02] constructs a TTRP routing plan by means of a three step procedure consisting of an assignment of customers to vehicles, construction of the routes and a local search procedure. The initial two steps use the idea of Fisher and Jaikumar [FJ81]. Hence the number of employed vehicles is predetermined and contrary to the procedure of Semet [Sem95] (see 4.2), the trailers are preassigned to a truck. For every vehicle with and without trailer, an initial seed customer is chosen. Following the choice of the first seed customer, the remaining seed customers are chosen successively as the ones for which the distance to the depot and all previously chosen seed customers is the greatest. Therefore an alternate choice of the first seed customer may lead to a different set of seed customers altogether resulting in different starting solutions. From a number of generated starting solutions, the best is chosen. As initial seed customer Chao chooses the customer having the greatest distance to the depot. For the next initial seed the customer having the second longest distance to the depot is chosen and so forth. The assignment cost are determined regardless of the customer type (as was the case in the procedure by Fisher and Jaikumar [FJ81]). A relaxed version of the assignment problem is solved and the customers are assigned to vehicles by means of a heuristic, which may lead to violated capacity constraints. Next the routes are constructed according to the assignment. For vehicles without trailers this is solved as a regular routing problem. For vehicles with trailers this is solved for all assigned VC customers, after which remaining TC customers are successively added to new or existing subtours of the maintour. During the construction of the subtours the capacity constraint is never violated. Every route is then subjected to a local search procedure which tries not only to reduce the cost of the routing plan, but also to reduce any capacity violation which may exist. The local search procedure ends when the solution cannot be improved any further.

4.5 Scheuerer

Scheuerer [Sch04] introduces the first pure IP-formulation for the TTRP based on three-index arc variables for maintours, as well as a five-index variable indicating whether a certain truck traverses a certain arc on the n^{th} subtour starting at a certain customer or at the depot, where n is the number of customers (in the worst case, as many subtours starting at a VC customer or the depot are necessary as there are customers). Semet [Sem95] uses similar variables, but requires that at most one subtour originates at each VC customer, so that the second variable only has four indices in his formulation.

Scheuerer developed two methods for the construction of an initial solution to the TTRP, both based on heuristics.

T-Cluster: T-Cluster can be seen as a cluster based sequential insertion heuristic, in which customers are successively added to routes until either the capacity constraint or the maximum duration of a route would be violated. Each route is initialized by the seed customer which has the greatest distance to the depot, and with the vehicle having the largest capacity (thus routes using complete vehicles are constructed first). The seed customer is added to a fictitious route to the depot, and depending on the customers type it is either inserted on the maintour, or as a subtour originating from the depot. For vehicles without trailer this distinction no longer applies. Next customers are successively chosen according to a clustering criteria [Sch04, p.81] and added to the current route using cheapest insertion. If there are still unserviced customers at the end of the construction of the final route, these are added even though this might lead to a violation of both capacity constraints as well as route duration. Then the routes are subjected to a local search procedure to reduce overall costs. It is attempted to improve the root of every subtour using a *Subtour Root Refinement* procedure resembling that of Chao. Several initial solutions are generated by modifying the clustering criteria for each run, after which the best solution is chosen.

T-Sweep: T-Sweep is based on the well known *Sweep* [GM74] algorithm for the VRP. Routes are constructed by projecting a beam from the depot and rotate the beam (or sweep line) in a clockwise fashion thus adding customers as the beam hits them. Routes are first constructed using complete vehicles, then trucks. In case the insertion of a customer would violate either the capacity constraints or the route duration, a new route is initialized thus adding the customer to the newly created route. As was the case for the T-Cluster heuristic, the actual insertion of a customer is performed according to cheapest insertion. In case there are still unserviced customers during the construction of the last route, the capacity constraints as well as route duration constraints are relaxed in the same way as for T-Sweep.

Of the two construction heuristics Scheuerer concludes that T-Cluster performs the best.

Scheuerer then proceeds to improve the solution using a sophisticated metaheuristic based on the well known *Tabu Search* algorithm of Glover [G$^+$89][Glo90], using long, and short term memories. Infeasibility is handled using a *Shifting Penalty* term [GHL94], and the neighbourhoods of the search are restricted in order to speed up the procedure.

4.6 Yu et al.

Yu et al. [YLC08] describe the application of the Simulated Annealing (SA) [KGV83] heuristic to the TTRP. They use a special solution representation in order to represent the entire routing plan very efficiently. An initial solution is generated randomly, consisting of a sequence of the customers as well as delimiters in the solution representation which denote the beginning of a new route. The used representation guarantees that the generated routes are feasible with respect to capacity constraints. However, it is unclear if the representation guarantees feasible route duration. The number of employed vehicles may exceed the number of available vehicles, at which point a route combination procedure tries to merge routes in order to minimize the number of required vehicles. The route combination procedure continues until no more routes may feasibly be merged (with respect to capacity constraints). If the number of available vehicles is still exceeded, a penalty is added to the objective function in order to make such solutions unattractive. The neighbourhoods considered during a step of the SA procedure include the possibility of swapping two customers, insertion of a customer before another customer, and changing a customers type of service vehicle. The procedure includes the possibility of choosing the best of N-trials, using the neighbourhoods described above. Every three temperature reductions, a local search procedure which sequentially performs 2-Opt, swap, insertion, and change of vehicle type is used to improve the current best solution. For an efficient implementation of these neighbourhoods the reader is referred to [Irn08a] and [IFG06]. The neighbourhoods implemented in this thesis are described in detail in section 7.3.

The procedure ends when the temperature reaches a certain predefined threshold. The authors report improved solution qualities as well as shorter running times compared to the Tabu Search heuristic by Scheuerer (see 4.5).

In a note [LYC] the authors introduce the Relaxed TTRP (RTTRP), in which the number of employed vehicles and trailers is relaxed, thereby allowing their procedure to improve upon the obtained solutions from the TTRP.

4.7 Drexl, TTRP

Drexl [Dre07] introduces the first exact methods to solve the TTRP. His Mixed Integer Programming (MIP) formulation is based on a *time-space-operation network*, which is a simplified version of the time-space-operation-vehicle network he developed for the Vehicle Routing Problem with Trailers and Transshipments (VRPTT). Drexl describes two different formulations, one based on Path Variables, and the other based on Arc Variables.

It is important to note that Drexl considers optional parking in his TTRP model, thus parking places are not necessarily associated with a VC customer.

Two exact algorithms are compared for solving the TTRP, one is a branch-and-cut algorithm, while the other is a branch-and-price algorithm. For details on cut generation as well as branching strategies etc., the reader is referred to [Dre07].

Drexl concludes that the branch-and-price algorithm is vastly superior to the branch-and-cut algorithm. It is able to solve instances which are one order of magnitude larger than those solvable by the latter.

The largest instances that could be solved to optimality comprised 20 customers, 20 transshipment locations (parking places) and no time windows, using in excess of eight hours computational time.

4.8 Drexl, VRPTT

The Ph.D. thesis by Drexl also includes a description of the Vehicle Routing Problem with Trailers and Transshipments (VRPTT) [Dre07]. This model is a generalization of the TTRP model and is vastly more complicated. However, the model is of particular interest when considering some of the extra possible complication posed by Transvision A/S (see chapter 1, section 1.1), most notably by the possibility of a trailer being pulled by different trucks. For an in-depth explanation of the model please see [Dre07]. Drexl presents a MIP for the complete problem based on Turn Variables.

Two simplified MIPs based on a time-space-operation-vehicle network using arc variables, as well as one using path variables are shown, capturing only the “core” of the problem.

Drexl concludes that an implementation of a branch-and-price algorithm for the solving the simplified VRPTT is not straight forward due to two properties of the Resource Extension Functions (REFs) [Irn08b] being violated. Instead a branch-and-cut algorithm is implemented and tested.

The computational results show that even small instances are extremely hard to solve to optimality. Drexl reports that the largest instances that could consistently be solved had 4 customers and 4 transshipment locations (parking places) and 4 vehicles.

4.9 Remarks on Existing Methods

In this chapter a few methods for solving the TTRP were presented. Only Drexl truly attempts to solve the problem to optimality using exact algorithms, and while doing so, was only able to solve problem instances of a somewhat limited size. All other methods are based on heuristic approaches of varying degree of sophistication.

The goal is to at least be able to find feasible solutions to the benchmark instances developed by Chao [Cha02]. The smallest of those instances are comprised of at least 50 customers, thus justifying the use of a procedure based on a heuristic approach. With this in mind we continue to describe the programming model and the solution representation on which this procedure will be based.

Chapter 5

Modelling the ETTRP

In this chapter we define the programming model and describe the solution representation on which the heuristic approach for solving the ETTRP will be based. In the first section we introduce various definitions and notation in order to better describe the elements of the programming model. We start by considering an example which includes aspects of what we would essentially like our model to be able to express. The programming model should ideally be able to handle the constraints imposed by the TTRP model, as well as the additional constraints defined in section 3.2.

Let us consider the routing plan in figure 5.1. It is essentially identical to figure 3.1, except that customer numbers have been added to nodes, as well as the number of the vehicle serving a specific customer. Although not directly visible in the figure, the depot is assumed to have the customer number 0. We will use this routing plan as a basis for the definitions that follow.

5.1 Parking Places

Recalling the relaxed definition of the ETTRP (see section 3.1) in terms of parking places we continue to describe how these are represented within the solution representation. Let us consider the sequence representing route two of figure 5.1:

$$\sigma_2 = \{0_s, 17, 10, 14, 17, 12, 1, 5, 0_e\}$$

Given the figure and the sequence above it is clear that the number 17 in the sequence represents a parking place (since it is not the depot 0), and only parking places are allowed to be visited several times. It should be clear that arriving at 17 the first time symbolizes the de-coupling of the trailer, before the truck serves the customers 10 and 14 in a subtour. The second visit of 17 thus naturally symbolizes the re-coupling of the trailer, before the truck continues on its maintour to customer 12. This obviously leads us to the following rules regarding parking places:

Rule 1: Every use of a parking place p implies two occurrences p_s and p_e of p in the representation, of which both p_s and p_e may inherit the characteristics of p or define new (except location). In principle this allows us to model various aspects of

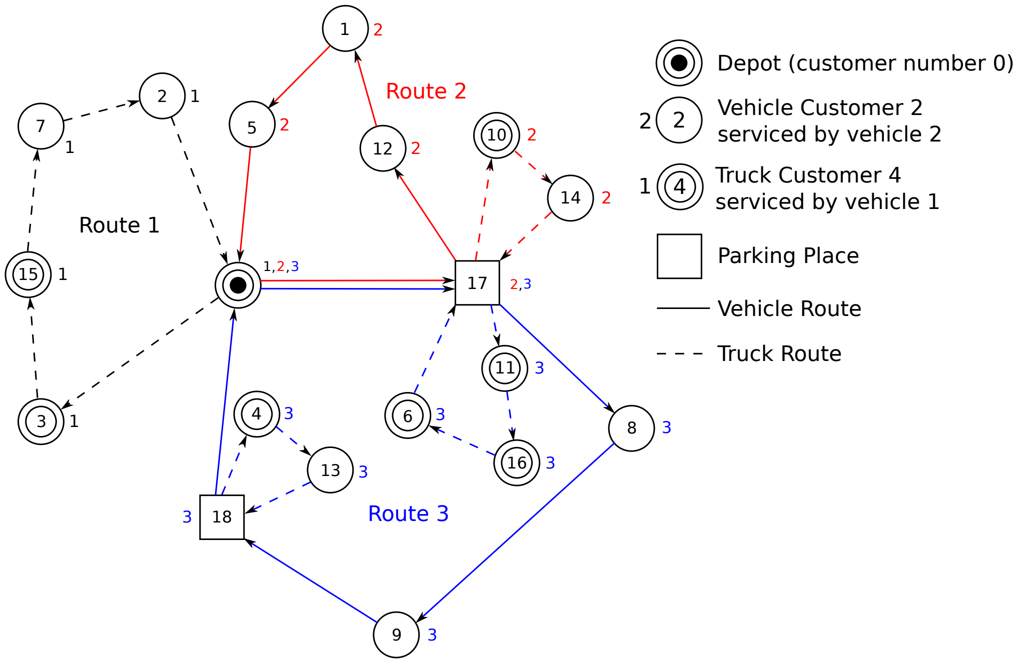


Figure 5.1: Example ETTRP Routing Plan

real-world applications. For example we may choose to associate a service duration with every parking place which may be interpreted as the time needed to de-couple and re-couple the trailer every time the truck visits the parking place.

Rule 2: For every pair p_s and p_e of p, we impose an ordering by letting p_s and p_e denote the parking start (de-coupling) and parking end (re-coupling) respectively. Therefore a vertex p_s must be visited before p_e.

Rule 3: No p'_s or p'_e of a parking place p' may occur between another parking place p_s and p_e of p.

Rule 4: No p_s or p_e of a parking place p may occur in a truck route.

Rule 5: p_s and p_e may not occur consecutively in the representation.

The above rules may be interpreted as follows: Rule 1 implies that for every used parking place we should both de-couple and re-couple the trailer. Rule 2 imposes a temporal ordering specifying that we must de-couple a trailer before we may re-couple it. Rule 3 simply implies that we do not allow subtours within subtours. Rule 4 implies that we do not allow subtours in truck routes. Finally, rule 5 states that an unused parking place should not be present in the solution representation. This follows naturally since parking places are no longer forced to be located at customers, and therefore visiting such parking places unnecessarily may be assumed to increase the value of the objective function. If p_s and p_e occur consecutively in the representation the solution is to simply delete the parking place altogether.

Using the above, we may instead write the sequence for route two as follows:

$$\sigma_2 = \{0_s, 17_s, 10, 14, 17_e, 12, 1, 5, 0_e\}$$

From this it should be clear that the above definitions allow us to represent a parking place an arbitrary number of times within the solution representation.

5.2 Earliest Departure (δ) and Latest Arrival (ψ)

In section 2.3 we introduced two helpful variables associated with each customer in a routing plan, the earliest departure δ_i and the latest arrival ψ_i of a customer $i \in V$. These variables are useful since they can help determine the amount of time slack a customer permits. In other words the variables denote the earliest and the latest a customer may be serviced without violating other parts of the route. These definitions are similar to those of Markus [GM09], but variations of these can be found in the literature most notably in Solomon [Sol87] and the use of *push-forward* for fast feasibility checks as well as Savelsbergh in [Sav92] using *time-slack* for minimizing route duration. In the master's thesis by Nikolajsen [Nik09] both slack-time and push-forward were used.

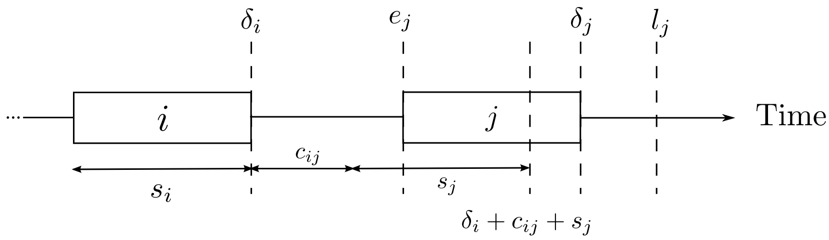
The earliest departure of a customer j is naturally dependent upon the customers serviced prior to servicing j. As usual we assume that a vehicle travels directly from customer i to customer j where c_{ij} is the direct travel time between i and j, a natural recursive formula for the computation of δ_j follows:

$$\delta_j = \begin{cases} e_j & \text{if } j = 0 \\ \max\{e_j, \delta_i + c_{ij}\} + s_j & \text{otherwise} \end{cases} \tag{5.2.1}$$

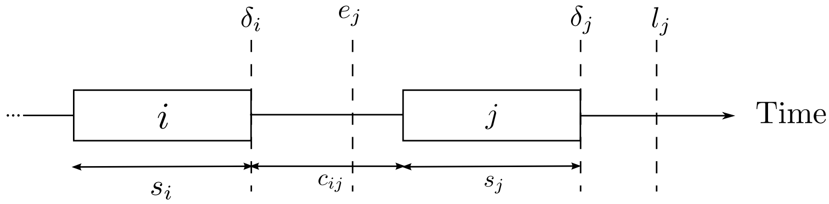
The depot, i.e. the case of $j = 0$ the earliest possible time of departure equals the earliest time of service e_0. For regular customers the earliest departure depends on two things. If service at a previous customer was finished early enough for the vehicle to arrive at customer j before e_j, then service is delayed until e_j, otherwise we arrive later and start service immediately. In both cases the service time is s_j yielding δ_j. The first situation is depicted in figure 5.2(a), the latter in figure 5.2(b).

Similarly, the latest arrival ψ_i of customer i depends on all customers following customer i. We use this in order to determine the latest we may arrive at i in order to still be able to serve the remaining customers of the route, i.e. how much the visit at customer i may be shifted forward in time without causing temporal violations. The latest a vehicle has to return to the depot is known in advance, namely l_0. As before, we define the latest arrival at customer i in a recursive manner as follows:

$$\psi_i = \begin{cases} l_i & \text{if } i = 0 \\ \min\{l_i, \psi_j - c_{ij} - s_i\} & \text{otherwise} \end{cases} \tag{5.2.2}$$



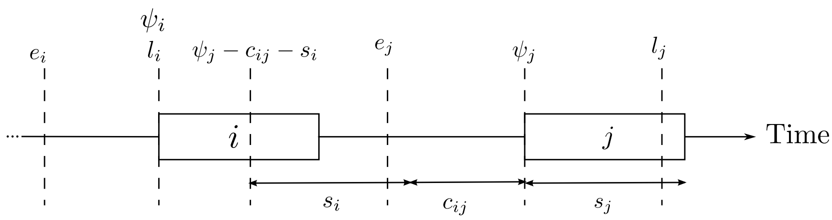
(a) $\delta_j = e_j + s_j$



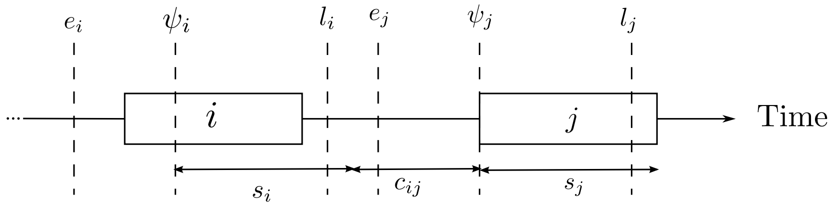
(b) $\delta_j = \delta_i + c_{ij} + s_j$

Figure 5.2: Visualizations of Earliest Departure δ.

The minimum function makes sure that customer j following customer i will be reached before its latest arrival time, and that the time window for customer i is not violated doing so. The two scenarios are depicted in figure 5.3(a) and 5.3(b).



(a) $\psi_i = l_i$



(b) $\psi_i = \psi_j - c_{ij} - s_i$

Figure 5.3: Visualizations of Latest Arrival ψ.

5.3 Segment (ς)

Before defining a *segment* we need to consider figure 5.1. From the figure we can easily devise the ordered sequences representing the tours of the routing plan, thus we have the following:

$$\sigma_1 = \{0, 3, 15, 7, 2, 0\}$$
$$\sigma_2 = \{0, 17_s, 10, 14, 17_e, 12, 1, 5, 0\}$$
$$\sigma_3 = \{0, 17_s, 11, 16, 6, 17_e, 8, 9, 18_s, 4, 13, 18_e, 0\}$$

A *segment* $\varsigma_k^{i,j}$ where $i, j \in |\sigma_k|, i \leq j$, defines a subsequence of customers from index i to j of route k. For example:

$$\varsigma_3^{2,6} = \{17, 11, 16, 6, 17\}$$

defines a segment corresponding to the first subtour of route 3. In other words a route may be seen as a concatenation of segments:

$$\sigma_3 = \varsigma_3^{1,1} \otimes \varsigma_3^{2,6} \otimes \varsigma_3^{7,8} \otimes \varsigma_3^{9,12} \otimes \varsigma_3^{13,13}$$

Segments are particularly useful since a segment may be interpreted as a single customer in terms of the total demand, as well as the time window associated with a particular segment. In the following we extend the concept of time windows to include segments.

It should be stressed that in the context of this thesis segments are only used to describe subtours.

5.3.1 Extending the Temporal Definitions to include Segments

Viewing a segment as single customers has a certain appeal during the initial construction of a solution for the ETTRP. As such we extend the concept of time windows to include segments in this section, while the details regarding creating a solution to the ETTRP is deferred to chapter 7.

Let i_s and i_e denote the start customer and end customer of a segment ς. Then it follows that the time window associated with ς may be calculated as follows:

$$[e_\varsigma, l_\varsigma] = \left[\min\{e_{i_s} + \sum_{i \in \varsigma} w_i, \psi_{i_s}\}, \psi_{i_s} \right] \tag{5.3.1}$$

Similarly the service duration of a segment ς may be calculated from:

$$s_\varsigma = a_{i_e} + s_{i_e} - e_\varsigma \tag{5.3.2}$$

In (5.3.1) e_ς is comprised of two terms. The first tries to delay the start of service for the segment until the minimum waiting time is achieved, the second makes sure that the start of service of the first customer of the segment is not violated. The upper bound on the time window for the segment is naturally ψ_{i_s} equalling the latest start of service for the first customer of the segment.

5.4 Load Balance (γ)

As mentioned previously, the model has to handle load constraints, and in order to do so, additional information needs to be stored in the model. Thus in addition to the normal data stored for each customer, we also store the current load of the truck as well as the load of its trailer (if any) at the time when the vehicle services a particular customer, or visits a particular parking place.

When modelling the traditional VRP it may be intuitive to consider the load of a vehicle to be maximal when it leaves the depot in order to be able to service as many customers as possible. However, this is not necessarily the ideal approach in the ETTRP setting since we now have to consider the load constraints. Consider for example a case where the capacity of the truck and the capacity of the trailer are the same. Here it would be illegal in terms of capacity constraints to service any customer in a subtour first, since this would lead to the truck becoming lighter than the trailer. If we instead model the capacities of the truck and trailer in a dynamic fashion by allowing both truck and trailer to be empty when they leave the depot we avoid this problem. In this new setting demands are added to the truck and to the trailer depending on the type of service the customer requires. All demands originating from truck customers must be placed on the truck explicitly, while demands from vehicle customers are allowed to be placed on both truck and trailer. Thus we propose to balance the loads in such a way as to minimize the usage of the trucks capacity (for later use in subtours), while still keeping the load constraints satisfied. In the following, the concept is illustrated by assuming routes are constructed by means of a sequential insertion heuristic.

Assume the capacity of a truck $Q_{truck} = 50$ and the capacity of a trailer is $Q_{trailer} = 30$. We let $\gamma = \binom{a}{b}$ denote the load of the truck a and the trailer b prior to visiting the corresponding customer respectively. Initially upon route construction both are initialized to 0 and the route is obviously empty:

$$\sigma = \{0_s, 0_e\}$$
$$\gamma_\sigma = \left\{\binom{0}{0}, \binom{0}{0}\right\}$$

Upon insertion of a vehicle customer 2 having demand $d_2 = 7$ we update the stored information according to the following:

$$\sigma = \{0_s, 2, 0_e\}$$
$$\gamma_\sigma = \left\{\binom{4}{3}, \binom{4}{3}, \binom{0}{0}\right\}$$

Next we insert vehicle customer 5 having demand $d_5 = 5$ and we update the load as follows:

$$\sigma = \{0_s, 2, 5, 0_e\}$$
$$\gamma_\sigma = \left\{\binom{6}{6}, \binom{6}{6}, \binom{3}{2}, \binom{0}{0}\right\}$$

Next assume we insert a truck customer 3 between customers 2 and 5 having a demand $d_3 = 4$ thus we update our stored load information as follows:

$$\sigma = \{0_s, 2, p_s, 3, p_e, 5, 0_e\}$$
$$\gamma_\sigma = \left\{\binom{8}{8}, \binom{8}{8}, \binom{7}{2}, \binom{7}{2}, \binom{3}{2}, \binom{3}{2}, \binom{0}{0}\right\}$$

From the above it follows that the insertion (or removal) of a customer into a route re-balances the load distribution on both truck and trailer. Insertion of a truck customer immediately adds capacity unto the truck while adding a vehicle customer either adds capacity to the trailer alone, or re-balances the load from the insertion point towards the start of the route. When every route is constructed the immediate load distribution may be obtained by examining load distribution in the start depot 0_s. In the above example we see that both the truck and the trailer have 8 units of cargo loaded at the depot 0_s, equalling the total demand of the route of 16. Upon visiting the first vehicle customer 2 we service it by unloading 6 units from the trailer and 1 unit from the truck equalling its demand $d_2 = 7$. Hence a customer i is serviced by unloading $\gamma_\sigma^i - \gamma_\sigma^{i+}$ units from truck and trailer. We may define the load distribution using backwards recursion in the following manner:

$$\gamma_i = \begin{cases} \binom{0}{0} & \text{if } i = 0_e \\ \binom{\gamma_{i+}+d_i}{\gamma_{i+}} & \text{if } i \text{ is subtour serviced} \\ \binom{\gamma_{i+}}{\gamma_{i+}+d_i} & \text{if } d_i + C^i_{trailer} \leq C^i_{truck} \\ \begin{pmatrix} \left\lceil \gamma_{i+} + \frac{d_i}{2} \right\rceil \\ \left\lfloor \gamma_{i+} + \frac{d_i}{2} \right\rfloor \end{pmatrix} & \text{otherwise} \end{cases} \tag{5.4.1}$$

It is important to note that when servicing from the truck or the trailer, only integral amounts may be unloaded.

5.5 Useful Functions

Previously, δ and ψ were defined as *earliest departure* and *latest arrival* respectively. Now we consider how the definitions prove their usefulness in the context of feasibility checks. Preferably such a check should be computable in constant time $\boldsymbol{O}(1)$ since in general we perform many more feasibility checks than actual update of route specific data such as arrival times etc.

5.5.1 Feasibility of Insertion

In the context of local search, it is often important to determine if the insertion of a customer i into route R is feasible or not. To determine this, we need to consider both temporal feasibility as well as load constraints.

Temporal: First we consider the temporal feasibility of inserting a customer c between customers $\alpha(e)$ and $\beta(e)$ on the edge e. Algorithm 1 is able to decide whether the insertion will cause any time window violations for the customers in the route in question. First we compute the earliest possible departure δ_c of customer c provided it is inserted after $\alpha(e)$ (line 1). Similarly we compute the latest arrival ψ_c of c provided it is inserted before $\beta(e)$ (line 2). The amount of temporal slack left by the insertion of c to customers preceding the insertion point, and customers following the insertion point is computed in lines 3 and 4. In line 5 we accept the insertion provided that both slack computations are non-negative. Assuming that the needed values δ and ψ used in the algorithm have been precomputed it should be clear that the temporal feasibility of insertion can be checked in $\boldsymbol{O}(1)$ time.

Algorithm 1: isTimeInsertable

Data: Customer c, Edge e
Result: true if c is insertable, false otherwise

1 initialize δ_c; `// calculated as if` c `was inserted after` $\alpha(e)$
2 initialize ψ_c; `// calculated as if` c `was inserted before` $\beta(e)$
3 forwardSlack $\leftarrow \psi_{\beta(e)} - (\delta_c + d(c, \beta(e)))$;
4 backwardSlack $\leftarrow \psi_c - (\delta_{\alpha(e)} + d(\alpha(e), c))$;
5 **return** ((forwardSlack ≥ 0) $\wedge$ (backwardSlack ≥ 0));

Load based: Next we consider the feasible insertion of customer c considering only load based constraints. Algorithm 2 shows how this can be accomplished given a customer c to be inserted on the edge e. Similarly to Algorithm 1 we check if customer c can feasibly be inserted between $\alpha(e)$ and $\beta(b)$.

To preserve space only the first part of the algorithm which deals with the insertion of a customer on a subtour edge is shown. However, the implementation of the remaining part of the algorithm dealing with insertion of a customer on a maintour edge is straight forward.

The customers insertion point is a subtour edge (checked in line 4), and we immediately determine if we have enough capacity left on the truck to service it directly (line 5). If there was enough capacity left, we end the insertion check by issuing a call to algorithm 1 in order to make sure that temporal constraints are also satisfied by the insertion. In lines 7-9 we determine how much of the extra capacity may be loaded unto the truck after which we determine if the load in the remainder of the route serviced from the truck may be serviced from the trailer instead. $C^{\alpha(e)}_{truckSum}$ in line 11 is the sum of demands serviced from the truck at vehicle customers from 0_s forwards until $\alpha(e)$. Thus we check if some of the demands of previously serviced vehicle customers may be shifted unto the trailer instead.

The algorithm ends with a call to Algorithm 1 which makes sure that the insertion of customer c on edge e will not cause a temporal violation in the route, hence the coupling of Algorithm 1+2 ensures us both temporal and load based feasibility. Furthermore it is clear that since algorithm 2 runs in $\boldsymbol{O}(1)$ the total feasibility of

Algorithm 2: feasibleInsertion

Data: Customer c, Edge e

Result: true if c is insertable, false otherwise

demand $\leftarrow d_c$;
if demand $+ C^{0_s}_{truck} + C^{0_s}_{trailer} > C_{truck} + C_{trailer}$ **then**
  return false;
if *e is a subtour edge* **then**
  if $C^{0_s}_{truck} +$ demand $\leq C_{truck}$ **then**
    return isTimeInsertable(c, e);
  if $C^{0_s}_{truck} +$ demand $> C_{truck}$ **then**
    truckCapLeft $\leftarrow C_{truck} - C^{0_s}_{truck}$;
    if $C^{0_s}_{trailer} +$ demand $-$ truckCapLeft $> C_{trailer}$ **then**
      return false;
    if $C^{\alpha(e)}_{truckSum} >$ demand $-$ truckCapLeft **then**
      return isTimeInsertable(c, e);
    return false;
else // e is a maintour edge
  ⋮
return isTimeInsertable(c, e);

inserting customer c can be decided in $\boldsymbol{O}(1)$.

5.5.2 Updating Load Information

Algorithm 2 depends on load information for both truck and trailer being available at the time the feasibility of inserting a customer is checked. This in turn means that we have to update the variables: C^i_{truck}[1], $C^i_{trailer}$[2], every time an insertion or a removal of a customer has been done. We perform the updating trivially using the following recursive algorithm 3, which is self explanatory.

Since the load based feasibility check depends on the availability of the variable $C_{truckSum}$ and since it is likely to change whenever an insertion, or a removal has been done, we use algorithm 4 to update its value after an insertion or removal of a customer.

Algorithm 4 is self explanatory except that we naturally have to distinguish the updating of the variable based on the service type of the customer c.

[1] Capacity of the truck immediately before servicing customer i

[2] Capacity of the trailer immediately before servicing customer i

Algorithm 3: updateBackwardDemands

Data: Customer c

Result: Updated values of C^c_{truck}, $C^c_{trailer}$ from c backwards until depot 0_s is reached

if $c \neq null$ **then**
  if *route is a truck route or c is subtour serviced* **then**
    $C^c_{truck} \leftarrow C^{c+}_{truck} + d_c$;
    $C^c_{trailer} \leftarrow C^{c+}_{trailer}$;
  else
    $C^c_{truck} \leftarrow$ re-balance according to (5.4.1);
    $C^c_{trailer} \leftarrow$ re-balance according to (5.4.1);
  return UPDATEBACKWARDDEMANDS(c^-)

Algorithm 4: updateForwardCapacities

Data: Customer c

Result: Updated values of $C^c_{truckSum}$ from c onwards until depot 0_e is reached

if $c \neq null$ **then**
  if *route is a truck route* **then**
    $C_{truckSum} \leftarrow 0$;
  else
    if *c is a vehicle customer and is maintour serviced* **then**
      $C_{truckSum} \leftarrow C^c_{truck} - C^{c+}_{truck} + C^{c-}_{truckSum}$;
    else
      $C_{truckSum} \leftarrow C^{c-}_{truckSum}$;
  return UPDATEFORWARDCAPACITIES(c^+)

5.5.3 Updating Temporal Information

Whenever a customer c is inserted or removed from a route we may need to recompute the variables δ and ψ used for temporal feasibility checks. Specifically the earliest departure δ may have changed for visits in the route from customer c onwards. Likewise, the latest arrival ψ may have changed for the visits in the route prior to customer c.

Algorithms 5 and 6 show how this can be done. Both algorithms recompute the waiting time w_i, arrival time a_i and the beginning of service s_i of a customer $i \in V$. Since these computations can be done in constant time $\boldsymbol{O}(1)$ the overall running time of both algorithms remains linear in the number of customers of the route in question $\boldsymbol{O}(|\sigma|)$.

5.5.4 Insertion and Removal

In the previous sections we covered how to update temporal as well as variables concerning load due to the insertion or removal of a customer c. Next, we briefly

Algorithm 5: updateEarliestDeparture

Data: Customer c

Result: Updated values of δ_c, w_c, a_c, and b_c from c onwards until depot 0_e is reached

if $c \neq null$ **then**
- $\delta_c \leftarrow \max\{\delta_{c^-} + d(c^-, c),\ e_c\} + s_c$; // (5.2.1)
- $a_c \leftarrow b_{c^-} + s_{c^-} + d(c^-, c)$; // (2.3.1)
- $b_c \leftarrow \max\{e_c,\ b_{c^-} + s_{c^-} + d(c^-, c)\}$; // (2.3.2)
- $w_c \leftarrow \max\{b_c - a_c,\ 0\}$; // (2.3.3)
- **return** UPDATEEARLIESTDEPARTURE(c^+)

Algorithm 6: updateLatestArrival

Data: Customer c

Result: Updated values of ψ_c, w_c, a_c, and b_c from c backwards until depot 0_s is reached

if $c \neq null$ **then**
- $\psi_c \leftarrow \min\{\psi_{c^+} - d(c, c^+) - s_c,\ l_c\}$; // (5.2.2)
- **if** $c^- \neq null$ **then**
  - $a_c \leftarrow b_{c^-} + s_{c^-} + d(c^-, c)$; // (2.3.1)
  - $b_c \leftarrow \max\{e_c,\ b_{c^-} + s_{c^-} + d(c^-, c)\}$; // (2.3.2)
  - $w_c \leftarrow \max\{b_c - a_c,\ 0\}$; // (2.3.3)
- **return** UPDATELATESTARRIVAL(c^-)

include algorithms coupling all those aspects and show how insertions (algorithm 7) and removals (algorithm 8) can be done.

It is assumed that the feasibility of insertion is checked for validity before calling either algorithm respectively.

The algorithms 1-8 are used extensively as subprocedures for larger algorithms implemented in this thesis, justifying their relevance and inclusion.

Having described the ETTRP model in terms of objectives as well as an in-depth explanation of how to model the problem including several useful algorithms for sub-procedures, we now proceed to describe several of the problem instances on which the implemented algorithms were tested.

Algorithm 7: insert

Data: Customer c and Edge e

insert c between $\alpha(e)$ and $\beta(e)$;
updateEarliestDeparture(c);
updateLatestArrival(c);
updateBackwardDemands(0_e^-);
updateForwardCapacities(0_s^+);

Algorithm 8: remove

Data: Customer c

insert c between $\alpha(e)$ and $\beta(e)$;
updateEarliestDeparture(c^-);
updateLatestArrival(c^+);
updateBackwardDemands(0_e^-);
updateForwardCapacities(0_s^+);

Chapter 6

Instances

In order to test the algorithms implemented in this thesis several categories of instances have been used.

Since the implemented algorithms are able to generate solutions to the TTRP, a suitable set of benchmark instances created by Chao [Cha02] were used in order to assess the quality and performance of the algorithms in that context. In section 6.1, a detailed description of these benchmark instances will be presented.

In addition, since the algorithms are also designed to be able to generate feasible solutions to the VRPTW, the benchmark instances of Solomon [Sol87] were used to assess the quality and performance of the algorithms in this respect as well. The Solomon benchmark instances will be discussed in greater detail in section 6.2.

Finally, a new category of instances were generated from the Solomon instances to encompass both aspects of the VRPTW and the TTRP, thus becoming ETTRP instances. We describe the process of creating these new instances in section 6.3.

All instances treated in this thesis are considered offline, meaning that all customers including their demands are known in advance.

6.1 Chao Benchmarks (TTRP)

Due to the lack of available benchmark instances for the TTRP, Chao [Cha02] selected seven basic VRP benchmark instances created by Christofides, Mingozzi and Toth (CMT) [CMT79] and used these as generators for the TTRP instances.

The TTRP instances were generated the following way. For each customer i in the CMT problem, the distance between i to its nearest neighbour customer is calculated and this is denoted by A_i. From each of the seven CMT benchmark instances three TTRP benchmark instances were created. In the first problem, 25% of the customers with smallest A_i values are specified as truck customers. In the second problem, this is increased to 50%, and in the third problem, it is 75%. The numbers and capacities of available trucks and trailers for all problems and the ratio of total demand to total capacity is presented in table 6.1.

The original instances CMT1-5 have between 50 and 199 randomly located customers, with a centralized depot. In instances CMT11-12, the customers are clustered,

Problem ID	Original Problem	Number of Customers		Trucks		Trailers		RDC
		VC	TC	Number	Capacity	Number	Capacity	
1		38	12					
2	CMT1	25	25	5	100	3	100	0.971
3		13	37					
4		57	18					
5	CMT2	38	37	9	100	5	100	0.974
6		19	56					
7		75	25					
8	CMT3	50	50	8	150	4	100	0.911
9		25	75					
10		113	37					
11	CMT4	75	75	12	150	6	100	0.931
12		38	112					
13		150	49					
14	CMT5	100	99	17	150	9	100	0.923
15		50	149					
16		90	30					
17	CMT11	60	60	7	150	4	100	0.948
18		30	90					
19		75	25					
20	CMT12	50	50	10	150	5	100	0.903
21		25	75					

Table 6.1: TTRP benchmark instances by Chao. The column RDC shows the ratio of demand to capacity. Ids 1-15 are unclustered instances while 16-21 are clustered.

and have a centralized depot in one instance and a decentralized depot in the other. The number of available trucks and trailers in the instances are to be interpreted as an upper bound, and therefore a routing plan which exceeds one of the limits is considered infeasible. All of the TTRP instances have a ratio of demand to total capacity above 90%.

6.2 Solomon Benchmarks (VRPTW)

The Solomon [Sol87] benchmarks are among the most common instances used to compare algorithms for the VRPTW. Like the CMT benchmarks mentioned previously, these benchmark instances can also be categorized by their structure. A total of 56 instances are grouped into 6 problem categories. A specific naming convention DTm is used to identify each instance, where D defines the structure of the customers; C, R, or RC. C identifies the instance as one where the customers are clustered. R indicates an instance with randomly located customers, and finally RC is used to indicate a mixture of randomly located and clustered customers. T is used to classify the scheduling horizon of the instance and can be either 1 or 2. If T is 1 the instance has a short scheduling horizon, whereas 2 means a long scheduling horizon. Lastly m is used to distinguish between instances of the same structure and type, but having variations in time windows. Versions of the instances exist with 25, 50, 75, and 100 customers, but in the context of this thesis we consider only those containing 100 customers.

When discussing the hardness of the problem at hand (see 3.4) we mentioned only a handful of Solomon instances remain unsolved to optimality, thus when considering the quality of solutions obtained from the algorithms implemented in this thesis, it makes good sense to use the lower bound as reference.

Solomon has compiled a list of best known solutions found by means of heuristics [Sol05]. Although it is not clear what the objective function of the heuristic methods is, it is assumed to equal that of (3.3.2), i.e. a minimization of employed vehicles and then total distance. Therefore comparisons to the best known heuristic values are only considered if the solutions obtained by the algorithms implemented in this thesis contain the same number of vehicles.

A compilation of the best known solutions obtained by means of heuristics and also optimal solutions can be found in table 8.2.

6.3 Generated Benchmarks (ETTRP)

In order to be able to evaluate the implemented algorithms on ETTRP instances, these would initially have to be constructed since we did not find any instances of this kind available in the literature. The construction of the instances is, however, very straight forward. We choose to use the same conversion algorithm used to convert VRP instances into TTRP instances (described in section 6.1) for converting the Solomon instances into ETTRP instances. In addition, a number of randomly placed parking places equalling the number of vehicle customers of the instance i.e. 25%, 50%, and 75% are added to the instances, having time windows equal to the depot. Let x_{max}, y_{max} and x_{min}, y_{min} denote the maximum and minimum x and y coordinates of all customers in the problem instance, then the coordinates of every parking place is chosen randomly from the intervals $x \in [x_{min}, \dots, x_{max}]$ and $y \in [y_{min}, \dots, y_{max}]$.

However, due to the sheer number of available VRPTW instances, only the first two instances of every category were chosen for conversion, resulting in the compilation which can be seen in table 6.2.

Instance Description and Analysis: As shown in table 6.2 the capacities of trucks and trailers are not fixed beforehand, the explanation for this follows: Let us consider the original VRPTW instance R101 (Values from [S.07]):

- Vehicle capacity 200
- Total demand of instance: 1458
- Optimal Solution: 1637.7 using 20 vehicles
- Total capacity available: 4000
- Demand to capacity ratio: 0.36

Even in the case where the number of employed vehicles is minimized as it is assumed was done for all heuristic solutions (see table 8.2) we see that 19 vehicles were used, resulting in a demand to capacity ratio of 0.38. Thus we may conclude that on average the vehicles are less than half full, hence it is unlikely that we may obtain a better solution to these instances if we increase the capacity of the trucks.

Problem ID	Original Problem	Number of Customers		Trucks		Trailers		RDC
		VC	TC	Number	Capacity	Number	Capacity	
R101_25		75	25					
R101_50	R101	50	50	∞	50-220	∞	50-220	-
R101_75		25	75					
R102_25		75	25					
R102_50	R102	50	50	∞	50-220	∞	50-220	-
R102_75		25	75					
C101_25		75	25					
C101_50	C101	50	50	∞	50-220	∞	50-220	-
C101_75		25	75					
C102_25		75	25					
C102_50	C102	50	50	∞	50-220	∞	50-220	-
C102_75		25	75					
RC101_25		75	25					
RC101_50	RC101	50	50	∞	50-220	∞	50-220	-
RC101_75		25	75					
RC102_25		75	25					
RC102_50	RC102	50	50	∞	50-220	∞	50-220	-
RC102_75		25	75					
R201_25		75	25					
R201_50	R201	50	50	∞	50-1200	∞	50-1200	-
R201_75		25	75					
R202_25		75	25					
R202_50	R202	50	50	∞	50-1200	∞	50-1200	-
R202_75		25	75					
C201_25		75	25					
C201_50	C201	50	50	∞	50-900	∞	50-900	-
C201_75		25	75					
C202_25		75	25					
C202_50	C202	50	50	∞	50-900	∞	50-900	-
C202_75		25	75					
RC201_25		75	25					
RC201_50	RC201	50	50	∞	50-1200	∞	50-1200	-
RC201_75		25	75					
RC202_25		75	25					
RC202_50	RC202	50	50	∞	50-1200	∞	50-1200	-
RC202_75		25	75					

Table 6.2: Generated ETTRP benchmarks. An infinite amount of trucks and trailers is assumed. Several runs with various capacities of truck and trailers within the specified limits were conducted.

In order to determine the effects of using trailers on these instances, a study of the effects of truck capacity on the value of the objective function was conducted and may be seen in section 8.3.1.

Chapter 7

Solving the ETTRP

This chapter is devoted to a discussion of the main techniques and algorithms used for solving the ETTRP. In the first section we discuss how to take advantage of different forms of technology in order to make the implementation more efficient. From there we proceed to describe how to construct initial feasible solutions to the problem using two well known algorithms in the process.

We introduce the various neighbourhood operators as well as describe the search strategy used to improve the initial solutions.

Finally we describe the Attribute Based Hill Climber (ABHC) as the main algorithm used to guide the search out of local optima, and the motivation for the choice of this particular metaheuristic.

Lastly we discuss various steps taken in order to change the default behaviour of the ABHC in the hopes it might have a beneficial impact on both solution quality and time needed to find good solutions.

7.1 Design and Technology

When designing the algorithms used for solving the ETTRP, it was done keeping the following points in mind:

- The hardware on which the algorithms would be deployed.
- The programming language used to implement the algorithms.
- Portability (minor concern).

Considering the fact that the main trend today still tends to add additional cores to CPUs in order to increase the processing power, it was decided early on to somehow utilize this extra processing power if available.

Since both C and C++ are commonly used for solving optimization problems, it was decided early on, to implement the algorithms in the C99 dialect of the C programming language, resulting in some 15,000 lines of C code (unfinished). However, obstacles in terms of implementation specific details became frequent, and it was decided to re-implement everything using the C++ programming language resulting in approximately 14,000 lines of C++ code, using the Qt 4.6.1 libraries developed by Nokia[1] to ease the implementation. The project was compiled using the GNU g++ compiler version 4.4.1 (Ubuntu 4.4.1-4ubuntu9).

All algorithms were tested on machines with the following configuration:

CPU Intel(R) Core(TM)2 CPU 6300 @ 1.86GHz

Main Memory 2059448 kB

OS Ubuntu Linux 9.10

KERNEL 2.6.31-19-generic #56-Ubuntu SMP Thu Jan 28 01:26:53 UTC 2010 i686 GNU/Linux

To utilize the hardware efficiently, a threaded approach was implemented. At the core of the overall program a thread pool, containing as many threads as there are processing cores is responsible for executing queued tasks.

When considering the overall structure of a program to solve discrete optimizations problems using heuristics, we may end up with a generalization consisting of phases similar to the following:

Phase 1: Construct one or more starting solutions.

Phase 2: Improve starting solutions by means of stochastic local search or similar.

Phase 3: Guide the search out of local optima by means of metaheuristics, while remembering the best seen solution so far.

Phase 4: Output the best seen solution.

Each of the phases, except the last described above, is considered a task in the context of this thesis, and as such can be scheduled for execution by the main thread pool. Individual tasks may be run an arbitrary number of times since they are designed as independent entities, and therefore none of the threads in question interact with each other. Thus in general phase 3 is run several times consecutively until some predefined stopping criterion is met.

As an example we may choose to create 10 tasks responsible for creating 10 initial solutions to the ETTRP, after which, we create 10 new local search tasks working on the initial 10 starting solutions, and so forth. Thus to summarize: As soon and as long there is work to be done, the thread pool will do it utilizing as many processing cores as possible.

A static object is used to keep track of the best seen solution and is thus the only object accessed by many threads concurrently. The contention is low, however, since access is only needed when a new global best solution is found which happens "rarely".

We now proceed to describe the heuristics used for constructing the initial solution to the problem.

[1] Qt Libraries, http://qt.nokia.com/

7.2 Construction Heuristics

Since traditional VRP problems do not include the concept of subtours, a new strategy must be devised taking these into account, or rather, to consider the fact that a subset of customers cannot be serviced by a complete vehicle. In the following the overall strategy is presented after which two well known construction heuristics implemented will be discussed.

7.2.1 Strategy

The approach chosen in this thesis for dealing with two distinct customer types when designing a construction algorithm can be summarized as follows:

1. First cluster the truck customers according to a predefined criterion.

2. Every cluster now denotes a VRP subproblem consisting solely of truck customers.

3. Solve each VRP subproblem (cluster) as a regular VRPTW problem, resulting in a number of routes or segments (see 5.3).

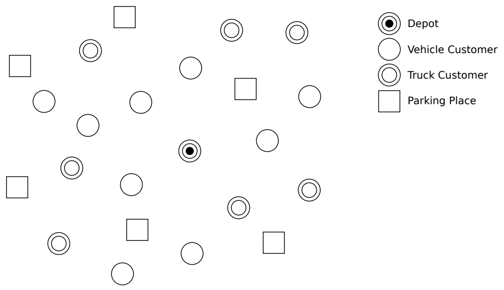
4. Include every segment as a single customer in the original problem which now consists of only vehicle customers and segments, both of which may be serviced by complete vehicles.

5. Solve the complete problem as a regular VRPTW.

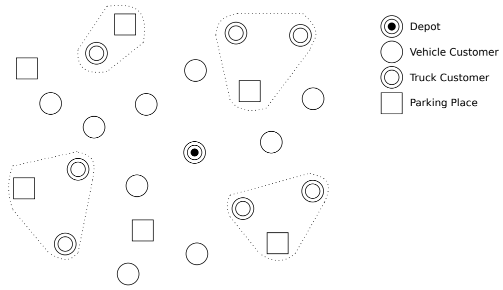
The entire concept of the procedure is shown in figure 7.1. Figure 7.1(a) represents the input problem consisting of a mixture of both truck and vehicle customers. In figure 7.1(b) the customers have been clustered around their nearest parking places. Figure 7.1(c) shows how the individual subproblems are solved as regular VRP's. Finally figure 7.1(d) shows how the main problem is solved for vehicle customers and the segments representing subtours.

In order for the above strategy to be successful we have to consider the following:

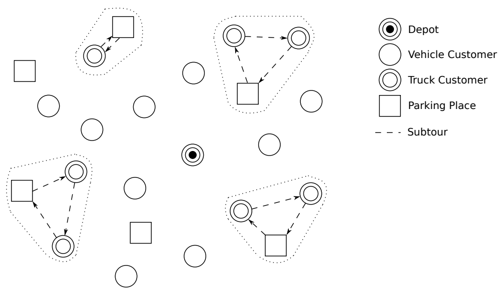
Clustering Criterion: It is clear that the choice of clustering criterion may have a huge impact on the quality of the solutions generated by the above strategy. In addition, it is obvious that all truck customers should be clustered around some parking place, which acts as a depot in the subproblems. Initially, it was decided to partition the truck customers according to their closest parking place in terms of euclidean distance. Obviously, this choice is not necessarily ideal when time windows are considered, since a clustered group of customers may be close in distance but far apart in a temporal sense. In 8.3.4 we consider how this may be improved.



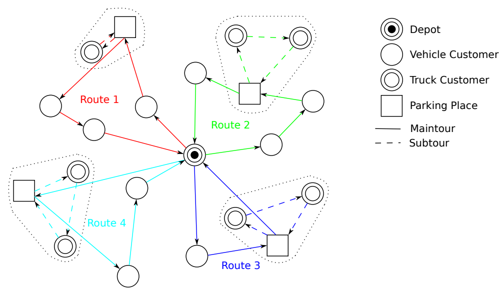
(a) Problem input consisting of both truck and vehicle customers.



(b) Clustering of truck customers around nearest parking places.



(c) Solving subproblems corresponding to individual Vehicle Routing Problems.



(d) Solving the main problem as a VRP including subproblems modeled as vehicle customers.

Figure 7.1: Illustration of how the procedure to create feasible starting solutions to the ETTRP works.

Subtour Demand: Since the subproblems essentially symbolize subtours in the main problem, it is clear that the capacity limit of the subproblems are bounded by the capacity of a truck.

Algorithm 9 used for clustering is self explanatory but included for completeness.

After applying Algorithm 9 we obtain clusters containing parking places grouped with a number of truck customers. As mentioned previously each parking place is to be interpreted as a fictitious depot in the subproblems that should be solved as regular VRPTW problems allowing routes to reach a maximum capacity equalling that of a truck in the main problem.

Next, the sequential tour-building heuristics I1 and I3 by Solomon [Sol87] primarily used for generating starting solutions for the VRPTW, is described.

Algorithm 9: cluster

```
Data: All truck customers c ∈ V_T
Result: A clustering of all truck customers c ∈ V_T around their nearest
        parking places
clusters ← ∅;
foreach c ∈ V_T do
    parkingPlace ← getNearestParkingPlace(c);
    if parkingPlace ∉ clusters then
        clusters[parkingPlace] ← c;
    else
        clusters[parkingPlace] ← clusters[parkingPlace] ∪ c;
```

7.2.2 Solomon I1

Solomon [Sol87] created several tour-building heuristics useful for obtaining initial solutions for the VRPTW.

The insertion heuristic commonly referred to as I1 considers two criteria κ_1 and κ_2 when determining which customer to insert next in the emerging route. We refer to κ_1 as the *insertion criterion*, and to κ_2 as the *selection criterion* both of which are defined as follows:

$$\kappa_1(i,u,j) = \alpha_1 \cdot \kappa_{11}(i,u,j) + \alpha_2 \cdot \kappa_{12}(i,u,j) \tag{7.2.1}$$

$$\kappa_2(i,u,j) = \lambda \cdot c_{0u} - \kappa_1(i,u,j) \tag{7.2.2}$$

$$\kappa_{11}(i,u,j) = c_{iu} + c_{uj} - \mu \cdot c_{ij} \tag{7.2.3}$$

$$\kappa_{12}(i,u,j) = b_j^{new} - b_j \tag{7.2.4}$$

Where $\alpha_1 + \alpha_2 = 1$ and $\alpha_1 \geq 0, \alpha_2 \geq 0, \mu \geq 0$ and $\lambda \geq 0$. b_j^{new} is the new time for service to begin at customer j, given that u is inserted.

Equation (7.2.1) is used to determine the best feasible insertion of customer u in the emerging route between customers i and j, whereas equation (7.2.2) calculates the savings of servicing customer u on the same route as customers i and j. (7.2.3) calculates the cost of inserting customer u between customers i and j. Equation (7.2.4) calculates the generated *push forward* at j when inserting customer u.

When constructing the routes, Solomon suggests initializing a new route using a seed customer which is chosen to be either:

(a) The farthest unrouted customer.

(b) The unrouted customer with the earliest deadline.

Algorithm 10 determines the cheapest edge on which a customer c should be inserted using the insertion criterion (7.2.1), otherwise the implementation is straight forward except for line 5 where we check the feasibility of inserting customer c on edge e using Algorithm 2.

Algorithm 10: cheapestPair

Data: Customer c

Result: A Customer-Edge pair $\langle c, e \rangle$ denoting the cheapest edge e on which to insert c

resultPair $\leftarrow$ *null*;
minCost $\leftarrow \infty$;
foreach *Edge e in the emerging route* **do**
    cost $\leftarrow \kappa_1(\alpha(e), c, \beta(e))$; // (7.2.1)
    if cost $<$ minCost $\wedge$ FEASIBLEINSERTION(c, e) **then**
        minCost $\leftarrow$ cost;
        resultPair $\leftarrow \langle c, e \rangle$;
return resultPair;

Algorithm 11 is used to determine which pair of customer-edge relations yields the biggest savings using the selection criterion (7.2.2).

Algorithm 11: maxSavingsPair

Data: List L of Customer-Edge pairs $\langle c, e \rangle$

Result: Customer-Edge pair resulting in the maximum savings

resultPair $\leftarrow$ *null*;
maxSavings $\leftarrow -\infty$;
foreach *Customer-Edge pair* $\langle c, e \rangle \in L$ **do**
    savings $\leftarrow \kappa_2(\alpha(e), c, \beta(e))$; // (7.2.2)
    if savings $>$ maxSavings **then**
        maxSavings $\leftarrow$ savings;
        resultPair $\leftarrow \langle c, e \rangle$;
return resultPair;

The entire insertion algorithm (Algorithm 12) can be described as follows: Observe that complete vehicle routes are constructed before any truck routes (line 3 and 15). In line 6 we make sure to loop as long as there are unrouted customers remaining. In line 8 we compute the best insertion edges for each unrouted customer. In line 12 we check two things: if the list containing customer-edge pairs is empty and whether there still remain unrouted customers. If this is the case it means no more of the unrouted customers may feasibly be inserted and we initialize a new route. If the check in line 18 is satisfied, we have several possible candidates for insertion in which case we insert the one with the cheapest insertion cost using Algorithm 7.

Algorithm 12: insertionHeuristic

Data: Set of unrouted customers U
Result: A routing plan R

choose seed $s \in U$ according to (a) or (b); `// See page 45`
initialize new route r using seed s;
assign next available complete vehicle to r or truck otherwise;
$R \leftarrow \{r\}$;
$U \leftarrow U \setminus \{s\}$;
while $|U| > 0$ **do**
  customerEdgeList $\leftarrow \emptyset$;
  foreach *customer* $c \in U$ **do**
    $\langle c, e\rangle \leftarrow$ CHEAPESTEDGEPAIR(c);
    if $\langle c, e\rangle \neq$ *null* **then**
      customerEdgeList $\leftarrow$ customerEdgeList $\cup \langle c, e\rangle$;
  if $|$customerEdgeList$| = 0 \wedge |U| > 0$ **then**
    choose seed $s \in U$ accordingly;
    initialize new route r using seed s;
    assign next available complete vehicle to r or truck otherwise;
    $R \leftarrow R \cup \{r\}$;
    $U \leftarrow U \setminus \{s\}$;
  if $|$customerEdgeList$| > 0$ **then**
    $\langle c, e\rangle \leftarrow$ MAXSAVINGSPAIR(customerEdgeList);
    INSERT(c, e);
    $U \leftarrow U \setminus \{c\}$;

Improving I1

Dullaert and Bräysy [DB03] describe how the insertion heuristic I1 may be improved in case only few customers exist in each route. According to Dullaert and Bräysy the insertion cost can be underestimated in case an additional customer is inserted between the depot and the first customer of the route, because the generated push backward is not taken into account. Inserting a customer u between the depot i and the first customer j advances the depot departure time. This is also the case even if there is no waiting at customer j before inserting u. If j would remain the first customer until the route is finalized, the vehicle would leave the depot at time $b_j - c_{ij}$. If customer u is inserted between i and j and remains the first customer until the route is finalized, the vehicle would leave the depot at time $b_u - c_{iu}$. As a result, inserting customer u between i and j advances the depot departure time by $(b_j - c_{ij}) - (b_u - c_{iu})$. This is defined as the push backward generated by inserting customer u between i and j.

Dullaert and Bräysy argue that the time insertion of inserting customer u between the depot i and the first customer j is equal to the push backward in addition to the generated push forward. Thus in order to take this into account equation (7.2.4) is changed to the following:

$$\kappa_{12}(i,u,j) = \begin{cases} [(b_j - c_{ij}) - (b_u - c_{iu})] + (b_j^{new} - b_j) & \text{if } i = 0_s \\ b_j^{new} - b_j & \text{if } i \neq 0_s \end{cases} \tag{7.2.5}$$

All other aspects of Algorithm 12 remain the same.

It will be investigated if improved solutions may be obtained by including the Push Backward Maximum Push Forward (PBmaxPF) introduced by Dullaert and Bräysy.

7.2.3 Solomon I3

Another sequential tour-building heuristic by Solomon commonly referred to as I3 was implemented. This heuristic tries to account for the urgency of servicing a customer u when deciding which customer to service. The insertion criterion κ_1 and the selection criterion κ_2 are defined as follows:

$$\kappa_1(i,u,j) = \alpha_1 \cdot \kappa_{11}(i,u,j) + \alpha_2 \cdot \kappa_{12}(i,u,j) + \alpha_3 \cdot \kappa_{13}(i,u,j) \tag{7.2.6}$$

$$\kappa_2(i,u,j) = \kappa_1(i,u,j) \tag{7.2.7}$$

$$\kappa_{11}(i,u,j) = c_{iu} + c_{uj} - \mu \cdot c_{ij} \tag{7.2.8}$$

$$\kappa_{12}(i,u,j) = b_j^{new} - b_j \tag{7.2.9}$$

$$\kappa_{13}(i,u,j) = l_u - b_u \tag{7.2.10}$$

Where $\alpha_1 + \alpha_2 + \alpha_3 = 1$, $\alpha_1 \geq 0$, $\alpha_2 \geq 0$, $\alpha_3 \geq 0$, $\mu \geq 0$, and $\lambda \geq 0$. Here the last term κ_{13} accounts for the urgency of servicing customer u as can be seen from (7.2.10).

It should be clear that Algorithms 10 and 11 can easily be adapted to use the equations (7.2.6)-(7.2.10) instead, thus in the implementation we may easily switch between the I1 and I3 construction heuristics.

7.2.4 Tunable Parameters

Besides the parameters mentioned previously in regards to the Solomon insertion heuristics I1 and I3, additional parameters are introduced in order to tune the behaviour of the construction heuristics. Please note that all parameters discussed in this section refer only to the construction heuristics.

Seed Customers: The initial choice of seed customer when each route is initialized potentially has a big impact on the quality of the solution generated. Empirical tests using the following choices of initial seed customers were conducted.

1. Farthest customer from depot.
2. Customer with earliest deadline.

3. Customer with lowest service duration time (Segments in general have higher service time duration).
4. Nearest customer from depot.
5. As (2) but picking the median instead.
6. Highest demand (Segments in general have higher demands).
7. Randomly chosen seed.

Limiting Subtour Demand: Furthermore, it may be the case that setting the upper bound on subtour capacity to that of a truck makes route construction inflexible, and therefore we introduce a variable $\nu \in [0 \ldots 1]$ as a factor used to limit the allowed capacity of subtours during the initial route construction. Thus setting $\nu = 0.5$ limits subtour capacity to 50% of C_{truck}. Please note that this restriction is only used by the construction heuristic, during the local search phase it is disregarded.

Push Back: The push back discussed in 7.2.2 is configurable and may be either enabled or disabled during route construction to see if it may have a beneficial impact on the solutions generated.

Randomized Parking Places: Instead of assigning every truck customer to its closest parking place during the clustering step, we instead assign every truck customer to a random parking place. This may either be enabled or disabled during route construction, and is included for completeness in order to determine the necessity of using "good" parking places during route construction in order to obtain high quality solutions.

7.2.5 Creating the Initial Solution

All the aspects needed for creating the initial solution to the ETTRP have been covered. For completeness, the process is reiterated by algorithm 13.

Algorithm 13: createRoutingPlan

Data: Two sets of unrouted customers V_T (truck customers) and V_V (vehicle customers)
Result: A routing plan R

$R \leftarrow \emptyset$;
clusters $\leftarrow$ CLUSTER(V_T);
foreach $\rho \in$ clusters **do**
  $V_V \leftarrow V_V \cup$ INSERTIONHEURISTIC(ρ);
$R \leftarrow$ INSERTIONHEURISTIC(V_V);
return R;

In line 4 we use the fact that we may view a segment returned by algorithm 12 as a single customer and hence may include it as a vehicle customer in the set V_V. This stems from the fact that the segments returned by the heuristic are in fact subtours and therefore may be serviced by a complete vehicle. Also recall that we have extended the temporal definitions to include segments and hence as far as the insertion heuristic is concerned every segment is simply treated as a vehicle customer.

Next, the different neighbourhood operators implemented in this thesis will be discussed.

7.3 Neighbourhoods

Since the VRP is a well studied combinatorial optimization problem, many different algorithms both exact and heuristics have been created the last decades in order to solve the problem. In the context of heuristics the same can be said for the neighbourhoods which are explored in order to improve a solution. However, since the problems treated in this thesis can be seen as generalizations of the VRPTW we may restrict the choice of neighbourhood operators such that certain undesirable effects do not occur. The well known `2-opt` operator shown in figure 7.2 shows how the path from $i+1$ to j has been reversed. Thus customer $i+1$, which was serviced early in the route before the move, is now serviced later than j. Although this path reversal may be feasible with respect to the customers time windows in some instances, one might assume that in general this is not the case. This is backed by a survey of Bräysy and Gendreau [BG05, p.12] stating that only 10% of all solution improvements involved path reversals.

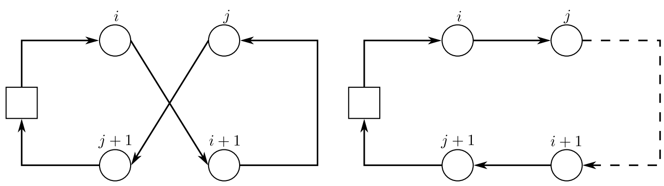


Figure 7.2: Example of the 2-Opt operator applied to a route. The edges $(i, i+1)$ and $(j, j+1)$ are replaced by edges (i, j) and $(i+1, j+1)$, reversing the service ordering of customers between $i+1$ and j.

Many neighbourhoods were considered to be implemented in this thesis ranging from classic node exchange neighbourhoods such as relocate and exchange to the more exotic variants of ejection chains such as those proposed by by Rego [Reg98] and [Reg01] as well as Sontrop, Van Der Horn and Uetz [SVDHU05]. Various edge exchange neighbourhoods such as `2-Opt`, `3-Opt`, and `Or-Opt` were also considered. The `Or-Opt` is a special case of the `3-Opt` where no path is reversed as a result of applying the operator.

In terms of efficiency the articles by Irnich, Funke, and Grünert [IFG06, FGI05] and Irnich [Irn08a] were considered also, since the concept of sequential search compared to lexicographic seems promising, however, from an implementation point of view it was decided to go in a more simplistic direction to avoid neighbourhoods which involved path reversals.

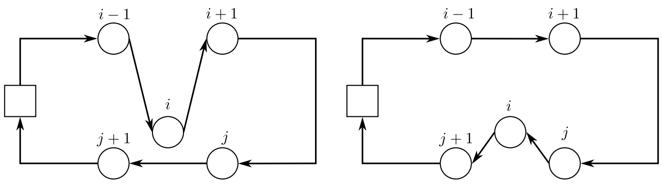
In a survey of Bräysy and Gendreau [BG05] a summary of various construction heuristics as well as neighbourhoods was presented. From these; `relocate` and `exchange` were chosen to be implemented in this thesis. Furthermore motivated by the fact that both Scheuerer [Sch04] and Chao [Cha02] claim that solution quality for the TTRP depends on the possibility of a subtour refinements, such an operator was implemented as well (`subtour relocation`). The latter, however, in a more restricted version when compared to the versions of Chao and Scheuerer.

The following sections describe the various neighbourhood operators implemented in this thesis.

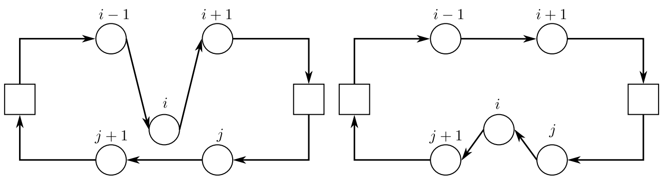
7.3.1 Relocate Neighbourhood

The principle of the `relocate` neighbourhood operator is the removal of a customer from a route and its insertion elsewhere, in either the same route or another. The two situations are depicted in figure 7.3. The operator removes edges $(i-1, i)$, $(i, i+1)$, and $(j, j+1)$, followed by insertion of edges $(i-1, i+1)$, (j, i), and $(i, j+1)$.

It is clear, that the removal of a customer i is always feasible in a temporal context, however, the insertion of customer i may cause a temporal shift in the route in which i is inserted. Thus as is the case for any neighbourhood considered for the VRPTW we need to check and guarantee feasibility before i may be inserted. One advantage of the `relocate` neighbourhood is that the temporal changes does not affect an entire path as was the case of the `2-Opt` operator (which has to be checked for feasibility).



(a) The relocate operator applied to customer i in a single route. The square symbolizes the depot.



(b) The relocate operator applied to customer i. Here i is moved from one route to another. The squares represent the same depot.

Figure 7.3: Two examples of applying the relocate neighbourhood operator

Motivation: The motivating factors of implementing the `relocate` neighbourhood may quickly be summarized as follows:

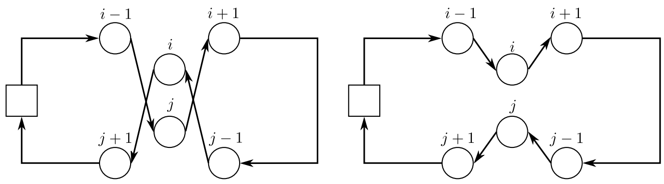
1. The implementation is straight forward.
2. The operator allows the overall number of routes to be reduced.
3. The feasibility checks needed may be computed in constant time $\boldsymbol{O}(1)$.
4. The size of the neighbourhood is only $\boldsymbol{O}\left(n^2\right)$.

Relocate Neighbourhood Restrictions: The following restrictions are imposed in order to preserve feasibility:

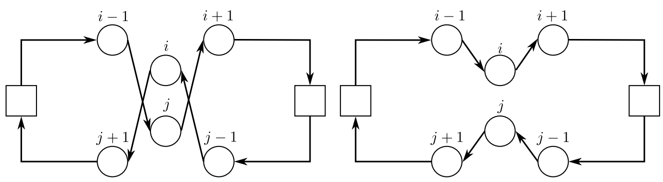
- The depot may not be relocated.
- Parking places may not be relocated.
- A truck customer may only be inserted on a subtour edge.
- Removing the last customer of a subtour deletes the parking places denoting the subtour (as required by rule 5 of section 5.1).
- Removing the last customer of a tour may cause the tour to be deleted[2].

7.3.2 Exchange Neighbourhood

The `exchange` neighbourhood considers two customers i and j which may be located within the same or different routes. The operator then tries to exchange the two customers as depicted in figure 7.4.



(a) The exchange operator applied to customers i and j of a single route. The square symbolizes the depot.



(b) The exchange operator applied to customers i and j of different routes. The squares represent the same depot.

Figure 7.4: Two examples of applying the exchange neighbourhood operator

As can be seen the exchange is done by replacing the edges $(i-1, j)$, $(j, i+1)$, $(j-1, i)$, and $(i, j+1)$ by the edges $(i-1, i)$, $(i, i+1)$, $(j-1, j)$, and $(j, j+1)$.

[2] This will be covered in section 7.4.2

The usefulness of the operator is arguable since once one customer i is fixed, the choice of the other customer j has to be compatible with i in terms of time windows. Furthermore, it is clear that the number of customers within any route remains unchanged by this operator.

Motivation: The motivating factors for including the `exchange` operator is due to the load constrained nature of the TTRP instances. Every single TTRP instance considered in this thesis has a load to demand ratio above 90%. Thus the `exchange` operator might indeed be able to allow the search to reach solutions that would otherwise be impossible to reach using only the `relocate` operator. As before, the characteristics of the `exchange` neighbourhood are summarized:

1. The implementation is straight forward.
2. Solutions which may not otherwise be reachable could be visited.
3. Feasibility checks may be performed in constant time $\boldsymbol{O}(1)$.
4. The size of the neighbourhood is only $\boldsymbol{O}\left(n^2\right)$.

Exchange Neighbourhood Restrictions: As was the case of the `relocate` neighbourhood operator the following restrictions were imposed in order to preserve feasibility:

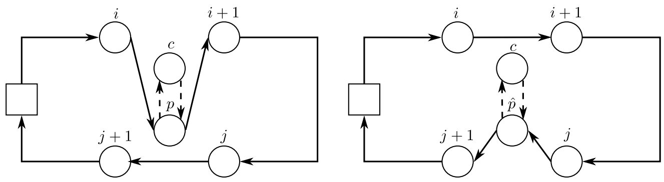
- Neither the depot nor a parking place may be a candidate for an exchange.
- An exchange is only considered feasible if both customers i and j are vehicle customers **or** they are compatible with respect to the way they are serviced. Thus if i is serviced on a subtour, then j must also be serviced on a subtour and vice versa.

The last restriction in the list above serves to ensure that no truck customer i may be considered a candidate in an exchange with a vehicle customer j that is serviced on a maintour.

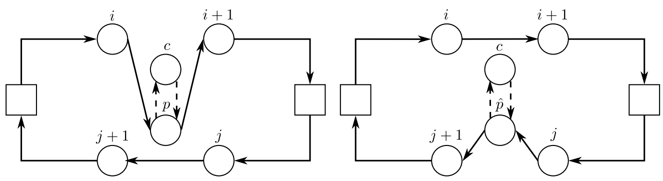
7.3.3 Subtour Relocation Neighbourhood

As routes change dynamically by applying the `relocate` and `exchange` neighbourhood operators, it becomes increasingly important to not restrict the initial location of subtours. Thus an additional neighbourhood operator called `subtour relocation` is introduced, which enables the creation of a new subtour by relocating a truck customer, as can be seen in figure 7.5.

The figure shows a simplified case in which the subtour contains only a single customer c. Because of this the entire subtour rooted at p may be removed as can be seen in the right figure of both 7.5(a) and 7.5(b). When a parking place $\hat{p}$ is to be chosen for customer c, it is done in such a way, that the insertion cost is minimized,



(a) The subtour relocation operator applied to customer c. Customer c is relocated within the same route, using $\hat{p}$ as parking place. Thus in the figure p and $\hat{p}$ are to be interpreted as different although shown at the same location in the figure.



(b) The subtour relocation operator applied to customer c. Customer c is relocated to a new route, using $\hat{p}$ as parking place. Thus in the figure p and $\hat{p}$ are to be interpreted as different although shown at the same location in the figure.

Figure 7.5: Two examples of applying the subtour relocation neighbourhood operator

i.e. for every possible parking place determine the cost of inserting c into $\hat{p}$ to which the cost of inserting $\hat{p}$ into the route is added.

The name `subtour relocation` may be slightly misleading since only a single customer is relocated and not an entire subtour. However, as is often the case in problems including time windows, this restriction may allow for a greater exploration of the search space, and by applying the `subtour relocation` operator followed by a number of regular `relocate` operations, it is possible to reach the same solutions as the subtour refinements of Chao and Scheuerer would allow.

Motivation: The inclusion of this neighbourhood operator is motivated by the following list:

1. Dynamic subtours in the sense that we may create new and delete existing subtours. Note, however, that deletion may also be achieved by applying the `relocate` operator, since relocating the last customer of a subtour will result in the removal of the subtour.

2. Feasibility checks may be performed in constant time $\boldsymbol{O}(1)$.

3. The size of the neighbourhood is $\boldsymbol{O}\left(n^3\right)$.

4. The implementation is straight forward.

Subtour Relocation Neighbourhood Restrictions: The neighbourhood is restricted in the following way to preserve feasibility:

- Only truck customers may be considered for relocation and thus for subtour creation.
- The destination tour has to be serviced by a complete vehicle.
- Removing the last customer of a tour may cause the tour to be removed[3].

In order to speed up the search for the cheapest parking place to use for the relocated truck customer, we cache which parking place to use depending on the edge where the truck customer is to be inserted. Thus for consecutive runs we are able to check the cost of insertion in constant time. In case the result was not previously cached it has to be calculated costing $\boldsymbol{O}\left(n^3\right)$.

All implemented neighbourhoods have been described and we now proceed to describe the metaheuristic implemented in this thesis.

7.4 The Attribute Based Hill Climber (ABHC)

The Attribute Based Hill Climber (ABHC) is a relatively new algorithm first introduced by Whittley and Smith [WS04] in 2004. In the article it was applied to the Quadratic Assignment Problem and the Travelling Salesman Problem. For both problems it was shown to be competitive with existing algorithms. Derigs and Kaiser applied the algorithm to the VRP [DK07] in 2006, where it proved to be competitive with the best known heuristics for the VRP. In his masters thesis Nikolajsen [Nik09] implemented the ABHC and applied it to the Site Dependent VRP with time windows (SDVRPTW) yielding promising results.

7.4.1 The ABHC Algorithm

The Attribute Based Hill Climber is a parameter free algorithm which is based on the principles of the Tabu Search algorithm by Glover [G$^+$89].

The main component of the algorithm is a set of atomic attributes, for which the value of the objective function of the best solution they were part of, is associated. The algorithm searches through all neighbouring solutions picking the one with the lowest value of the objective function. A move is accepted if the new objective value improves at least one attribute. This continues until no more moves are acceptable, in which case the best found solution during the search is returned.

Initially all attributes are initialized to the worst possible value, which in minimization problems is ∞, except for attributes included in the starting solution which have their value set to the value of the objective function.

Derigs and Kaiser [DK07] conducted a study in order to determine the best choice of attribute set for the VRP. It was concluded that the set of edges constituted the best choice. In the context of this thesis, however, it was decided to deviate from this choice because it may lead to ambiguity in the attribute set, since the same parking place may be included in several routes. The choice for attributes in this thesis is a tuple $\langle tour\_id, edge \rangle$.

[3] This will be covered in section 7.4.2

Algorithm 14: Standard ABHC

Data: A routing plan R

Result: A routing plan R_r

1 $R_{best} \leftarrow R$;
2 $R_{current} \leftarrow R$;
3 **while** $R_{current} \neq null$ **do**
4   IDENTIFYWORSTATTRIBUTES(R);
5     $R_{current} \leftarrow null$; `// Set attributes of` $f(R_{current})$ `to` ∞
6   **foreach** $R_{\mathcal{N}} \in \mathcal{N}(R)$ **do**
7   **if** $f(R_{\mathcal{N}}) < f(R_{current}) \wedge$ ACCEPT(R,R_n) **then**
8             $R_{current} \leftarrow R_{\mathcal{N}}$;
9   **if** $R_{current} \neq null$ **then**
10         $R \leftarrow R_{current}$;
11   UPDATE($R_{current}$);
12   **if** $R_{current} < R_{best}$ **then**
13             $R_{best} \leftarrow R_{current}$;
14 **return** R_{best};

Algorithm 14 describes the ABHC, which will be discussed more closely in the following.

To avoid looking through all attributes, to see if a move improves on one of them, a sorted list of worst attributes of the current solution using the IDENTIFYWORSTATTRIBUTES function is maintained.

The ACCEPT function determines if the new objective value improves on at least one of the attributes which is still in the solution after the move, and if it does, the move is accepted. At every iteration the entire neighbourhood $\mathcal{N}$ (line 6) defined by the neighbourhood operators described in previous sections is searched exhaustively for acceptable moves always keeping track of the best feasible move. Once the entire neighbourhood has been evaluated, the best move is made (line 10) and the attributes comprising this new solution are updated using the UPDATE function.

The ABHC overcomes the problems of escaping local minima by allowing moves that worsen the solution as long as at least one of the attributes improves. This feature allows ABHC to explore the search landscape very extensively, while still favouring better solutions due to its *Best Improve* nature. Every single move in the search space requires at least one attribute to measurably improve, which means that the algorithm is bound to terminate at some point, but most notably, that the algorithm will never visit the same solution twice. In short the behaviour of the algorithm is that of a *Steepest descent - Mildest ascend* algorithm.

The ABHC was chosen since it has been shown in the literature to be competitive with some of the best metaheuristics for the VRP. It is almost parameter free, meaning that only the choice of attribute set has to be determined in advance. Last but not least, it is relatively easy to implement.

Algorithms like Simulated Annealing, Tabu Search etc., all require substantial tuning of parameters depending on the specific problem or instance, which speaks favourably for using ABHC. Unfortunately, the very strength of the ABHC algorithm is also its main drawback. Since it cannot be tuned in the normal sense, its running time cannot be directly influenced.

Both Scheuerer and Chao implemented variants of the Tabu Search algorithm to solve the TTRP. Yu et al. based their approach on Simulated Annealing, and therefore this thesis is as much a study of how well the ABHC performs on TTRP related problems.

The Best Improve search strategy (steepest descent) of the ABHC and the strict improvement criterion imposed on attributes make the overall behaviour of the ABHC deterministic. Thus, two consecutive runs of the algorithm on the same input will always produce the same solution.

7.4.2 Algorithm Strategy

In this section the overall search strategy of the ABHC is covered. Specifically the restart behaviour and the possibility of tour deletion during the search will be discussed.

Restart Behaviour:

Derigs and Kaiser [DK07] point out that another advantage of the ABHC is its ability to start from a local optimum and escape it successfully. Obviously, consecutive runs of the algorithm, and subsequent restarts from the same local optimum still makes the overall behaviour deterministic. The ABHC algorithm implemented in this thesis is in general always restarted from a previously obtained local optimum.

Tour Deletion:

Intuitively, it may be desirable to delete tours during the search to decrease the overall route length and thus the value of the objective function. However, it is not uncommon that problem instances with time windows may in fact utilize more vehicles and still have smaller overall tour length. For instance the best known solution to the Solomon instance $R102$, when the number of vehicles is minimized and then route length, has an objective value of 1486.12 using 17 vehicles. The optimal solution to the problem, considering only tour length as its main objective, has a value of 1466.6 using 18 vehicles. Hence reducing the number of utilized vehicles is not always synonymous with minimizing route length.

Since we cannot determine in advance if the deletion of a tour could result in a better solution, it is proposed to fork the search into another task, which is queued in the thread pool at the point where a move would result in the deletion of a route. Thus if ABHC is started on a solution containing 24 routes, and the entire search of the algorithm results in 3 moves which may delete a tour, then 3 separate searches each being a separate snapshot of how the solution looked at the particular moment, the move was chosen. Each of the forked searches may themselves fork searches, whenever a move would have deleted a tour from their own respective routing plan. The purpose of this is, naturally, to enable the possibility of exploring various other areas in the search space. An illustration of the concept is shown in figure 7.6.

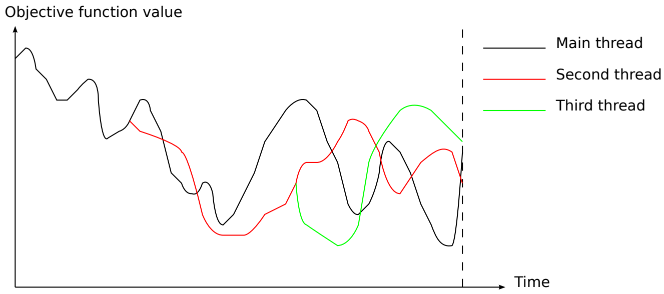


Figure 7.6: Illustration of a fictitious search progression. At the start of the search only the main thread exists, however, during the search a move might reduce the number of routes, thus starting a concurrent search as shown by the red line symbolizing the second thread. During the search of the second thread, another move might lead to yet another reduction in the number of routes, thus spawning another concurrent search illustrated by the green line symbolizing the third thread.

7.4.3 Adding Stochastic Behaviour

Among others, Hoos and Stützle [HS05, p. 37] claim that many high-performing local search algorithms make use of randomized choices when selecting neighbouring solutions. We explore the possibility of modifying the deterministic nature of the ABHC by adding stochastic behaviour whenever the choice of the best move should be determined. This is done by storing all *acceptable* moves of a neighbourhood operator $\mathcal{N}^{op}$ sorted by objective value in non-decreasing order.

$$\mathcal{N}^{op} = [m_i, m_{i+1}, \ldots],\ f(m_i) < f(m_{i+1})\ \forall i \in [0, \ldots, (|\mathcal{N}| - 1)]$$

Hence instead of always choosing a move $m = m_0$ it is instead proposed to choose m according to the following criterion:

$$m = m_i \text{ where } i = random(0,1)^r \cdot |\mathcal{N}|,\ r \in_R [2,3,4] \tag{7.4.1}$$

This way we favour best moves, but may occasionally pick a “worse” move essentially changing the default Best Improve behaviour of the ABHC. When starting the algorithm we specify a starting seed s for the random number generator in order to be able to reproduce results.

Recall that, as restart strategy (see 7.4.2) the ABHC is in general restarted from a previously obtained local minimum. When the algorithm terminates its i^{th} run using a random seed s we restart it again using a new seed $s + 1$, hereby (with high probability) preventing two consecutive runs of the algorithm to choose the same moves, and thus increase the possibility of escaping the local optimum, where the algorithm might otherwise have been stuck.

A possible drawback of the added stochastic behaviour might be the algorithms slower convergence towards a local optimum which may effectively increase the overall running time.

7.4.4 Speedup Methods

A drawback of the ABHC is its speed. Both Nikolajsen [Nik09] and Markus [GM09] explored various techniques in order to speed up the running time of the algorithm. In this section we also explore various approaches to speed up the ABHC algorithm.

A preliminary experiment was conducted to see if a deterministic behaviour in terms of non-improving iterations could be established. If for example all improving solutions were found during the first say 50% of the overall running time, then a premature termination of the search would speed up the algorithm substantially. Unfortunately, improvements in solution quality were still frequent just prior to the normal termination of the search, and therefore no pattern could be established (see for example figure 8.2).

In the following, various other measures which might potentially be able to speed up the algorithm will be investigated.

Start From Local Optimum

Derigs and Kaiser [DK07] report that the solution quality obtained by ABHC does not appear to depend on the quality of the starting solution. The algorithm does, however, appear to terminate faster in general, when started from a better initial solution. This claim will be investigated further in 8.3.7.

First-improve

Nikolajsen [Nik09] altered the ABHC in the following way in order to possibly speed up the algorithm: During the exhaustive neighbourhood search, if a move would improve the global best solution it would immediately be taken (checked in line 9) and the remaining search of the neighbourhood would be discarded. Algorithm 15 shows the changes applied to the standard algorithm.

The change does in no way guarantee a speedup of the algorithm, since the move towards a local minimum may become slower. On the other hand, the premature termination of the exhaustive neighbourhood search may compensate for the slower convergence. In 8.3.7 it will be investigated further, if the First-Improve strategy has a beneficial effect on the ABHC.

Caching

Inspired by M.Sc. Ph.D student Godskesen of IMADA, a caching of moves evaluated in each iteration of the algorithm will be investigated. Consider the following example: Assume that during iteration i we store the best move determined by the `relocate` neighbourhood as a move, which relocates customer c from route R_1 to R_2. During

Algorithm 15: First-Improve ABHC

Data: A routing plan R
Result: A routing plan R_r

$R_{best} \leftarrow R$;
$R_{current} \leftarrow R$;
while $R_{current} \neq null$ **do**
  IDENTIFYWORSTATTRIBUTES(R);
  $R_{current} \leftarrow null$;
  foreach $R_{\mathcal{N}} \in \mathcal{N}(R)$ **do**
    if $f(R_{\mathcal{N}}) < f(R_{current}) \wedge$ ACCEPT(R,R_n) **then**
      $R_{current} \leftarrow R_{\mathcal{N}}$;
      if $R_{current} < R_{best}$ **then**
        BREAK;
  if $R_{current} \neq null$ **then**
    $R \leftarrow R_{current}$;
    UPDATE$(R_{current})$;
    if $R_{current} < R_{best}$ **then**
      $R_{best} \leftarrow R_{current}$;
return R_{best};

iteration i we determine the best overall move to be an `exchange` move which exchanges customers d and e from routes R_3 and R_5 respectively. The algorithm moves accordingly by picking the `exchange` move and we update data structures according to the algorithm. In iteration $i+1$ we would normally re-evaluate the entire `relocate` neighbourhood, but this is not necessary since only routes R_3 and R_5 were changed in the previous iteration. Thus in iteration $i+1$ we restrict the exhaustive neighbourhood search to only consider the cached moves of iteration i which included routes R_3 and R_5 (those moves are invalidated in the cache and need to be recomputed). The remaining moves, which did not involve routes R_3 and R_5, are still valid. In essence, this is equivalent to pruning the neighbourhood.

Since solutions are already stored in order to add stochastic behaviour to the ABHC, the changes needed for adding stochastic caching are minimal (the cached move to pick is chosen randomly as defined by equation (7.4.1)). Because of the stochastic behaviour of the caching, the results obtained with the feature enabled, are not directly comparable to results obtained without it.

As mentioned previously caching was also used to optimize the performance of the subtour relocation neighbourhood operator.

Granular Neighbourhoods

Toth and Vigo describe an effective form of neighbourhood pruning for the Tabu Search algorithm called Granular Tabu Search [TV03]. It will be investigated if this neighbourhood pruning may be implemented within the ABHC framework. In short the neighbourhood moves are filtered according to a granularity threshold defined in the following way:

$$\vartheta = \beta \cdot \frac{z^{'}}{(n+K)^{'}}$$

where β is a suitable *sparsification parameter* and $z^{'}$ is the value of a heuristic solution determined by any fast heuristic. n denotes the number of customers in the problem and K denotes the number of available vehicles in the problem. The granularity threshold is used to limit the edges which may be considered in a move, i.e. the cost (length) of the edge may not exceed the threshold. Certain edges are, however, always considered legal: Edges incident to the depot as well as the edges of the best solution found. The intuition behind the granular neighbourhood is to limit the evaluation of moves that try to insert "long" arcs in the solution thus effectively pruning the neighbourhood.

In the implementation the value of β is set to 1.25 as suggested in [TV03], and increased to 1.75 if the number of non improving iterations reaches the threshold $n_d = 15 \cdot n$. Once an improving solution is found, β is reset to 1.25.

Chapter 8

Experimental Results

The following chapter is devoted to the experimental analysis of the implemented algorithms. The two sections that follow describe preliminary studies on known benchmark instances from the literature in order to verify the validity and quality of the solutions generated by the algorithms.

In the following section, the benchmark instances by Chao [Cha02] are briefly studied, after which experiments were conducted on the VRPTW benchmarks by Solomon [Sol87]. Finally, a more thorough investigation of the algorithms performance on the generated benchmarks is presented.

8.1 Chao Instances

The experimental research is started by testing the implemented algorithms on the benchmark instances by Chao (see 6.1). As mentioned previously, the main focus of this experiment is to check the validity of the model and therefore the following experiments are to be considered only preliminary and are included primarily as a proof of concept.

From a performance perspective in terms of running time, the implemented algorithms perform acceptably on the smaller instances with up to 100 customers. For larger instances the model is simply too inefficient in terms of computational time needed to obtain good quality solutions. This, however, is not necessarily a fault of the model itself, but most likely a consequence of the way the ETTRP model is not well suited to solve the TTRP instances. For a discussion on how to modify the implemented model to better fit with the model of Chao, the reader is referred to chapter 9.

Keeping the performance perspective in mind, this preliminary experiment is restricted to testing on the first nine TTRP instances p01-p09, since these contain between 50 and 100 customers. No parameter tuning was conducted for any of the experiments.

The initial test was run using the following setup:

1. The choice of seed for every route was the farthest unrouted customer from the depot.

2. The construction heuristic was chosen to be I1 without push-back using standard parameter settings: $\alpha_1 = 1.0, \alpha_2 = 0.0, \mu = 1.0, \lambda = 1.0$.

3. The starting solution was improved using a subsidiary local search procedure based on the neighbourhoods discussed in the previous chapter using the first improve strategy until a local optimum was reached.

4. Caching of moves was enabled.

5. Stochastic moves were enabled.

6. First-Improve strategy was chosen for the ABHC.

7. The Granular neighbourhood pruning was not enabled.

8. The algorithm was not allowed to restart.

Table 8.1 summarizes the results obtained by the ABHC. All results are the best of 10 runs for all algorithms. As can be seen, the ABHC seems to perform well in terms of solution quality, even improving on a previously best known solution for instance p02 which may also be seen in figure 8.1. In terms of running time the ABHC is up to a factor of 20 slower on average than the other algorithms, which is a clear drawback.

Chao TTRP Instances						
Instance	Chao Min.	Scheuerer Min.	Yu et al. Min.	ABHC Min.	Previously Best Known	Dev. (%)
p01	565.02	566.80	566.82	564.675	564.68	0.0
p02	662.84	615.66	612.75	**610.273**	611.53	**-0.21**[2]
p03	664.73	620.78	618.04	618.037	618.04	0.0
p04	857.84	801.60	808.84	798.529	798.53	0.0
p05	949.98	839.62	839.62	839.616	839.62	0.0
p06	1084.82	936.01	934.11	930.641	930.64	0.0
p07	837.80	830.48	830.48	830.478	830.48	0.0
p08	906.16	878.87	875.76	875.693	872.56	0.36
p09	1000.27	942.31	912.64	929.820	912.02	1.95

Table 8.1: Comparison of the Tabu Search algorithms of Chao and Scheuerer and the Simulated Annealing algorithm by Yu et al. as well as the ABHC implemented in this thesis. The reported results are the best of 10 runs for each algorithm. The column for best known solution lists solutions identified by heuristics. The last column denotes the deviation of the best found solution using ABHC compared to the best solution known.

The evolution of the best solution obtained on the first TTRP instance p01 is visualized in figure 8.2. Of interest is the great improvement in objective function value towards the end of the search, which indicates that, although no significant improvement could be made for some time, the algorithm might still improve its solution a great deal. This behaviour seemed to be reoccurring quite a lot on TTRP instances.

[2] The new best solution for the p02 instance is included in appendix B

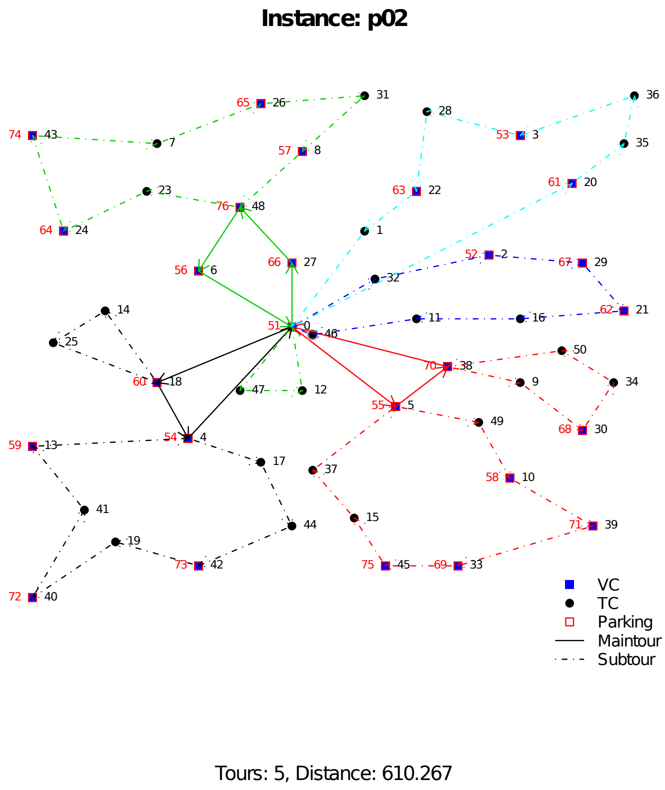


Figure 8.1: Best known solution for the TTRP instance p02

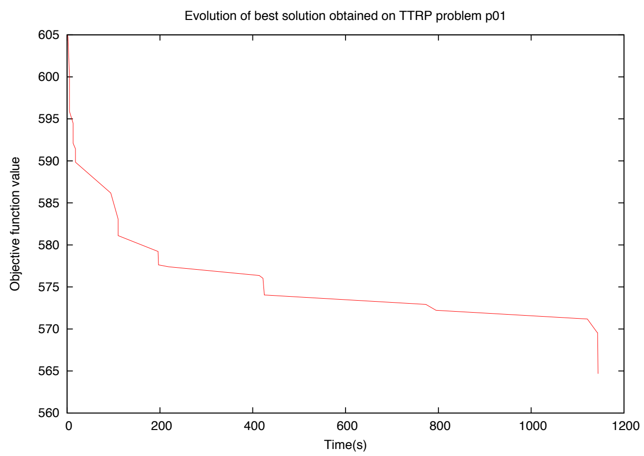


Figure 8.2: Algorithm progression for TTRP instance p01. The steep decline at the end of the search after a long period of vaguely improving moves shows that a premature end of the search might prevent the algorithm from obtaining valuable solutions.

8.1.1 Conclusion

Given the minimal parameter tuning during this preliminary test setup it may be concluded that the implemented algorithm performs acceptably in terms of solution quality, but that the required running time is too high when used as a TTRP solver. Furthermore, given the nature of the algorithm, one has to check the validity of the solutions since they do not necessarily respect the constraints given in the original TTRP, i.e. a parking place (associated with a customer in the TTRP) may indeed be used in another route as can be seen in figure 8.3.

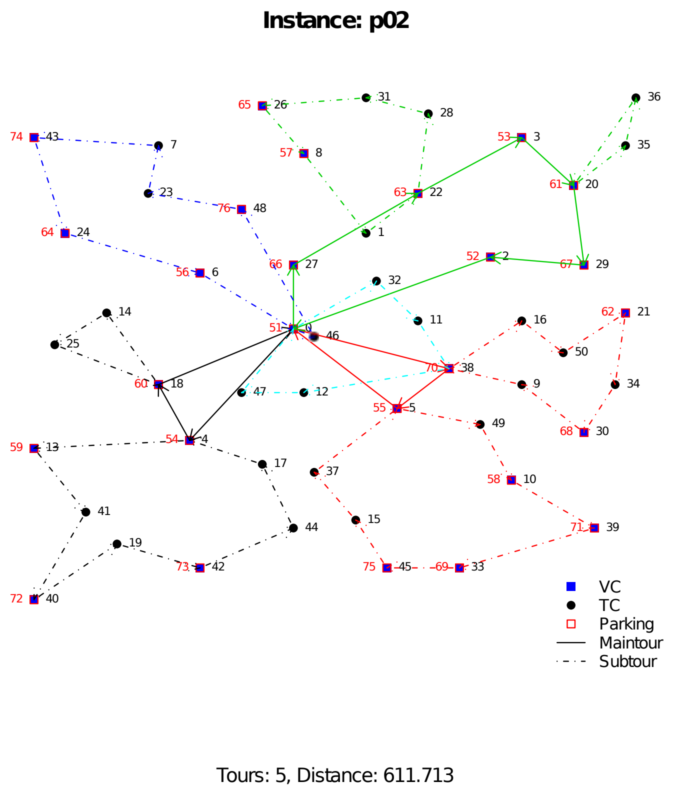


Figure 8.3: Example of illegal TTRP solution. Observe how customer with id 38 is serviced in the cyan subtour, yet it is used as a parking place in the red route.

Without further investigation, it is assumed that the main contributing factor to the high running time of the implemented algorithm is due to the costly updating of both temporal and load variables every time a move in the neighbourhood is executed. One could conduct a study of the real impact these updates have on the overall running time, but such a study will not be conducted in this thesis.

Further explanations of the algorithms high running time could be attributed to the somewhat restricted `subtour relocation` operator. Scheuerer [Sch04, p. 139] points out that the subtour root refining procedure is a valuable component of the Tabu Search algorithm he implemented. In this context it should be noted that his version of the subtour root refining procedure is able to shift an entire subtour onto a new root, which would require several moves in the implementation of this thesis.

A thorough study on this theory will not be treated in this thesis, however, since the shifting of an entire subtour is not likely to be possible when time windows and load feasibility have to be considered.

Scheuerer implemented a *Shift Move* and a *Swap Move* (This move is identical to the λ-Interchange) which are essentially identical to the `relocate` and `swap` operators implemented in this thesis, except that both of Scheuerers operators allow shifting or swapping of up to two consecutive customers, which might also allow his algorithm to quickly converge towards better solutions. Again, a thorough investigation would have to be conducted to shed light on such a claim.

8.2 Solomon Instances

A preliminary experimental analysis was conducted on the algorithms performance on the Solomon VRPTW benchmarks [Sol87], in order to check the validity and the quality of the solutions obtained on problems which have time windows. As was the case of the experiments on the TTRP benchmarks, no initial tuning of parameters was performed, and the setup equalled the following:

1. The choice of seed customer was chosen randomly among those described in section 7.2.
2. The construction heuristic was chosen to be I1 using the standard parameter settings: $\alpha_1 = 1.0, \alpha_2 = 0.0, \mu = 1.0, \lambda = 1.0$
3. The starting solution was improved using a subsidiary local search procedure based on the neighbourhoods discussed in the previous chapter using the first improve strategy.
4. Caching of moves was enabled.
5. Stochastic moves were enabled.
6. First-Improve strategy was chosen for the ABHC.
7. The Granular neighbourhood pruning was not enabled.

All tests were allowed to run for two hours of wall clock time, and the reported results are the best of 10 runs. The obtained results, which can be seen in table 8.2, all lie within 1% of the optimal solutions (for those of which an optimal solution is known), which all in all indicates that the implemented algorithm is capable of obtaining competitive results.

The best known results obtained by heuristic methods, although not clearly stated on the web page where the results were obtained, is assumed to use a different objective function, namely the one mentioned in section 3.3: minimizing employed vehicles then total distance. Thus we only perform a comparison in case the results obtained by the ABHC uses the same amount of vehicles in its solution, or if the run of the ABHC resulted in multiple solutions containing different number of vehicles employed.

100-customer Solomon instances													
Instance	ABHC Min. Dist.			Optimal			Opt. Dev.	ABHC Min. Vehicle			Heuristic		Heur. Dev.
	NV	Dist.	Time(s)	NV	Dist.	Time(s)	%	NV	Dist.	Time(s)	NV	Dist.	%
R101	20	1642.88	39	20	1637.7	1.87	0.32	19	1650.80	107	19	1645.79	0.30
R102	18	1472.81	935	18	1466.6	4.39	0.42	17	1486.12	807	17	1486.12	0.00
R103	14	1213.62	674	14	1208.7	23.85	0.41	-	-	-	13	1292.68	-
R104	11	976.61	944	11	971.5	23343.92	0.53	-	-	-	9	1007.24	-
R105	15	1364.35	1273	15	1355.3	43.12	0.67	14	1377.11	1225	14	1377.11	0.00
R106	13	1240.50	4014	13	1234.6	75.42	0.48	12	1258.93	3695	12	1251.98	0.56
R107	11	1074.94	1555	11	1064.6	1310.3	0.97	-	-	-	10	1104.66	-
R108	10	939.35	2027	10	932.1	5911.74	0.78	-	-	-	9	960.88	-
R109	12	1154.31	465	13	1146.9	1432.41	0.65	-	-	-	11	1194.73	-
R110	12	1074.41	1250	12	1068.0	1068.31	0.60	-	-	-	10	1118.59	-
R111	12	1054.06	749	12	1048.7	83931.48	0.51	-	-	-	10	1096.72	-
R112	10	957.84	1777	10	948.6	202803.94	0.97	-	-	-	9	982.14	-
C101	10	828.94	~0	10	827.3	3.02	0.20	10	828.94	~0	10	828.94	0.00
C102	10	828.94	~0	10	827.3	12.91	0.20	10	828.94	~0	10	828.94	0.00
C103	10	828.06	15	10	826.3	33.89	0.21	10	828.06	15	10	828.06	0.00
C104	10	824.78	155	10	822.9	4113.09	0.23	10	824.78	155	10	824.78	0.00
C105	10	828.94	~0	10	827.3	5.34	0.20	10	828.94	~0	10	828.94	0.00
C106	10	828.94	8	10	827.3	7.15	0.20	10	828.94	8	10	828.94	0.00
C107	10	828.94	~0	10	827.3	6.55	0.20	10	828.94	~0	10	828.94	0.00
C108	10	828.94	23	10	827.3	14.46	0.20	10	828.94	23	10	828.94	0.00
C109	10	828.94	13	10	827.3	20.53	0.20	10	828.94	13	10	828.94	0.00
RC101	15	1635.51	452	15	1619.8	12.39	0.97	-	-	-	14	1696.94	-
RC102	14	1461.23	389	14	1457.4	76.69	0.26	-	-	-	12	1554.75	-
RC103	11	1267.01	2536	11	1258.0	2705.78	0.72	11	1267.01	2536	11	1261.67	0.42
RC104	10	1135.85	1858	10	1132.3	65806.79	0.31	10	1135.85	1858	10	1135.48	0.03
RC105	15	1519.27	292	15	1513.7	26.73	0.37	-	-	-	13	1629.44	-
RC106	13	1378.23	452	13	1372.7	15891.55	0.40	-	-	-	11	1424.73	-
RC107	12	1213.99	261	12	1207.8	153.8	0.51	-	-	-	11	1230.48	-
RC108	11	1120.31	2123	11	1114.2	3365.0	0.55	-	-	-	10	1139.82	-
R201	8	1147.80	2257	8	1143.2	139.03	0.40	-	-	-	4	1252.37	-
R202	8	1036.51	4916	8	1029.6	8282.38	0.67	-	-	-	3	1191.70	-
R203	6	874.87	5160	6	870.8	54187.4	0.47	-	-	-	3	939.54	-
R204	5	736.80	4463	-	-	-	-	-	-	-	2	825.52	-
R205	5	955.82	5635	5	949.8	-	0.63	-	-	-	3	994.42	-
R206	5	879.89	3504	5	875.9	-	0.46	-	-	-	3	906.14	-
R207	4	797.99	7007	4	794.0	-	0.41	-	-	-	2	893.33	-
R208	3	706.74	5244	-	-	-	-	-	-	-	2	726.75	-
R209	5	859.39	1905	9	854.8	78560.47	0.54	-	-	-	3	909.16	-
R210	5	909.42	3495	6	900.5	-	0.99	-	-	-	3	939.34	-
R211	4	757.29	7061	-	-	-	-	-	-	-	2	892.71	-
C201	3	591.56	~0	3	589.1	203.34	0.42	3	591.56	~0	3	591.56	0.00
C202	3	591.56	~0	3	589.1	3483.15	0.42	3	591.56	~0	3	591.56	0.00
C203	3	591.17	23	3	588.7	13070.71	0.42	3	591.17	23	3	591.17	0.00
C204	3	590.60	34	3	588.1	-	0.43	3	590.60	34	3	590.60	0.00
C205	3	588.88	~0	3	586.4	416.56	0.42	3	588.88	~0	3	588.88	0.00
C206	3	588.49	~0	3	586.0	594.92	0.42	3	588.49	~0	3	588.49	0.00
C207	3	588.29	11	3	585.8	1240.97	0.43	3	588.29	11	3	588.29	.000
C208	3	588.32	17	3	585.8	555.27	0.43	3	588.32	17	3	588.32	0.00
RC201	9	1267.96	2609	9	1261.8	229.27	0.49	-	-	-	4	1406.91	-
RC202	8	1095.64	5183	8	1092.3	312.57	0.31	-	-	-	3	1367.09	-
RC203	5	929.04	4040	5	923.7	14917.36	0.58	-	-	-	3	1049.62	-
RC204	4	786.38	5749	-	-	-	-	-	-	-	3	798.41	-
RC205	7	1157.66	1051	7	1154.0	221.24	0.32	-	-	-	4	1297.19	-
RC206	6	1056.51	1689	7	1051.1	339.69	0.51	-	-	-	3	1146.32	-
RC207	6	966.08	6064	6	962.9	-	0.33	-	-	-	3	1061.14	-
RC208	4	781.23	2507	-	-	-	-	-	-	-	3	828.14	-

Table 8.2: Comparison of the ABHC, Optimal and Heuristic Solutions Found on the Solomon Benchmarks: The time unit is Wall-clock time. The results for optimal solutions are those reported by [JPSP] which were published April 2008. The objective function used for optimal solutions is that of route length minimization. Heuristic solutions were taken from [Sol05], last updated March 2005. The columns NV specify the number of vehicles used.

8.2.1 Conclusion

In some cases the use of exact methods is clearly justified as can be seen from the low running times used on some of the benchmark instances, and here heuristic methods such as the one implemented in this thesis are obviously inferior. The main point to conclude from the experiments, however, is that we may obtain good solutions to all VRPTW instances within a time frame of two hours. Again, it is worthwhile to note that although the implemented algorithms appear to be "slow", they were not designed specifically for solving problems which are strictly VRPTW.

8.3 Generated Instances

In this section the main results of running the implemented algorithms on the generated ETTRP instances is presented. A systematic tuning of algorithm parameters was conducted starting with those affecting the overall behaviour of the construction heuristics, and ending with parameters affecting the ABHC in general.

All experiments, except those used to calculate the Pareto Frontier of the generated instances, were conducted using the *race* package [Bir03] for the statistical computing software R [GI97]. The *race* package is a statistical framework for selecting the best configuration. The race was done unreplicated, meaning one run per setting on each instance. For elimination the Friedman test was used with a 95% confidence interval.

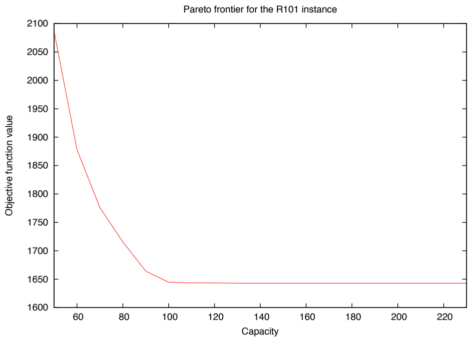
All results are available in appendix C.

8.3.1 Pareto Frontier of Generated Instances

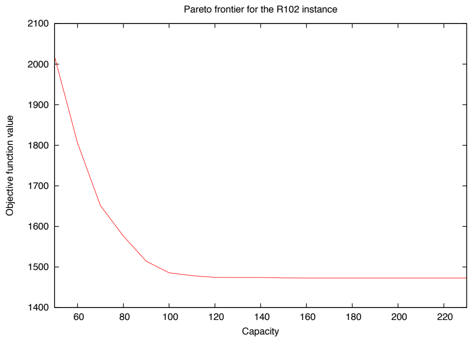
A preliminary study was conducted in order to determine the effects of the trucks capacity on the value of the objective function for each of the generated instances. All Pareto Frontiers were calculated using the implemented algorithms using the same parameter settings as was done for solving the VRPTW (see 8.2), without the use of trailers and a running time of 10 minutes CPU-time. Although the frontiers are not likely to be exact, since the values calculated were obtained by heuristics, it is still able to provide some insight regarding the structure of the instances.

As the figures 8.4-8.9 show, the value of the objective function is highly dependent on the capacity of the truck. Observe that as the capacity of the truck is increased the impact on the value of the objective function diminishes, until a certain point where increasing the capacity no longer improves the objective function. The cause of this is likely to be due to the time window constraints of the instances.

To be able to assess the performance of the algorithms better, and to be able to see an effect of solving the generated instances using trailers, it becomes clear that the capacities of truck and trailer have to be studied further, which will be discussed in the following section.

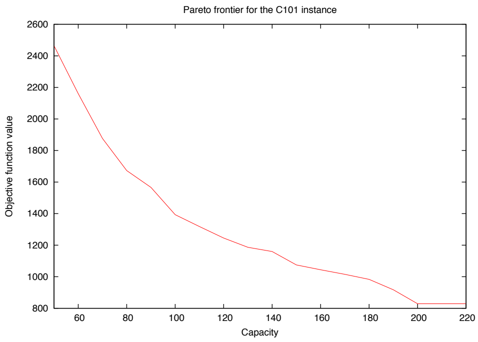


(a) Pareto frontier for instance R101

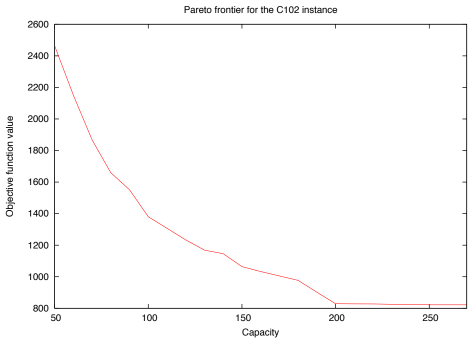


(b) Pareto frontier for instance R102

Figure 8.4: Pareto frontiers for instances R101 and R102. The figures show how the objective value appears to converge for truck capacities above 100.

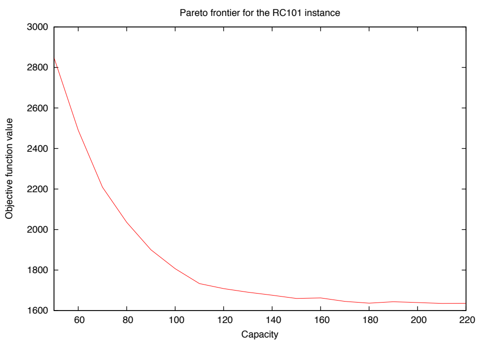


(a) Pareto frontier for instance C101

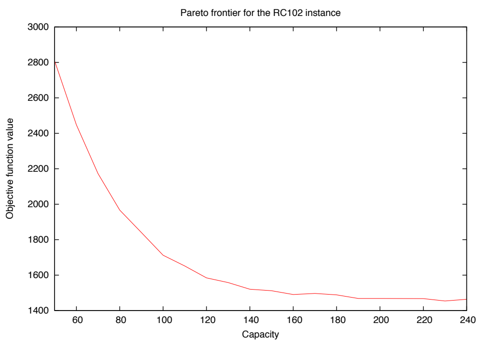


(b) Pareto frontier for instance C102

Figure 8.5: Pareto frontiers for instances C101 and C102. It appears as The objective value appears to converge for truck capacities above 200.

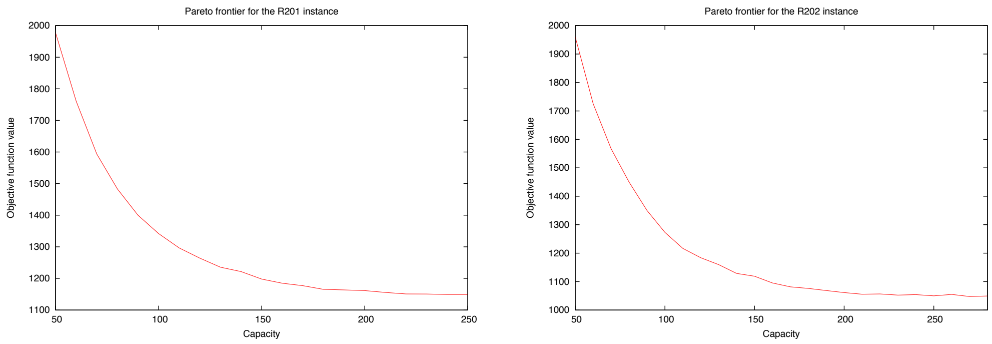


(a) Pareto frontier for instance RC101



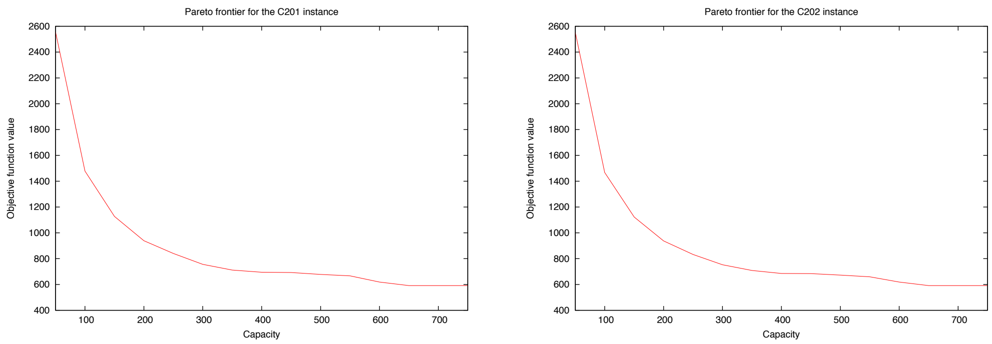
(b) Pareto frontier for instance RC102

Figure 8.6: Pareto frontiers for instances RC101 and RC102. The objective value appears to converge for truck capacities above 140.



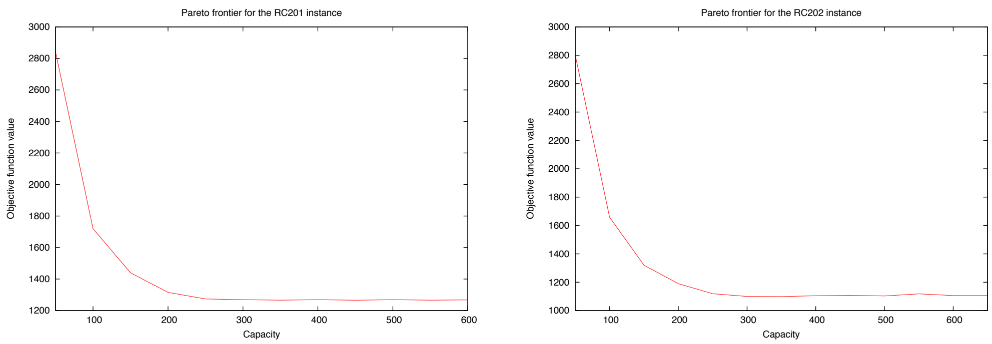
(a) Pareto frontier for instance R201 (b) Pareto frontier for instance R202

Figure 8.7: Pareto frontiers for instances R201 and R202. The objective value appears to converge for truck capacities above 150.



(a) Pareto frontier for instance C201 (b) Pareto frontier for instance C202

Figure 8.8: Pareto frontiers for instance C201 and C202. The objective value appears to converge for truck capacities above 300.



(a) Pareto frontier for instance RC201 (b) Pareto frontier for instance RC202

Figure 8.9: Pareto frontiers for instances RC201 and RC202. The objective value appears to converge for truck capacities above 250.

8.3.2 Additional ETTRP Instances

A number of additional instances were created based on the initial 36 which were described in section 6.3. In all aspects these additional instances are identical, except for the truck and trailer capacities. For each of the 36 generated instances a base instance was created with a maximum truck capacity of 50, as well as a maximum trailer capacity of 50 as shown in table 8.3. The lower limit of 50 was chosen since this is the lowest capacity for which all instances may feasibly be solved.

Problem ID	New Base	Capacity Increment	Capacity Limit
R101_25	R101_25_50_50	10	220
R101_50	R101_50_50_50	10	220
R101_75	R101_75_50_50	10	220
R102_25	R102_25_50_50	10	220
R102_50	R102_50_50_50	10	220
R102_75	R102_75_50_50	10	220
C101_25	C101_25_50_50	10	220
C101_50	C101_50_50_50	10	220
C101_75	C101_75_50_50	10	220
C102_25	C102_25_50_50	10	220
C102_50	C102_50_50_50	10	220
C102_75	C102_75_50_50	10	220
RC101_25	RC101_25_50_50	10	220
RC101_50	RC101_50_50_50	10	220
RC101_75	RC101_75_50_50	10	220
RC102_25	RC102_25_50_50	10	220
RC102_50	RC102_50_50_50	10	220
RC102_75	RC102_75_50_50	10	220
R201_25	R201_25_50_50	50	1200
R201_50	R201_50_50_50	50	1200
R201_75	R201_75_50_50	50	1200
R202_25	R202_25_50_50	50	1200
R202_50	R202_50_50_50	50	1200
R202_75	R202_75_50_50	50	1200
C201_25	C201_25_50_50	50	1200
C201_50	C201_50_50_50	50	1200
C201_75	C201_75_50_50	50	1200
C202_25	C202_25_50_50	50	1200
C202_50	C202_50_50_50	50	1200
C202_75	C202_75_50_50	50	1200
RC201_25	RC201_25_50_50	50	1200
RC201_50	RC201_50_50_50	50	1200
RC201_75	RC201_75_50_50	50	1200
RC202_25	RC202_25_50_50	50	1200
RC202_50	RC202_50_50_50	50	1200
RC202_75	RC202_75_50_50	50	1200

Table 8.3: Additional ETTRP benchmark instances. Here an instance R101_25_50_50 denotes the instance R101 with 25% truck customers and truck capacity of 50 as well as trailer capacity of 50. The column Capacity Increment denotes the capacity limits of the next instance i.e. R101_25_60_60 has the capacities incremented by 10. The column Capacity Limit denotes the capacities of the last instance within each group i.e. R101_25_220_220

Given the base instances from the table and the increment used to denote the capacities of the next generated instance within each group (corresponding to a row), a total of 756 instances were created.

8.3.3 Parameter Tuning of Construction Heuristics

In order to save time, the experiments involving the construction heuristics were based on values suggested by Solomon in [Sol87]. First, the parameters for each algorithm are listed after which, the results will be presented.

Insertion Heuristic I1: The parameters used for $(\mu, \lambda, \alpha_1, \alpha_2)$ were: $(1, 1, 1, 0)$, $(1, 2, 1, 0)$, $(1, 1, 0, 1)$, and $(1, 2, 0, 1)$. The route seed customers were according to the list of section 7.2.4.

In addition the effect of the improvements suggested by Dullaert and Bräysy [DB03] (see 7.2.2) were applied to the above combinations (whenever $\alpha_2 \neq 0$), resulting in 42 different parameter settings.

Insertion Heuristic I3: The parameters used, $(\mu, \alpha_1, \alpha_2, \alpha_3)$ were: $(1, 0.5, 0.5, 0)$, $(1, 0.4, 0.4, 0.2)$, and $(1, 0, 1, 0)$. The selection of seed customers equals that of Insertion Heuristic I1. The number of parameter settings to be tested thus equals 21.

Limited Subtour Demand: For each of the previous 63 parameter settings a ratio ν ranging from 10-100% was applied to the maximum allowable demand of subtours during route construction, resulting in 630 parameter combinations.

Randomized Parking Places during Route Construction: Although plausible in the context of the TTRP, the concept of clustering the truck customers around random parking places is less appealing in the ETTRP setting. The reason is obviously that a feasible starting solution may be impossible to obtain in case the travel time from the depot to the parking place and from there to the customer exceeds the latest time of service of the customer. Because of this, the concept of a randomized clustering approach was abandoned.

Results

The overall best configuration determined by the *race* was to use the Solomon Insertion Heuristic I1 using the following settings:

$$(\mu, \lambda, \alpha_1, \alpha_2, \nu, \text{push-back}) = (1, 2, 1, 0, 1, \text{disabled})$$

Thus the capacity of subtours should not be limited during construction ($\nu = 1$), the push-back improvements to insertion heuristic I1 should be disabled, and the choice of seed customer for new routes was the farthest unrouted customer.

A total of 5912 experiments using 189 instances were conducted in order to determine the best configuration. The results of the race are available in appendix C.1.

8.3.4 Refining the Starting Solution

Since truck customers were clustered around their closest parking place during the initial construction of subtours, it is obvious that every cluster has customers which are geometrically close. However, the customers of a cluster might be far apart when time windows are considered, thus the constructed subtours may potentially include high waiting times. To remedy this, it was proposed to run the ABHC on all subproblems using a refined objective function which minimizes the sum of the total route length and the total waiting time:

$$\omega(R) = \min \left[\sum_{k=1}^{K} L(R_k) + \mathcal{W}(R_k) \right] \tag{8.3.1}$$

This refinement step uses a simplified ABHC whose only neighbourhood is defined by the `relocate` operator, which should suffice considering only truck customers are concerned and no subtours exist. In order to further limit the computational time spent during the procedure, an upper limit of 300 non improving moves, that may be taken before the procedure ends, was introduced. Algorithm 16 captures how the overall route creation procedure is modified.

Algorithm 16: createRoutingPlan with Subtour Optimization

Data: Two sets of unrouted customers V_T (truck customers) and V_V (vehicle customers)
Result: A routing plan R

1 $R \leftarrow \emptyset$;
2 clusters $\leftarrow$ CLUSTER(V_T);
3 **foreach** $\rho \in$ clusters **do**
4     $V_V \leftarrow V_V \cup$ INSERTIONHEURISTIC(ρ);
5 CLUSTEROPTIMIZER(clusters);
6 $R \leftarrow$ INSERTIONHEURISTIC(V_V);
7 **return** R;

In line 5 the cluster optimization is initiated just prior to the regular route creation of line 6.

Results

A separate *race* was run in order to determine the best setting to use for the subtour optimization, thus a similar test was conducted as was done for the initial tuning of the construction heuristics, with the exception of the additional subtour optimization. Using the subtour optimization procedure after the initial construction determined the Insertion Heuristic I1 to be the best using the following configuration:

$$(\mu, \lambda, \alpha_1, \alpha_2, \nu, \text{push-back}) = (1, 2, 1, 0, 0.9, \text{disabled})$$

Thus subtours should be limited to 90% of the truck capacity during route construction, and push-back should not be enabled.

A total of 4998 experiments using only 172 instances were conducted in order to determine the best configuration. The results of the race are available in appendix C.2.

8.3.5 Configuration Results for the Construction Heuristic

To determine the best overall configuration for the construction heuristics, the best configurations determined previously were compared using a separate *race* which determined that the subtour optimization procedure was inferior to the best configuration determined in section 8.3.3:

$$(\mu, \lambda, \alpha_1, \alpha_2, \nu, \text{push-back}) = (1, 2, 1, 0, 1, \text{disabled})$$

The best configuration was determined after conducting 1002 tests using 501 instances. The results of the race are available in appendix C.3.

In all remaining tests the best parameter settings for the construction heuristics remains fixed to the setting above.

8.3.6 Results for the Construction Heuristic

Starting solutions generated by the construction heuristic using the best setting determined previously are available in tables D.1-D.6 in the appendix. As was the case for the Pareto frontier computations, the solution values seem to converge as the capacity of the vehicle is increased. A possible explanation for this behaviour is likely due to time window restrictions, this claim, however, should to be investigated further, but was not done in this thesis because of limited time. In the table the solution value for solving the instances using only trucks is listed for comparison. Thus the following observations may be made:

1. For low capacities of both truck and trailer the instances may be solved with fewer vehicles employed compared to solving the instances using only trucks.

2. The advantage of increased capacity is quickly lost in the randomized instances R101, R102, R201, and R202. In the clustered instances as well as the mixed instances the addition of a trailer appears to be beneficial for increasing capacities of truck and trailer.

3. In general the value of the objective function appears to converge faster for increased capacities compared to solving the problems without the use of trailers.

4. For large truck and trailer capacities the benefit of using a trailer for route construction purposes, seems negligible.

8.3.7 Parameter Analysis for ABHC

The parameter analysis for the Attribute Based Hill Climber was conducted using the *race* package described in section 8.3. The algorithms participating in the race setup consist of all combinations of the following behaviour modifying parameters:

- Starting from a local optimum i.e. by applying the local search procedure to the starting solution.
- Caching of moves. (As mentioned in section 7.4.4 caching adds stochastic behaviour and measuring the computational speedup is therefore not directly possible).
- First improve strategy instead of best improve strategy.
- Stochastic moves as mentioned in section 7.4.3.
- Granular neighbourhood as mentioned in section 7.4.4.

Resulting in a total of 32 combinations in a race, in which the algorithms were allowed a maximum of ten minutes CPU time.

Although the effects of caching are not directly comparable when regular moves are considered, the effects of caching parking places for the `subtour relocation` neighbourhood is. The speedup of caching which parking place to use for a relocated TC customer is illustrated in figure 8.10 by running the implemented algorithms on the instance C101_25_100_100 with and without caching enabled. As can clearly be seen, a considerable improvement in the time spent to reach a similar solution, is obtained.

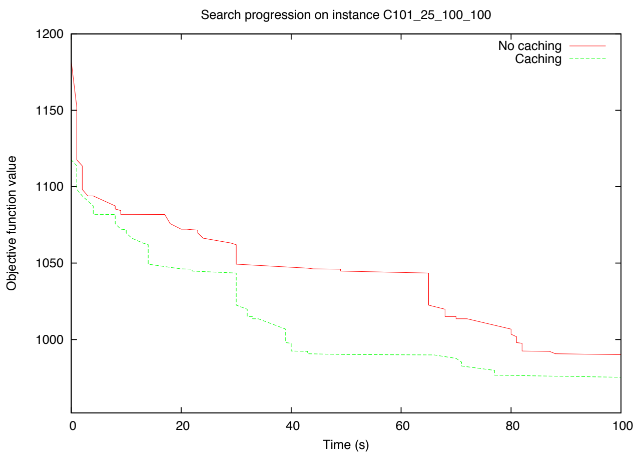


Figure 8.10: Search progression using caching of parking places for the subtour relocation neighbourhood for the instance C101_25_100_100. As can be seen the time needed to reach the same solution is significantly lower.

Results

Unfortunately the *race* for selecting the best configuration for the ABHC had to be terminated prematurely due to deadline considerations. Of the 32 initial tested configurations all except two had been eliminated when the race was ended, and thus the following two configurations may be considered equal:

- Caching of moves and stochastic behaviour.
- Caching of moves, stochastic behaviour, and first improve strategy.

A total of 122 instances as well as 568 experiments were conducted before the race was terminated.

In figure 8.11 the boxplot showing the two remaining configurations is shown. As can be seen, the quality of solutions produced by both appear to vary very little.

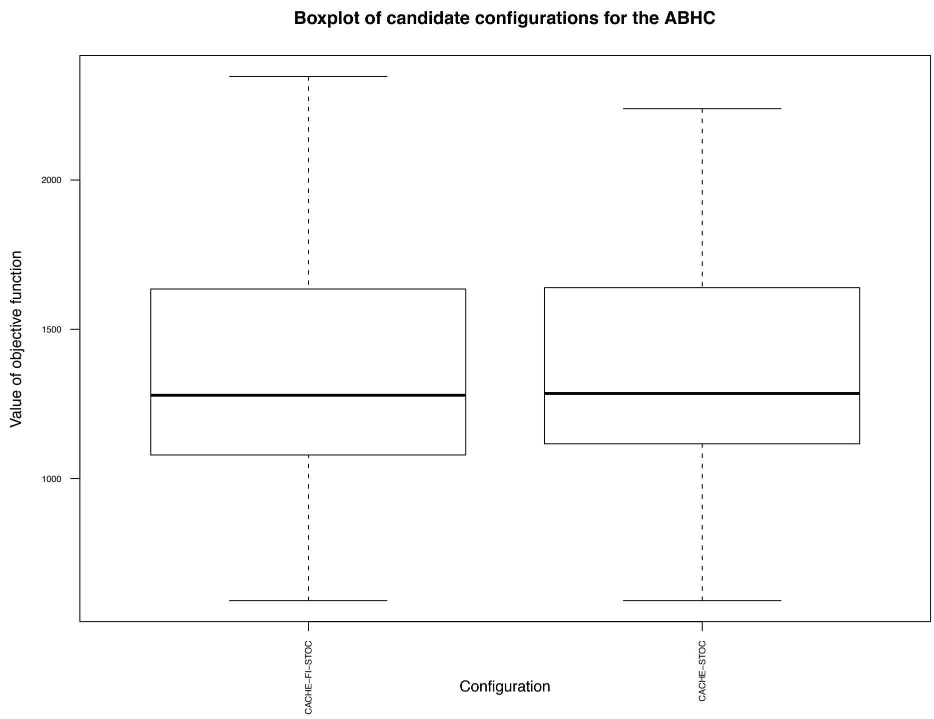


Figure 8.11: Boxplot of the two candidate configurations of the ABHC. As can be seen the quality of the solutions obtained by either configuration are very similar.

Therefore, it may ultimately be concluded that both caching and stochastic behaviour has a beneficial impact on the quality of the solutions obtained by the ABHC in the context of the ETTRP. In addition, the benefit from starting from a local optimum appears to be negligible since neither of the two remaining candidates used that strategy.

Based on these results, the configuration chosen for the final tests is the following:

ABHC Setting: Caching, Stochastic, First Improve strategy.

8.3.8 Final Results

Using the best found configurations for both construction heuristics and the ABHC the best results obtained for the generated ETTRP instances is presented in table 8.4. In addition, the quality of the solutions is compared to the Pareto frontiers generated in 8.3.1, in order to determine if, and how big an impact the use of trailers has. The algorithm was allowed a total of 10 minutes CPU time, and the reported results are the best of 10 runs. For each instance the capacity is given in the leftmost column and thus indicates the limit of a truck. When solving the instances with both a truck and a trailer, the capacity defines the limit on the trailer as well. Hence if the capacity is listed as 50 then the total capacity of a complete vehicle is 100.

Randomized ETTRP instances R101 and R102 (Short scheduling horizon)

As figures 8.12 and 8.13 indicate, the value of the objective function quickly converges and becomes relatively stable for capacities above 100. The incentive to use trailers for problems of this kind is thus complicated by the fact that the extra capacity granted by the use of a trailer quickly becomes insignificant. As previously mentioned (8.3.1) the likely cause of this is due to the short scheduling horizon of the problem i.e. the time windows becoming the dominating constraint.

As the number of truck customers increases, the number of employed trailers decreases. A possible explanation for this might be due to a combination of both load and time window constraints. In these instances, having many truck customers and an overall short scheduling horizon, might make the use of trailers infeasible or unattractive since the extra distance involved in parking the trailers could make further visits impossible within the allowed time limits.

Clustered ETTRP instances C101 and C102 (Short scheduling horizon)

Judging from the obtained results, the incentive to use trailers for solving the clustered ETTRP instances C101 and C102 is much higher. As can be seen by table 8.6 and figure 8.14 both distance and number of vehicles may be reduced. As was the case for the randomized instances R101 and R102. The use of trailers becomes irrelevant once the time windows become the dominant constraint, i.e. for capacities above 200. The results indicate that both the number of employed vehicles as well as the overall distance travelled may be reduced by the use of trailers for capacities of up to 180.

Random and Clustered ETTRP instances RC101 and RC102 (Short scheduling horizon)

The results for the random and clustered instances RC101 and RC102 mimic those of the clustered instances except for two things:

1. The use of trailers in the instances with 75% truck customers appears to be undesirable even for low overall capacities.

2. The convergence happens for lower capacities, thus supporting (1).

Random and ETTRP instances R201 and R202 (Long scheduling horizon)

The random instances R201 and R202 show the convergence when solving the instances using trailers earlier than the case where no trailer is used. This is most likely due to a too early termination of the search. In general, however, it appears as if the incentive to use trailers is higher when compared to the randomized instances in which the scheduling horizon was shorter i.e. R101 and R102, most notably, however, when the number of truck customers is below 75%.

Clustered ETTRP instances C201 and C202 (Long scheduling horizon)

The clustered instances C201 and C202 show a somewhat small improvement in the objective function value, however, the number of employed vehicles is significantly lower when compared to solving the instances without the use of trailers.

It is worthwhile to note that specifically for the instance C201 the number of employed trailers is often zero and thus the solution in these cases mimics the solution obtained by solving the instance with the truck alone. An explanation for the lack of employed trailers might be due to the fact that a great percentage of a cluster of customers may consist of truck customers.

Random and Clustered ETTRP instances RC201 and RC202 (Long scheduling horizon)

In these instances the incentive to employ trailers appears to diminish as the capacity exceeds 150. Again, the slightly worse results obtained here are likely to be explained by the limited computational time allowed for the algorithm.

Concluding remarks

As mentioned, and as can be seen, it is clear that solving the problems using a truck alone for capacities above the convergence threshold, becomes a lower bound on how good solutions may be obtained using trailers, since using a trailer in that case involves extra travel distance due to the fact that the trailer has to be parked. The reason why the implemented algorithms are not able to obtain these solutions is simply due to the limited computational time allowed in the tests. Given more time the solutions obtained will converge towards the lower bound as the search will favour solutions not containing trailers.

For the sake of argument a run of the algorithm on the ETTRP instance R101 with 75% truck customers, using truck and trailer capacities of 110, and allowing two hours CPU time, is included in appendix E. This particular setup was chosen since figure 8.12 would suggest the algorithms inability to produce a competitive result in that case.

In general it may be concluded that the proper use of trailers can have a positive impact on the quality of the solutions obtained. Time windows as well as load constraints should be considered factors which may reduce the benefit, the real impact of the latter still remains to be researched however.

Results for ETTRP instance R101											
Capacity	T. Dist.	NV	T+T Dist. (25% TC)	NV	NT	T+T Dist. (50% TC)	NV	NT	T+T Dist. (75% TC)	NV	NT
50	2086.45	31	1765.43	21	14	1859.39	25	11	1962.39	28	5
60	1878.24	26	1735.73	22	13	1806.87	22	10	1869.25	25	5
70	1776.01	23	1703.15	20	9	1760.32	22	10	1823.24	23	6
80	1715.62	21	1682.61	20	14	1733.14	22	9	1745.96	21	3
90	1664.16	20	1680.47	21	11	1710.97	20	9	1743.56	21	6
100	1644.64	20	1678.32	20	15	1700.41	20	8	1728.99	21	4
110	1643.52	20	1664.86	20	13	1690.87	20	8	1754.51	21	6
120	1643.34	20	1663.04	20	12	1715.85	20	9	1747.16	21	5
130	1642.88	20	1685.24	20	13	1698.88	20	8	1743.74	20	5
140	1642.88	20	1677.37	20	14	1699.10	20	8	1715.10	20	4
150	1642.88	20	1672.74	20	12	1700.00	12	9	1738.39	21	4
160	1642.88	20	1672.44	21	12	1706.53	21	10	1726.55	21	3
170	1642.88	20	1667.37	20	11	1710.43	20	8	1741.13	22	4
180	1642.88	20	1660.70	20	12	1723.68	20	8	1704.71	21	4
190	1642.88	20	1677.39	20	13	1697.89	20	9	1718.16	20	4
200	1642.88	20	1672.35	20	12	1705.92	20	9	1731.29	21	5
210	1642.88	20	1653.85	20	10	1702.74	21	10	1724.04	21	5
220	1642.88	20	1666.41	20	12	1699.80	12	11	1753.48	21	3

Table 8.4: Compilation of results for ETTRP instance R101. The column T. Dist shows results using a truck of the specified capacity alone. The columns denoted T+T Dist. show results obtained by solving the instance using both truck and trailer. NV denotes the total number of vehicles used, while NT denotes the number of trailers used.

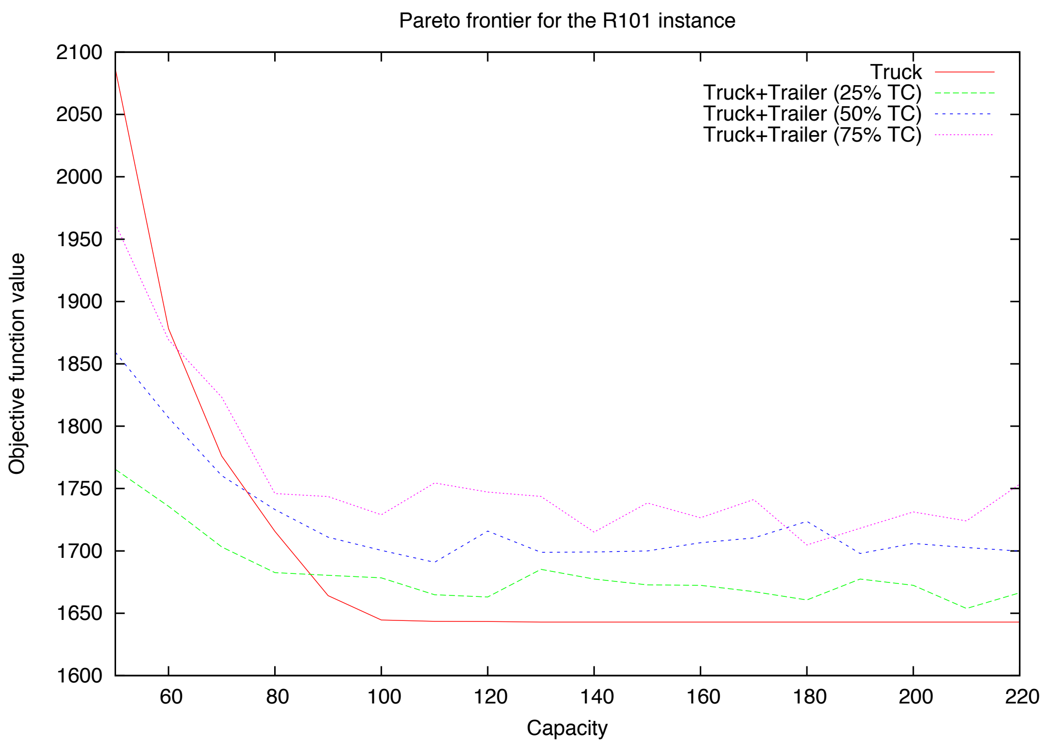


Figure 8.12: Pareto frontiers for ETTRP instance R101

Results for ETTRP instance R102											
Capacity	T. Dist.	NV	T+T Dist. (25% TC)	NV	NT	T+T Dist. (50% TC)	NV	NT	T+T Dist. (75% TC)	NV	NT
50	2017.98	31	1623.83	20	13	1761.53	23	12	1927.60	28	7
60	1805.78	26	1572.47	20	12	1685.28	21	9	1783.05	24	6
70	1651.57	22	1550.14	19	12	1619.68	19	9	1693.37	21	5
80	1576.27	20	1526.18	19	11	1591.04	19	9	1626.04	20	4
90	1514.30	19	1503.71	18	13	1604.70	19	9	1584.78	19	5
100	1485.71	18	1519.71	18	14	1596.10	19	8	1602.61	19	4
110	1478.74	18	1510.22	18	12	1569.49	19	9	1566.09	18	3
120	1474.46	18	1498.71	18	12	1582.27	19	9	1592.14	19	6
130	1474.17	18	1491.60	18	11	1593.38	19	10	1582.04	19	4
140	1474.17	18	1497.71	19	13	1619.83	19	10	1550.55	18	4
150	1473.10	18	1513.59	18	11	1566.46	19	10	1587.46	18	3
160	1472.81	18	1499.14	19	12	1570.92	18	9	1567.71	19	4
170	1472.81	18	1497.36	19	12	1579.09	18	8	1566.11	18	3
180	1472.81	18	1494.05	18	11	1588.07	19	10	1584.39	19	5
190	1472.81	18	1494.55	18	13	1569.74	18	9	1584.56	18	6
200	1472.81	18	1497.41	19	12	1562.11	19	8	1533.16	18	2
210	1472.81	18	1494.45	19	12	1550.99	18	9	1567.99	18	6
220	1472.81	18	1498.72	19	13	1603.03	19	9	1559.29	19	5

Table 8.5: Compilation of results for ETTRP instance R102. The column T. Dist shows results using a truck of the specified capacity alone. The columns denoted T+T Dist. show results obtained by solving the instance using both truck and trailer. NV denotes the total number of vehicles used, while NT denotes the number of trailers used.

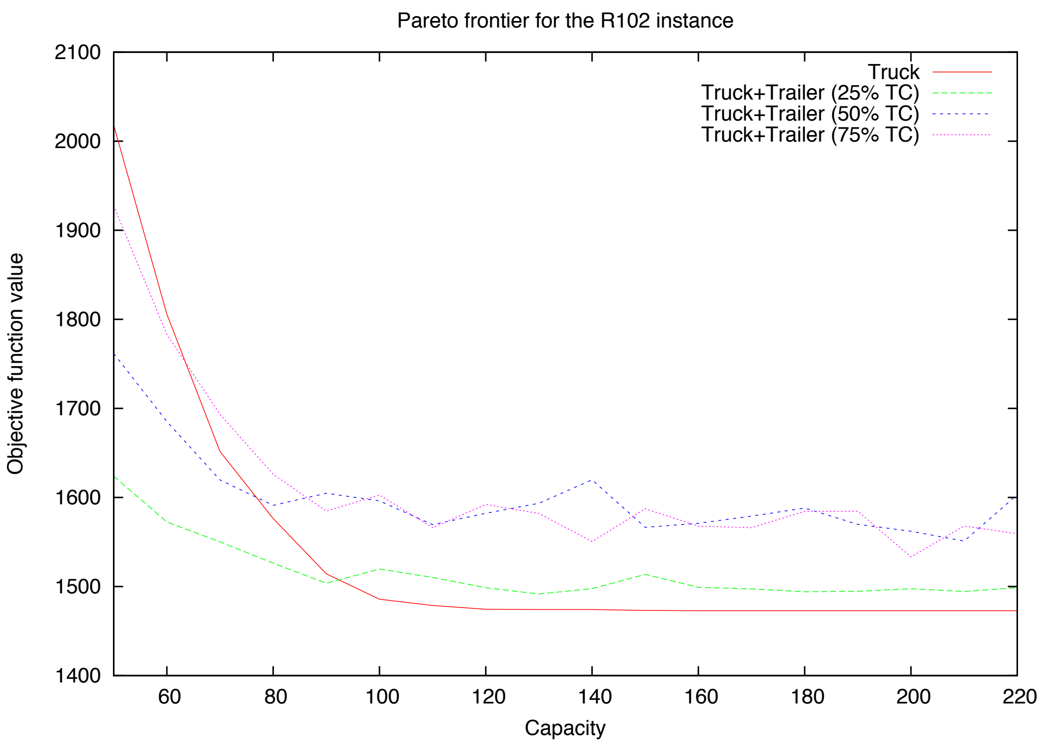


Figure 8.13: Pareto frontiers for ETTRP instance R102

Results for ETTRP instance C101											
Capacity	T. Dist.	NV	T+T Dist. (25% TC)	NV	NT	T+T Dist. (50% TC)	NV	NT	T+T Dist. (75% TC)	NV	NT
50	2464.16	37	1603.93	22	16	1916.14	27	12	2179.63	30	8
60	2160.42	31	1410.96	19	15	1666.97	23	11	1965.69	27	6
70	1876.59	26	1292.33	17	12	1564.39	20	10	1709.41	23	7
80	1672.81	23	1175.23	15	9	1421.08	18	7	1568.63	21	3
90	1566.06	21	1108.54	13	9	1370.17	17	7	1460.85	19	4
100	1393.45	19	1002.19	12	8	1233.34	17	3	1373.16	18	4
110	1318.08	17	953.18	11	8	1181.61	14	5	1313.65	16	2
120	1245.29	16	925.93	11	8	1095.89	13	6	1200.36	15	3
130	1186.75	14	925.68	11	8	1065.47	13	4	1118.42	14	3
140	1159.83	13	923.01	11	8	1019.94	12	4	1076.39	13	3
150	1074.91	13	910.86	10	8	996.47	12	3	1040.38	12	2
160	1044.10	12	893.62	10	8	989.50	12	3	997.26	12	3
170	1015.32	11	877.13	10	8	941.44	11	3	959.06	11	3
180	983.23	11	844.91	10	5	933.18	11	3	957.14	11	2
190	917.44	11	838.79	10	4	926.39	11	3	921.51	11	2
200	828.94	10	845.25	10	5	834.19	10	1	840.26	10	1
210	828.94	10	844.25	10	4	838.19	10	2	842.20	10	1
220	828.94	10	845.19	10	4	834.19	10	1	846.22	10	2

Table 8.6: Compilation of results for ETTRP instance C101. The column T. Dist shows results using a truck of the specified capacity alone. The columns denoted T+T Dist. show results obtained by solving the instance using both truck and trailer. NV denotes the total number of vehicles used, while NT denotes the number of trailers used.

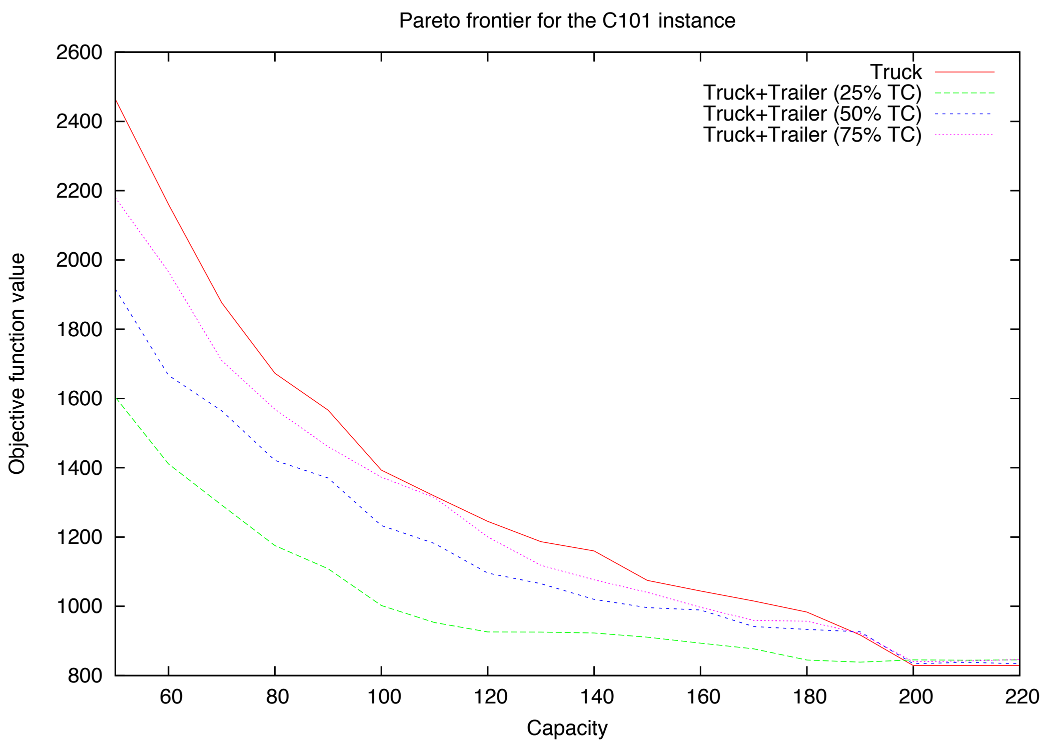


Figure 8.14: Pareto frontiers for ETTRP instance C101

Results for ETTRP instance C102											
Capacity	T. Dist.	NV	T+T Dist. (25% TC)	NV	NT	T+T Dist. (50% TC)	NV	NT	T+T Dist. (75% TC)	NV	NT
50	2463.64	37	1569.45	22	16	1907.68	26	12	2249.46	31	9
60	2151.04	31	1401.06	19	13	1659.49	22	11	1993.60	26	8
70	1867.36	26	1294.08	17	11	1532.11	19	9	1704.57	23	8
80	1659.03	23	1171.30	15	10	1440.31	18	7	1563.53	20	7
90	1551.19	21	1087.94	13	10	1345.53	16	9	1461.17	19	5
100	1380.17	19	954.49	12	8	1248.14	16	5	1359.38	17	4
110	1307.06	17	939.76	11	8	1206.14	13	7	1303.32	16	5
120	1233.43	16	926.73	11	8	1156.91	14	3	1151.99	14	4
130	1168.42	14	917.37	11	8	1106.55	12	4	1124.23	14	3
140	1146.06	13	912.22	11	8	1023.07	12	5	1066.30	13	3
150	1064.04	13	901.99	10	8	996.95	12	3	1038.94	13	2
160	1032.97	12	901.95	10	8	994.57	11	4	994.52	12	3
170	1005.02	11	869.44	10	6	978.39	11	4	980.93	11	3
180	976.85	11	861.87	10	6	945.98	11	4	966.19	11	3
190	901.85	10	861.04	10	7	906.31	10	4	905.81	10	2
200	828.94	10	858.74	10	6	855.68	10	3	839.35	10	1
210	828.03	10	861.48	10	6	851.68	10	2	845.39	10	1
220	827.55	10	842.60	10	5	855.51	10	3	853.21	10	2

Table 8.7: Compilation of results for ETTRP instance C102. The column T. Dist shows results using a truck of the specified capacity alone. The columns denoted T+T Dist. show results obtained by solving the instance using both truck and trailer. NV denotes the total number of vehicles used, while NT denotes the number of trailers used.

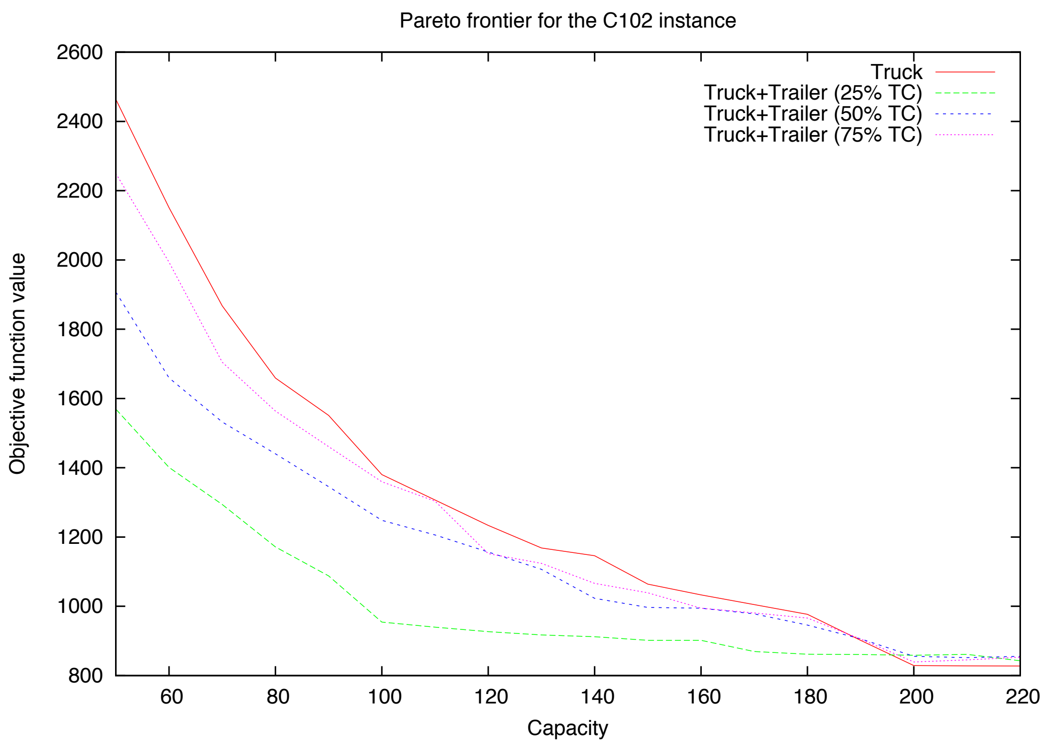


Figure 8.15: Pareto frontiers for ETTRP instance C102

Results for ETTRP instance RC101											
Capacity	T. Dist.	NV	T+T Dist. (25% TC)	NV	NT	T+T Dist. (50% TC)	NV	NT	T+T Dist. (75% TC)	NV	NT
50	2848.58	36	2229.48	23	16	2418.70	28	11	2665.57	32	7
60	2490.21	30	2035.66	20	15	2225.70	23	9	2363.62	27	5
70	2209.04	26	1908.85	19	13	2054.54	21	10	2160.15	25	3
80	2035.73	23	1862.28	18	13	1947.52	21	9	2013.97	22	3
90	1900.00	21	1835.89	18	12	1878.74	19	10	1914.21	20	4
100	1806.82	20	1792.44	18	13	1811.57	19	7	1826.79	19	2
110	1733.13	18	1755.97	17	11	1758.15	18	8	1744.61	18	0
120	1708.42	18	1726.20	17	6	1739.47	19	7	1729.23	18	2
130	1690.47	17	1726.26	17	11	1734.41	17	5	1722.81	17	4
140	1676.00	17	1703.23	17	10	1716.68	17	8	1697.45	17	0
150	1659.77	16	1719.97	16	9	1683.09	17	3	1688.26	18	2
160	1662.58	16	1703.05	17	9	1698.83	17	4	1688.94	18	3
170	1645.34	16	1695.05	16	8	1701.40	17	6	1647.50	16	2
180	1636.50	15	1692.56	16	10	1698.49	17	7	1674.54	17	1
190	1643.53	16	1687.16	17	10	1683.33	17	4	1668.86	17	1
200	1639.73	16	1696.34	17	10	1693.08	17	6	1664.44	16	2
210	1635.21	16	1667.86	16	8	1686.64	18	6	1677.30	16	3
220	1636.04	16	1695.45	17	9	1686.30	15	6	1656.26	16	3

Table 8.8: Compilation of results for ETTRP instance RC101. The column T. Dist shows results using a truck of the specified capacity alone. The columns denoted T+T Dist. show results obtained by solving the instance using both truck and trailer. NV denotes the total number of vehicles used, while NT denotes the number of trailers used.

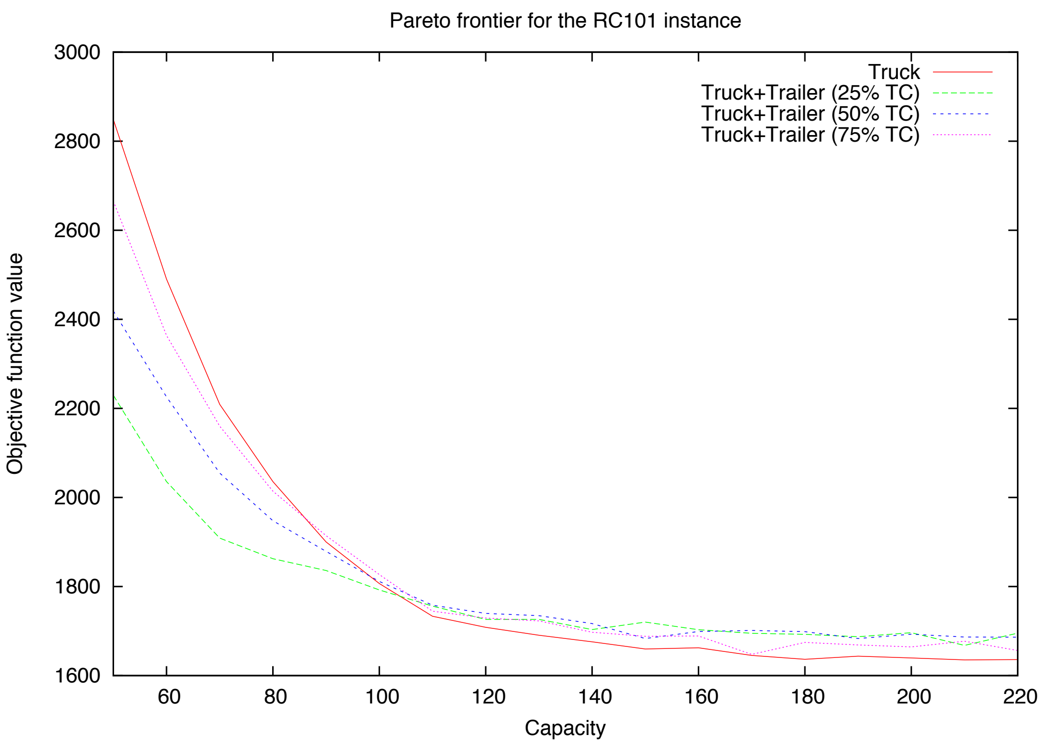


Figure 8.16: Pareto frontiers for ETTRP instance RC101

Results for ETTRP instance RC102											
Capacity	T. Dist.	NV	T+T Dist. (25% TC)	NV	NT	T+T Dist. (50% TC)	NV	NT	T+T Dist. (75% TC)	NV	NT
50	2808.43	36	2069.53	23	17	2400.20	26	14	2615.99	32	6
60	2448.00	30	1897.01	20	13	2147.87	22	13	2274.94	26	6
70	2172.66	26	1763.16	18	11	1965.31	21	10	2092.43	23	6
80	1966.79	22	1676.76	17	15	1830.52	19	9	1915.04	21	4
90	1840.41	20	1650.95	16	10	1778.54	18	8	1820.74	20	2
100	1712.11	18	1608.32	16	13	1699.04	18	6	1728.10	19	2
110	1651.59	17	1575.45	16	11	1641.24	17	9	1658.11	18	8
120	1584.57	16	1548.33	15	9	1631.80	16	7	1612.32	16	3
130	1557.84	16	1550.40	15	10	1605.85	16	8	1588.21	16	3
140	1520.35	15	1541.55	15	10	1571.23	15	7	1557.41	15	2
150	1512.14	14	1524.31	15	9	1599.55	15	10	1536.00	16	1
160	1490.65	14	1554.74	15	10	1611.05	15	11	1529.95	15	1
170	1496.77	14	1537.99	15	11	1579.04	16	10	1535.16	15	2
180	1488.77	14	1530.61	15	12	1576.02	15	9	1530.07	15	3
190	1468.27	14	1525.07	15	9	1570.94	15	7	1501.65	15	2
200	1468.12	14	1518.03	15	10	1591.61	15	11	1495.66	15	1
210	1467.96	14	1510.08	14	9	1558.38	15	8	1500.72	15	1
220	1467.65	14	1504.43	15	9	1573.58	15	8	1483.29	15	2

Table 8.9: Compilation of results for ETTRP instance RC102. The column T. Dist shows results using a truck of the specified capacity alone. The columns denoted T+T Dist. show results obtained by solving the instance using both truck and trailer. NV denotes the total number of vehicles used, while NT denotes the number of trailers used.

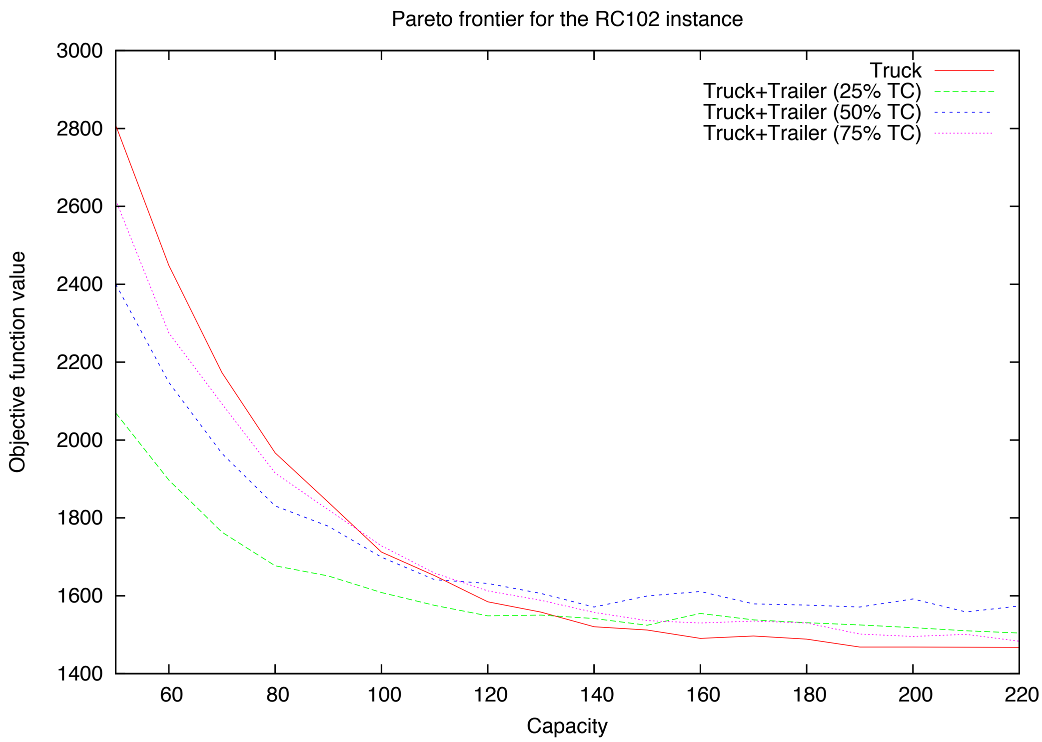


Figure 8.17: Pareto frontiers for ETTRP instance RC102

Results for ETTRP instance R201											
Capacity	T. Dist.	NV	T+T Dist. (25% TC)	NV	NT	T+T Dist. (50% TC)	NV	NT	T+T Dist. (75% TC)	NV	NT
50	1977.51	30	1523.22	18	15	1647.03	22	9	1814.30	27	6
60	1760.71	26	1402.68	16	11	1507.46	19	9	1662.82	23	5
70	1593.79	22	1344.64	15	8	1430.77	17	7	1551.62	20	5
80	1483.11	19	1296.82	13	8	1375.34	15	6	1493.91	17	5
90	1400.12	17	1265.56	12	7	1345.85	14	5	1445.64	16	4
100	1341.74	15	1244.47	11	10	1306.60	13	4	1363.43	15	2
110	1296.31	14	1230.14	11	7	1288.04	12	5	1328.96	14	1
120	1264.16	13	1230.41	10	8	1266.29	12	5	1327.79	13	2
130	1235.49	12	1217.72	10	8	1270.89	11	5	1298.72	13	2
140	1221.59	12	1218.79	10	6	1245.29	11	4	1275.11	13	2
150	1197.89	11	1207.59	9	6	1243.40	10	4	1244.67	12	1
160	1184.80	10	1211.18	9	7	1251.06	10	5	1249.44	11	1
170	1177.07	10	1207.36	8	5	1229.45	9	3	1249.38	10	2
180	1165.37	10	1207.22	10	6	1219.54	10	3	1207.02	11	1
190	1163.66	9	1193.60	9	5	1192.80	10	3	1207.52	9	0
200	1161.55	9	1220.96	10	9	1228.43	10	3	1233.05	10	1
210	1155.61	8	1210.82	10	6	1216.84	10	3	1190.45	9	0
220	1150.96	8	1201.58	9	3	1224.00	9	4	1228.27	11	1
230	1150.60	8	1218.86	9	6	1176.92	9	2	1192.70	10	0
240	1149.10	8	1200.18	9	6	1207.72	9	3	1188.82	8	1
250	1149.10	8	1197.49	10	7	1193.71	9	2	1194.33	10	0

Table 8.10: Compilation of results for ETTRP instance R201. The column T. Dist shows results using a truck of the specified capacity alone. The columns denoted T+T Dist. show results obtained by solving the instance using both truck and trailer. NV denotes the total number of vehicles used, while NT denotes the number of trailers used.

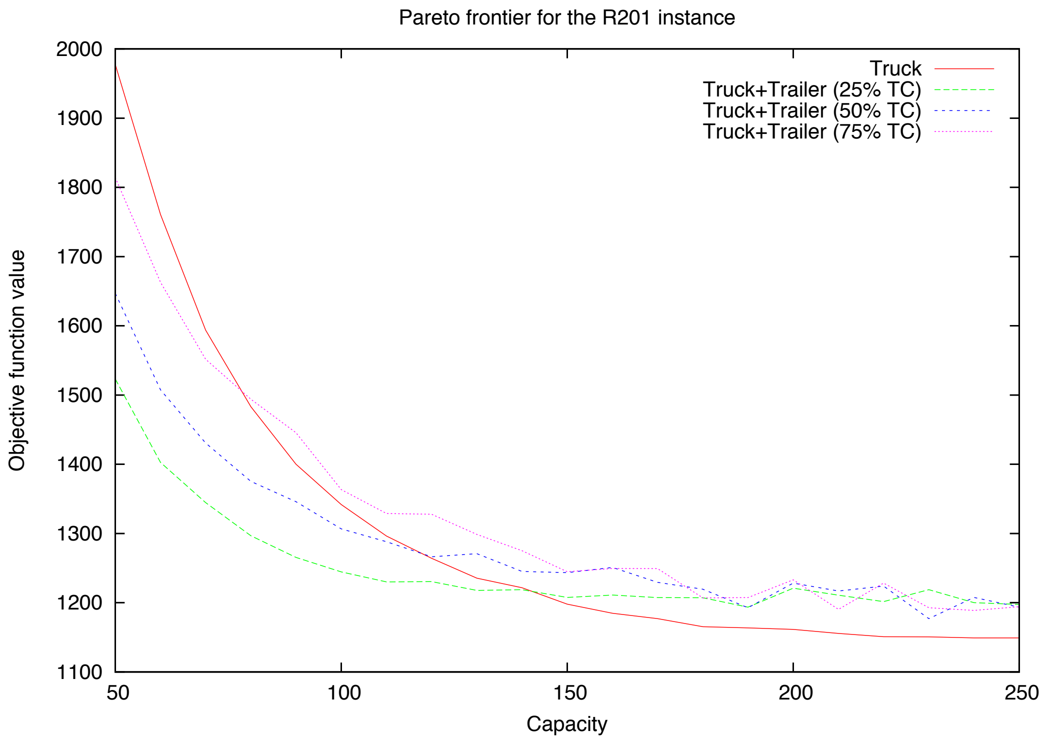


Figure 8.18: Pareto frontiers for ETTRP instance R201

Results for ETTRP instance R202									
Capacity	T. Dist.	NV	T+T Dist. (25% TC)	NV	NT	T+T Dist. (50% TC)	NV	NT	T+T Dist. (75% TC)	NV	NT
50	1959.86	30	1465.36	17	14	1597.88	21	10	1798.01	27	5
60	1725.12	26	1338.85	15	11	1460.89	19	8	1615.78	22	5
70	1567.19	22	1272.07	13	9	1358.54	16	7	1579.69	19	5
80	1450.18	19	1227.63	12	9	1294.31	13	7	1466.01	17	5
90	1350.00	17	1177.94	11	7	1263.83	12	7	1366.59	15	4
100	1273.10	15	1165.90	11	7	1222.49	12	6	1344.43	14	3
110	1216.66	14	1136.94	10	7	1200.49	11	5	1275.76	14	4
120	1183.94	13	1141.29	8	7	1185.63	10	5	1260.18	13	3
130	1159.53	12	1144.27	8	8	1203.23	10	5	1248.92	11	3
140	1128.77	11	1135.95	8	5	1195.57	9	5	1179.24	12	1
150	1118.95	11	1141.76	9	7	1179.12	8	6	1226.37	10	3
160	1095.22	10	1143.78	8	7	1142.32	9	4	1205.25	10	3
170	1081.43	9	1147.19	7	6	1166.49	7	5	1230.82	10	5
180	1076.10	9	1127.96	8	6	1170.46	9	4	1162.40	9	2
190	1068.46	8	1126.67	8	6	1172.30	7	4	1191.10	10	2
200	1061.41	8	1130.47	7	6	1183.29	9	6	1196.11	10	3
210	1055.60	8	1138.87	7	6	1184.08	9	6	1183.15	8	2
220	1056.47	8	1144.30	8	7	1177.70	8	5	1186.85	10	2
230	1052.38	8	1112.98	6	5	1165.82	8	4	1179.98	9	2
240	1054.16	8	1121.81	7	5	1183.83	7	6	1189.56	7	2
250	1049.78	7	1110.83	6	5	1155.68	7	4	1143.39	8	2
260	1054.82	7	1128.92	6	4	1170.10	8	5	1153.48	9	1
270	1047.48	7	1113.55	7	6	1179.25	7	6	1159.74	8	2
280	1049.26	6	1126.46	6	5	1134.31	9	4	1163.41	8	2

Table 8.11: Compilation of results for ETTRP instance R202. The column T. Dist shows results using a truck of the specified capacity alone. The columns denoted T+T Dist. show results obtained by solving the instance using both truck and trailer. NV denotes the total number of vehicles used, while NT denotes the number of trailers used.

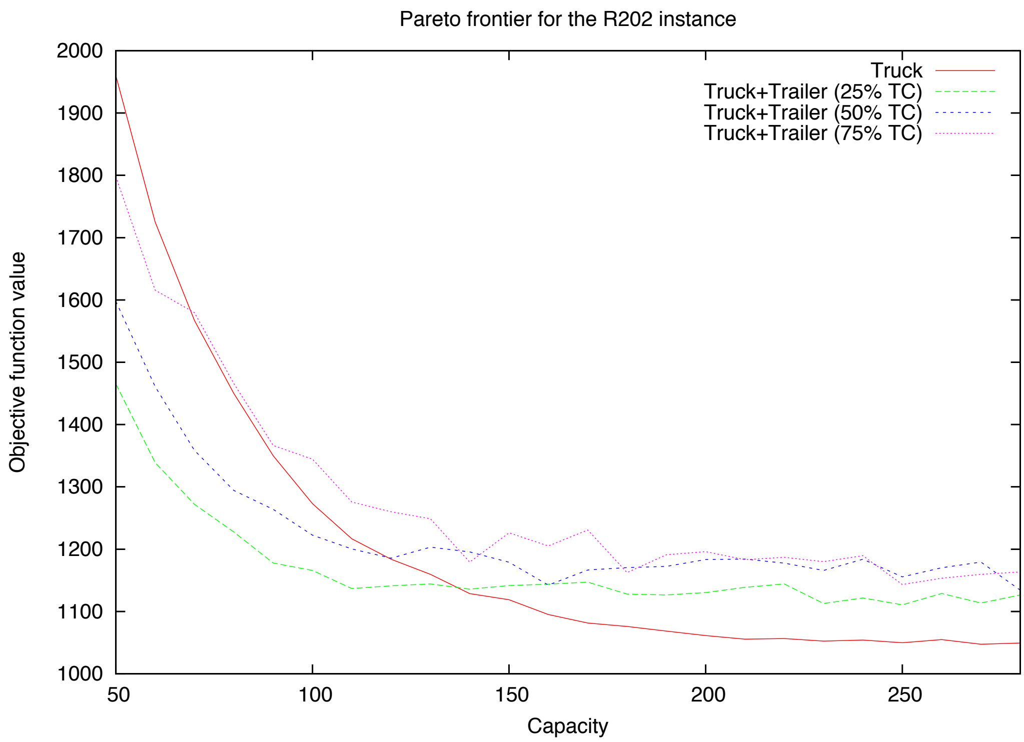


Figure 8.19: Pareto frontiers for ETTRP instance R202

Results for ETTRP instance C201											
Capacity	T. Dist.	NV	T+T Dist. (25% TC)	NV	NT	T+T Dist. (50% TC)	NV	NT	T+T Dist. (75% TC)	NV	NT
50	2556.05	37	1777.80	21	18	2025.04	26	14	2333.13	31	8
100	1478.13	19	1095.01	12	7	1251.58	14	6	1410.55	16	4
150	1127.40	13	883.62	7	6	1032.04	10	4	1108.68	11	3
200	938.70	10	809.06	6	4	905.93	8	3	951.57	9	1
250	840.50	8	768.92	5	4	828.65	7	2	863.89	7	1
300	756.00	7	714.44	5	2	756.00	7	0	758.58	7	0
350	711.46	6	699.82	5	2	744.31	5	1	711.46	6	0
400	695.02	5	681.04	5	1	716.87	6	1	702.48	6	0
450	692.52	5	673.95	4	2	737.19	5	2	692.69	5	0
500	677.99	4	670.05	5	1	706.21	5	1	682.38	5	0
550	666.87	4	660.50	4	1	679.75	4	1	666.87	4	0
600	618.62	4	624.42	4	1	618.62	4	0	618.62	4	0
650	591.56	3	616.20	4	1	591.56	3	0	591.56	3	0
700	591.56	3	615.78	4	2	591.56	3	0	591.56	3	0
750	591.56	3	615.78	4	1	591.56	3	0	591.56	3	0

Table 8.12: Compilation of results for ETTRP instance C201. The column T. Dist shows results using a truck of the specified capacity alone. The columns denoted T+T Dist. show results obtained by solving the instance using both truck and trailer. NV denotes the total number of vehicles used, while NT denotes the number of trailers used.

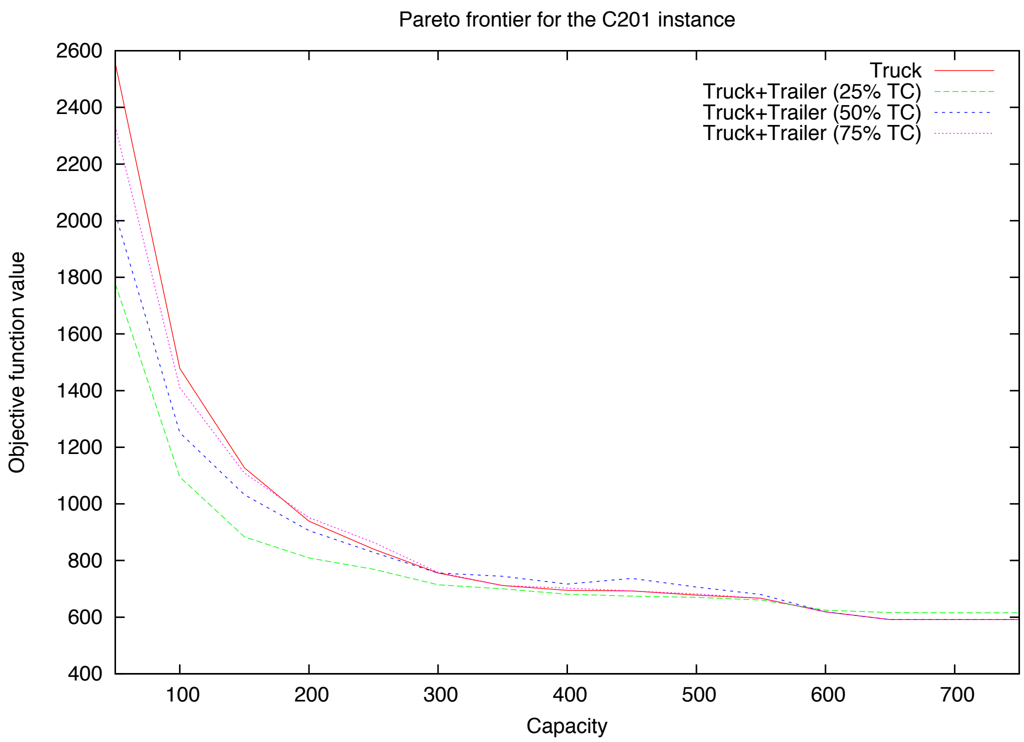


Figure 8.20: Pareto frontiers for ETTRP instance C201

Results for ETTRP instance C202											
Capacity	T. Dist.	NV	T+T Dist. (25% TC)	NV	NT	T+T Dist. (50% TC)	NV	NT	T+T Dist. (75% TC)	NV	NT
50	2553.58	37	1723.66	21	16	2014.86	26	12	2366.78	31	7
100	1467.15	19	1094.81	11	8	1248.11	14	7	1428.59	16	4
150	1122.44	13	874.82	8	6	1025.92	9	4	1117.11	11	3
200	937.24	10	802.12	6	5	914.69	7	4	992.92	8	4
250	832.59	8	753.34	5	4	856.24	6	3	901.12	8	2
300	752.72	7	721.70	4	3	792.22	6	3	850.62	7	3
350	708.55	6	697.88	5	2	774.01	5	2	786.91	6	2
400	685.24	5	692.15	5	2	745.67	5	1	775.78	6	2
450	684.47	5	688.00	4	3	747.91	5	2	779.32	6	2
500	672.76	4	673.78	4	2	723.77	5	1	761.92	5	2
550	659.55	4	671.41	4	2	701.03	4	1	717.18	5	1
600	618.62	4	682.53	4	2	677.15	4	2	677.65	4	1
650	591.56	3	661.93	3	2	661.07	4	1	666.08	4	1
700	591.56	3	657.81	4	2	653.11	4	1	620.00	4	1
750	591.56	3	699.49	4	3	663.84	4	1	591.56	3	0

Table 8.13: Compilation of results for ETTRP instance C202. The column T. Dist shows results using a truck of the specified capacity alone. The columns denoted T+T Dist. show results obtained by solving the instance using both truck and trailer. NV denotes the total number of vehicles used, while NT denotes the number of trailers used.

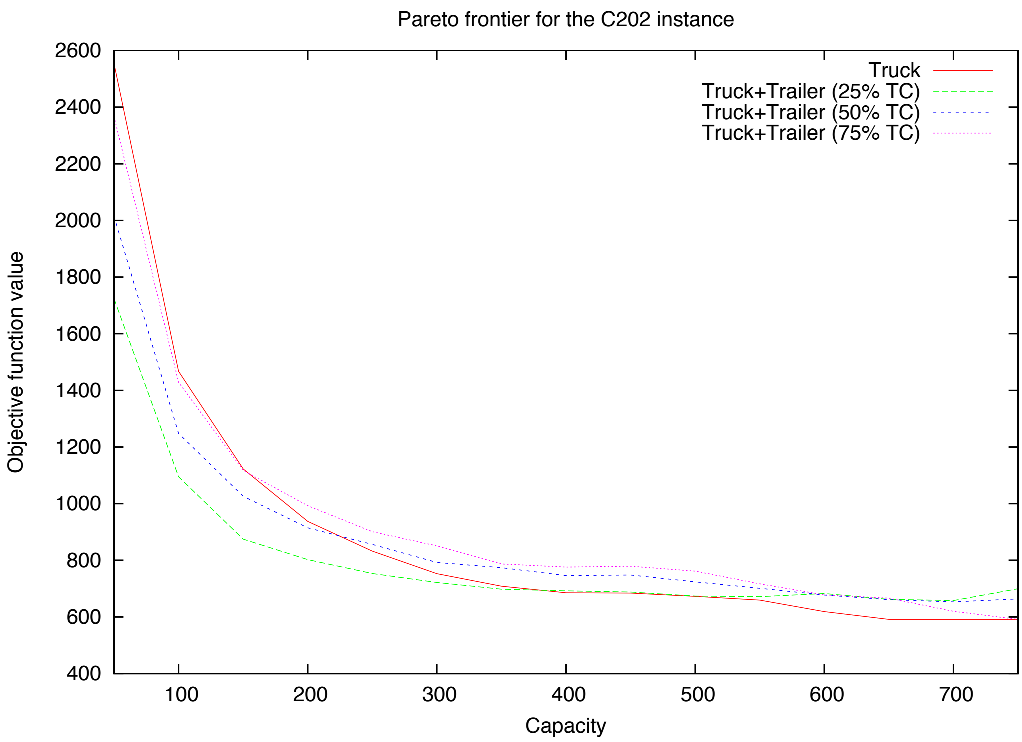


Figure 8.21: Pareto frontiers for ETTRP instance C202

Results for ETTRP instance RC201												
Capacity	T. Dist.	NV	T+T Dist. (25% TC)	NV	NT	T+T Dist. (50% TC)	NV	NT	T+T Dist. (75% TC)	NV	NT
50	2841.35	36	2100.62	22	16	2336.53	25	12	2641.22	31	6
100	1719.82	18	1495.34	12	9	1625.75	14	10	1706.22	17	3
150	1439.74	12	1382.37	11	9	1431.92	10	8	1473.14	13	2
200	1315.51	10	1310.30	10	6	1359.26	11	4	1332.36	10	0
250	1273.69	10	1275.95	10	5	1312.37	10	5	1305.20	10	2
300	1269.01	9	1291.71	9	7	1311.22	10	3	1283.10	10	1
350	1265.90	9	1282.50	10	6	1304.72	10	3	1306.28	10	2
400	1269.16	9	1284.28	9	6	1308.42	9	5	1316.08	9	2
450	1265.56	9	1291.95	9	6	1301.87	9	4	1310.03	10	3
500	1269.16	9	1283.08	10	4	1300.32	9	3	1294.54	9	1
550	1266.22	9	1289.51	9	5	1287.90	10	3	1306.87	10	1
600	1267.69	9	1284.41	9	6	1300.63	10	4	1314.61	10	2

Table 8.14: Compilation of results for ETTRP instance RC201. The column T. Dist shows results using a truck of the specified capacity alone. The columns denoted T+T Dist. show results obtained by solving the instance using both truck and trailer. NV denotes the total number of vehicles used, while NT denotes the number of trailers used.

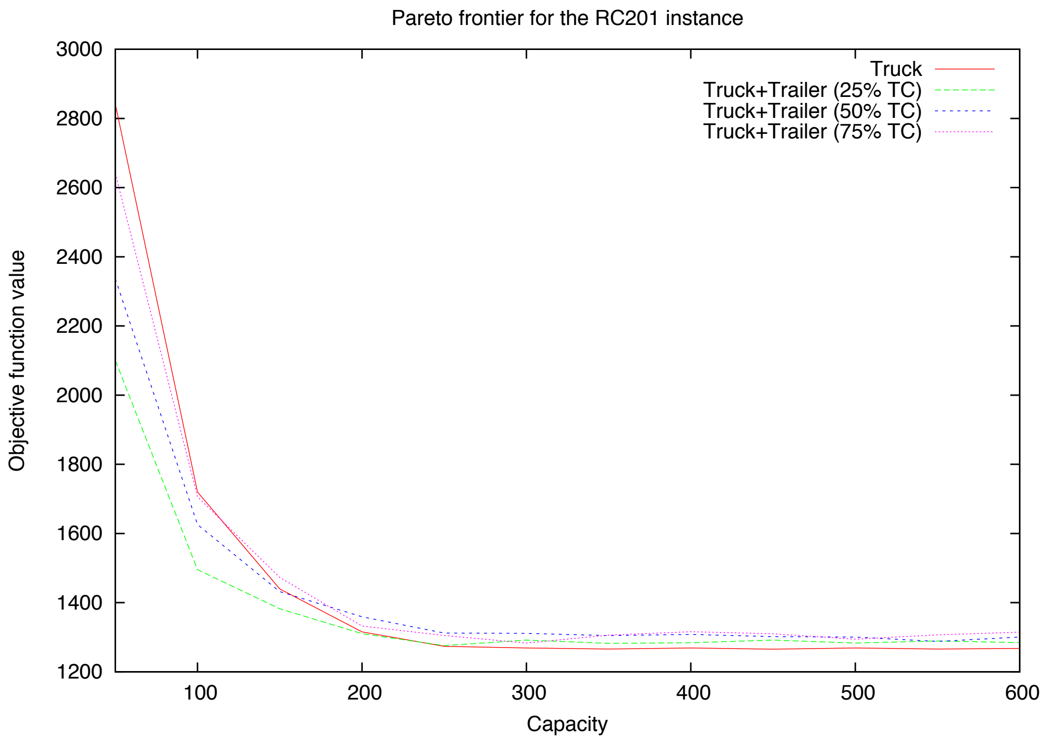


Figure 8.22: Pareto frontiers for ETTRP instance RC201

Results for ETTRP instance RC202											
Capacity	T. Dist.	NV	T+T Dist. (25% TC)	NV	NT	T+T Dist. (50% TC)	NV	NT	T+T Dist. (75% TC)	NV	NT
50	2801.32	36	1966.74	21	16	2298.48	26	12	2587.62	31	6
100	1658.36	18	1358.00	11	9	1507.93	14	8	1614.54	16	3
150	1319.70	12	1208.69	9	8	1316.68	10	5	1346.92	12	4
200	1189.60	9	1136.11	8	8	1222.24	10	8	1236.13	10	3
250	1119.07	9	1129.47	8	8	1199.32	10	8	1172.63	10	1
300	1099.54	8	1133.86	8	8	1208.83	9	7	1190.50	10	3
350	1098.80	8	1131.32	8	6	1154.11	8	5	1169.08	8	3
400	1104.27	7	1121.45	8	7	1193.42	8	6	1171.76	8	3
450	1107.28	8	1133.51	8	7	1155.35	9	7	1156.08	8	3
500	1103.03	7	1144.84	8	7	1161.15	8	6	1165.13	8	4
550	1117.75	8	1131.34	8	7	1158.54	9	5	1144.33	8	2
600	1105.87	8	1126.71	8	7	1156.06	9	7	1163.27	8	4
650	1106.15	7	1143.81	8	8	1144.51	8	5	1178.77	9	4

Table 8.15: Compilation of results for ETTRP instance RC202. The column T. Dist shows results using a truck of the specified capacity alone. The columns denoted T+T Dist. show results obtained by solving the instance using both truck and trailer. NV denotes the total number of vehicles used, while NT denotes the number of trailers used.

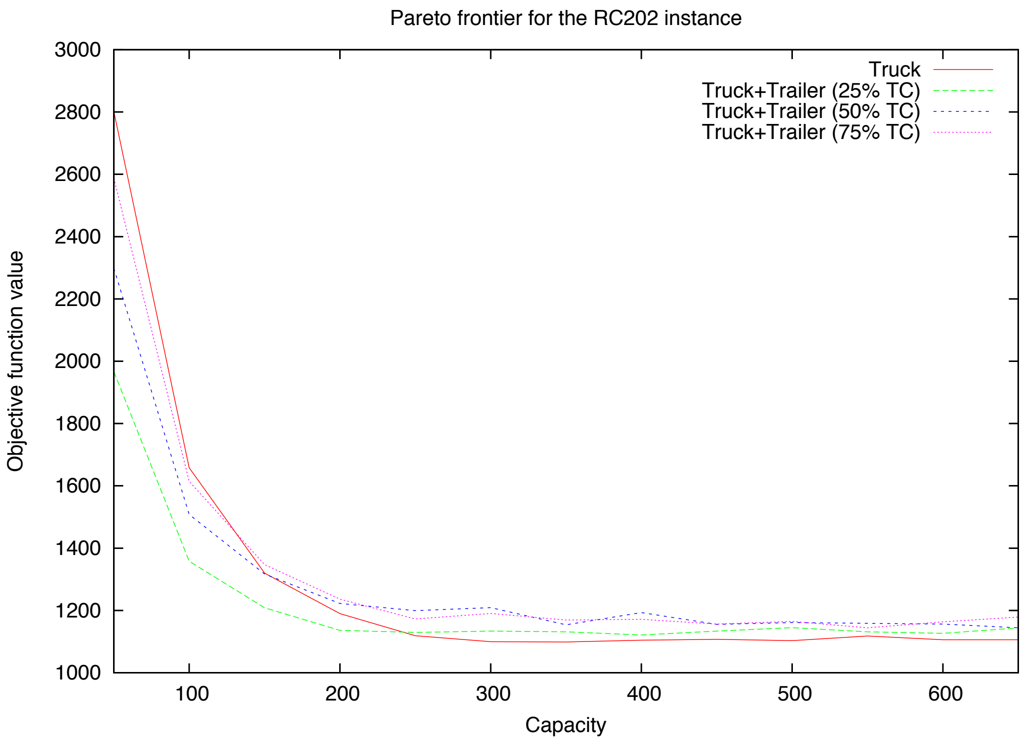


Figure 8.23: Pareto frontiers for ETTRP instance RC202

Chapter 9

Future Work and Perspective

Although the work contained in this thesis may describe a problem that appears as a simple VRP involving trailers, the actual amount of work put into the framework for solving the ETTRP is probably less obvious. As a consequence more time available for experiments and analysis of the implemented algorithms could have been desired. The choice of implemented extensions was to model time window constraints as well as load constraints, since it appeared as if these would be useful additions to the problem, especially in a real-world application. The remainder of this chapter describes areas where future work might elaborate on various details, which due to the limited time available for this thesis were not treated. Various other extensions to the model and a brief description on how these could have been implemented is also included.

9.1 Possible Model Adaptations

In this section, the possibilities of adapting our model in such a way that it may be used to either handle further constraints, or to be used to describe "simpler" models, will be explored.

9.1.1 Modelling Trailer Sharing

First, figure 9.1 is considered in order to devise the needed modifications of the model such that it may sufficiently express the additional constraints which should be handled.

In the figure we see a routing plan using three trucks as well as a single trailer. The figure is to be interpreted as follows: Truck 1 initially leaves the depot with the trailer coupled, and services a vehicle customer after which it de-couples the trailer at the parking place shown (1). Truck 1 then services a mixture of both truck and vehicle customers on its way back to the depot, shown to the left in the figure. Meanwhile, truck 2 which has no trailer coupled has left the depot and proceeds to service two vehicle customers after which it is headed for the parking place also visited by truck 1. Upon arrival at the parking place it re-couples the trailer left from truck 1, and proceeds to service a vehicle customer on its way to the next parking place shown to the lower right in the figure. Arriving at the parking place, truck 2 de-couples the

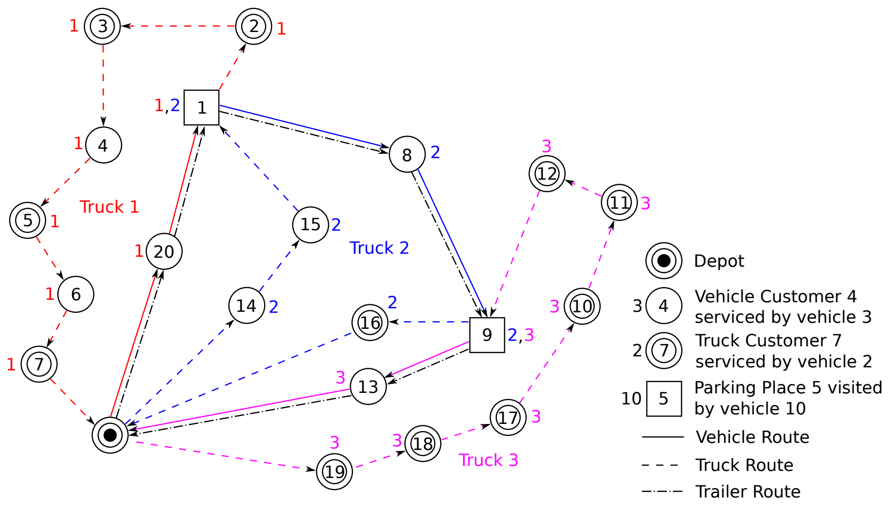


Figure 9.1: ETTRP Routing Plan using Trailer Sharing

trailer and proceeds to service a single truck customer on its way back to the depot. Truck 3 has left the depot, bound for its first delivery. After completing all its services it re-couples the trailer left by truck 2 at the parking place, after which it services a single vehicle customer on its way back to the depot.

From figure 9.1 it should be clear that all three trucks make use of the single trailer in the routing plan. If we look at the route the trailer takes, we see that it is first pulled by truck 1, after being parked it is pulled by truck 2, until finally it is returned to the depot by truck 3.

Trailer Exchanges: Let us consider the first part of the sequence of route one assuming truck 1 does not share the trailer.

$$\sigma_1 = \{0_s, 20, 1_s, 2, 3, 1_e, 4, \dots\}$$

Here 1_e symbolizes that the trailer is picked up again at parking place 1.

Let us assume the first part of truck 2's sequence is:

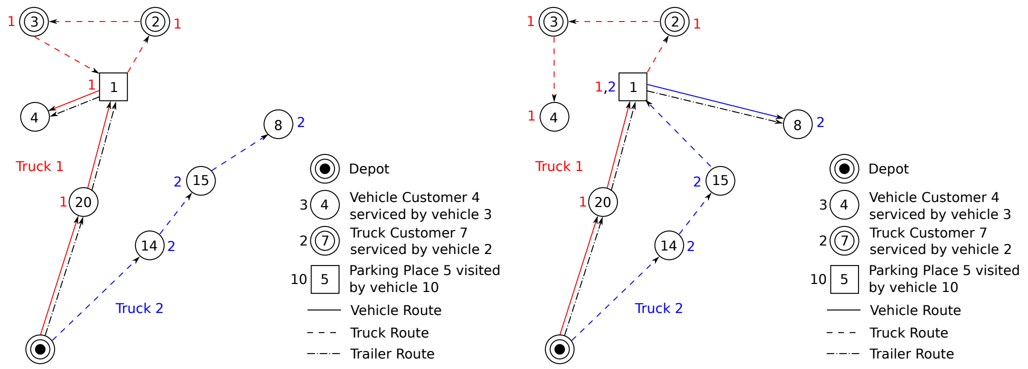
$$\sigma_2 = \{0_s, 14, 15, 8, \dots\}$$

Figure 9.2(a) visualizes the situation.

At this point we choose to relocate parking place 1_e from route one into route two between customers 15 and 8 resulting in the following sequences:

$$\sigma_1 = \{0_s, 20, 1_s, 2, 3, 4, \dots\}$$
$$\sigma_2 = \{0_s, 14, 15, 1_e, 8, \dots\}$$

The updated situation is shown in figure 9.2(b). Thus by applying the `relocate` operator to a parking end node p_e, essentially allows us to share a trailer between



(a) Visualization of trailer sharing part 1 (b) Visualization of trailer sharing part 2

Figure 9.2: Visualization of how trailer sharing may be possible using the relocate neighbourhood operator.

different trucks. Naturally, one has to consider several aspects in order to preserve feasibility:

- The insertion point of a relocated node p_e has to be a subtour edge in which case two scenarios are possible:
  1. The route is a truck route: The feasibility of all remaining customers has to be considered not only with respect to time windows, but also service type i.e. truck customers in the remainder of the route now have to be serviced in a subtour which first has to be determined.
  2. The route is a vehicle route: The feasibility checks remain the same as above, but in addition we now have to further relocate the parking place that terminates the subtour into which the node p_e is inserted. This might be implemented using an ejection chain procedure for example.
- The relocated node p_e should inherit the load information from its corresponding p_s counterpart.
- Both the route to relocate from and the route into which a node p_e is relocated, have to be checked for load feasibility prior to the move.

The above list of conditions is not necessarily complete, but at the very least the listed points have to be considered. All in all, it appears as if it may be possible, however, at a likely very high computational price. Specifically if additional subtours have to be inserted as described above.

9.1.2 Adapting the ETTRP for the TTRP Model by Chao

Model Considerations: We may adapt the ETTRP model to the TTRP model by Chao [Cha02] by rethinking how parking places should be represented in the model. Doing so will not only simplify the model, but will also increase its efficiency in terms of computational complexity in the context of local search etc. First recall the parking rules 1-5 of section 5.1. Since the model of Chao is stricter with respect to the location of parking places as well as the use of a parking place in general (they may only be reused within a routing plan within the same route, and doing so has to happen consecutively), we need to modify the rules to take this into account. Rule 1 needs to be changed to:

Rule 1: For every vehicle customer c we automatically allocate a pair p_s and p_e associated with the customer (same location). The edge connecting c and p_s is considered static, hence no customers may be inserted between c and p_s, nor may the edge be removed. For all practical purposes the vehicle customer c may be interpreted as a segment $\varsigma_c = \{c, p_s, p_e\}$. The segment may grow (as defined next) but may never shrink to become "smaller" than ς_c.

Rules 2-4 remain unchanged, however, rule 5 needs to be changed to:

Rule 5: If at some point a customer is inserted in the subtour originating at c, i.e. between the parking places p_s and p_e associated with c, we immediately allocate a new pair of parking places p'_s and p'_e and insert this pair after p_e. For the sake of simplicity this extra allocated pair of parking places is referred to as an "excess parking place".

Furthermore the following three rules are introduced:

Rule 6: At any given time, if the number of empty excess parking places is greater than one, we immediately remove these excess parking places (leaving only one in the representation).

Rule 7: Relocating a vehicle customer c into another route can be handled in two ways:

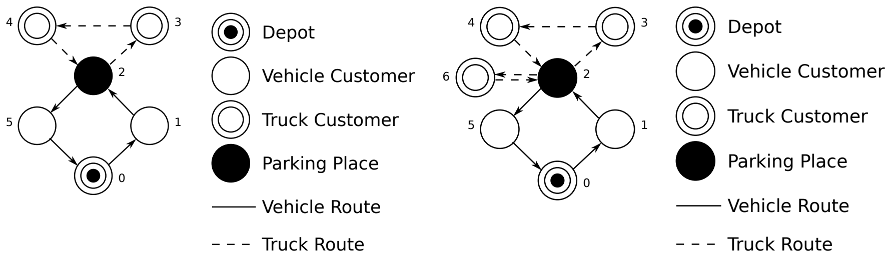
1. Deny relocation if the segment ς_c contains nonempty subtours.

2. Allow relocation of the entire segment ς_c (including subtours).

Rule 8: Upon visiting a parking place p_s or p_e cargo may be relocated freely from truck to trailer or vice versa.

Although rule 6 is not strictly necessary, its purpose is to keep the representation as small as possible. Rule 7 makes sure that if a vehicle customer c is to be relocated, then any parking places associated with it also have to be relocated. Finally rule 8 allows cargo transfers at parking places in general.

To illustrate the representation we consider figure 9.3. Sub figure 9.3(a) shows a vehicle route having a single subtour originating at vehicle customer 2 (which is



(a) Single subtour at customer 2 (b) Multiple subtours at customer 2

Figure 9.3: Example TTRP routing plan

shown as a parking place). The resulting representation of the sequence is easy to derive given the definitions above:

$$\sigma_a = \{0_s, 1, 1_s, 1_e, 2, 2_s, 3, 4, 2_e, 5, 5_s, 5_e, 0_e\}$$

Similarly we may derive the representation of the sequence for the route given in figure 9.3(b):

$$\sigma_b = \{0_s, 1, 1_s, 1_e, 2, 2_s, 3, 4, 2_e, 2_s, 6, 2_e, 2_s, 2_e, 5, 5_s, 5_e, 0_e\}$$

It is clear that the segment $\varsigma_b^{3,4} = \{1_s, 1_e\}$ denoting the empty subtour rooted at customer 1 does not influence the objective function of (3.3.1) since the distance $\underbrace{d(1, 1_s)}_{=0} + \underbrace{d(1_s, 1_e)}_{=0} + d(1_e, 2)$ equals $d(1, 2)$.

Adapting the Construction Heuristic: The construction heuristic has to be modified in order to generate feasible TTRP solutions. Therefore, when assigning truck customers to parking places, we instead assign the truck customers to static segments defined previously, while keeping the total demand, of such segments below that of a complete vehicle. In case a truck customer would violate the demand it must be assigned to a different segment instead.

Applying the definitions above to the ETTRP model, we are capable of emulating the TTRP model of Chao, since:

- Parking places are statically assigned to vehicle customers as in the original model in such a way that empty subtours do not contribute to the objective function.
- An arbitrary number of subtours from every vehicle customer are possible as long as these subtours appear consecutively.
- Cargo may be relocated freely at parking places.
- Subtours cannot appear in truck routes or within other subtours.

Furthermore, the `subtour relocation` operator may be removed entirely since we may simply relocate a truck customer using `relocate` into the static subtours. This should potentially save a lot of computational time. Similarly we avoid evaluating moves that relocate a vehicle customer into its own subtour as we do using the ETTRP model for solving the TTRP, also saving computational time.

The overall speed of the algorithm may further be improved by simply removing redundant functions used for time window feasibility checks as well as updating of temporal variables. Similarly the checks for load constraints, and updating of the associated variables may be greatly simplified.

9.1.3 Modelling Limited Trailer Parking Time

Although no thorough investigation has been conducted in order to determine if the model may be adapted so that the parking time of a trailer may be limited, it is believed that it may be accomplished by defining a stricter time window for the re-coupling node p_e associated with the trailer. Thus if we assume a trailer may be parked for τ time units, then we may associate a time window on the re-coupling node $p_e = [e_{p_s}, (\delta_{p_s} + \tau)]$. Thus if the arrival time at p_s changes, so does the latest re-coupling time of the trailer. Feasibility of changing the arrival time at p_s therefore has to take the time windows of p_e^+ and p_e^- into account.

9.2 More Extensive Testing of the Algorithms

Overall it would have been desirable to conduct a more thorough experimentation on the implemented algorithms. In particular since the running time of the algorithms was rather long both the number of instances on which to test the algorithms but also the allowed running time, had to be limited. Although a number of instances from each class of Solomon problems was converted to corresponding ETTRP instances, it is difficult to know if all aspects of the class of problem were captured in the corresponding ETTRP problem.

9.2.1 Load Constraints

Further experiments which might indicate the impact on the solution quality due to load constraints would have been desirable. Unfortunately there was not enough time available to conduct such additional experiments for this thesis.

9.3 Improvements in Implementation

Although the algorithms of this thesis were implemented in the C++ programming language, and the various procedures were carefully thought through, certain aspects of the code could possibly be reimplemented more efficiently.

9.3.1 Distributed Computing

An obvious extension to the already threaded approach implemented in this thesis is to allow the problem to be solved concurrently in a distributed manner. A server would be responsible for keeping track of the best solution found so far, and upon spawning a task to be executed by the thread pool, we also allow clients to retrieve jobs from this pool of tasks and report back the solution to the server once solved.

9.3.2 Detailed Neighbourhood Structure Analysis

A thorough investigation of the neighbourhood structure would most likely be beneficial in order to possibly improve running times. Analysing the possibility of applying concepts similar to those of a Variable Neighbourhood Search might decrease running times by restricting which neighbourhood operators to apply during the search. A brief attempt was made to gradually add neighbourhood operators as the number of non-improving moves increased (depending on the neighbourhood size), but the attempt did not provide any conclusive results. A more detailed investigation into the concept however, might provide further insights.

Chapter 10

Conclusion

This thesis was motivated by a concrete problem description by the company Transvision A/S [A/S75], in which a classic distribution problem should take advantage of the extra capacity granted by the use of trailers. The problem itself has received little attention in the literature, although the previous works of Chao [Cha02] and Scheuerer [Sch04] regarding the Truck and Trailer Routing Problem (TTRP) served as initial inspiration.

Two additional constraints namely time windows and load constraints were added to the model giving rise to the a new problem definition: the Extended Truck and Trailer Routing Problem (ETTRP). Therefore, the implemented algorithms in this thesis are able to solve vehicle routing problems of various sorts with and without load constraints: The Capacitated Vehicle Routing Problem (CVRP), the Vehicle Routing Problem with Time Windows (VRPTW), the TTRP, and the ETTRP.

The algorithms for constructing the initial solution to the ETTRP were implemented and evaluated for a wide variety of problem instances. The best performing algorithm was chosen based on statistical evidence.

In previous works, the used metaheuristics for solving the TTRP were the Tabu Search (TS) algorithm as well as the Simulated Annealing (SA) algorithm, and because of this it was decided to evaluate the performance of the Attribute Based Hill Climber (ABHC), since it had been shown as highly effective for various other discrete optimization problems. Preliminary experiments showed promising results although the algorithm showed high running times compared to TS and SA.

A variety of techniques were explored in order to speed up the running time of the ABHC:

- Changing the search strategy from Best improve to First improve.
- Caching of moves.
- Granular neighbourhood pruning.

In addition, the deterministic behaviour of the ABHC was considered and changed in order to add stochastic behaviour, resulting in improved performance in terms of obtained solution qualities, and, as before, these results were backed up by statistical evidence.

Overall the implemented algorithms perform well, and in most cases they are able to determine that the use of trailers has a beneficial impact on the solutions obtained, even though the problem involves time windows and load constraints. However, due to the complex nature of the ETTRP the allowed computational time of the algorithms may have to be extended beyond the ten minutes allowed in this thesis. Although it may be concluded that the use of trailers is beneficial, some questions still remain unanswered. Specifically it remains undetermined how big an influence the load constraints have on the obtained solutions, and is thus an area where future work might provide further insights.

The concept of trailer sharing was considered included in this thesis, but due to the additional complexities added, it was not possible to extend the model to include the concept in due time. Trailer sharing thus also remains unexplored and could likewise be considered an interesting point of investigation in future work.

Appendix A

Problemformulering

Algoritmer til Truck and Trailer problemet

Specialeoplæg for Ralph Zitz

Formålet med specialet er at udvikle effektive algoritmer til det såkaldte truck and trailer problem. Problemet er et klassisk distributionsproblem, med et antal kunder, der skal serviceres fra en terminal. Bilerne består af en forvogn og en trailer (hænger) med hver sin kapacitet. Nogle kunder har som restriktion, at hængeren ikke kan komme ind på deres lokation, hvilket betyder, at hængeren må kobles fra. Det er forbundet med en omkostning at koble hængeren til og fra. Der er en ekstra kørselsomkostning ved at køre med hænger. Der er angivet et antal lokationer, hvor det er muligt at stille hængeren. Problemet består i at finde ud af, hvornår og hvor hængeren skal kobles fra/til, således at kørselsomkostningerne minimeres. Allerede i den beskrevne version er problemet meget vanskeligt (NP-hårdt). Problemet, der har stor praktisk relevans og er stillet til os af firmaet Transvision A/S, har ikke været genstand for megen opmærksomhed i litteraturen og oftest kun i versioner hvor trailere og biler allerede er parret og trailere ikke kan byttes rundt mellem forvogne. Samtidig tillades f.eks at traileren efterlades hos kunden, noget som sjældent vil være praktisk muligt i virkelige anvendelser hvor man man forstille sig en række yderligere komplikationer:

1. En hænger, der er efterladt kan samles op af andre forvogne
2. Der er læsseregler for, hvor meget der må være på hængeren i forholdt til hvad der er på forvognen. Det kunne f.eks. være at traileren ikke må veje mere end forvognen.
3. En hænger må kun være efterladt et vist tidsrum
4. Flere terminaler, hvor bilerne starter fra. Hængerne skal tilbage til deres startterminal (se 1)
5. Tidsvinduer på kunderne, hastigheden bilen kører med afhænger om den har hænger på eller ej.

Specialet skal indeholde et overblik over state of the art indenfor algoritmer til problemet, samt implementationer af en række egne algoritmer som dels skal kunne løse den simple version uden de mange ekstra mulige komplikationer, dels kan håndtere udvalgte af disse (med henblik på at komme tættere på praktisk anvendelighed).

Appendix B

Improved Solution for TTRP instance p02

```
# Instance: p02, Distance: 610.267, Total Demand: 777
# Route: 1, Demand: 197, Route Length: 149.644
0 60 18 14 25 60 54 17 44 42 19 40 41 13 54 4 0
# Route: 2, Demand: 191, Route Length: 149.105
0 5 55 49 10 39 33 45 15 37 55 70 9 30 34 50 38 70 0
# Route: 3, Demand: 198, Route Length: 143.178
0 51 47 12 51 27 76 23 24 43 7 26 31 8 76 48 6 0
# Route: 4, Demand: 95, Route Length: 70.1522
0 32 2 29 21 16 11 46 0
# Route: 5, Demand: 96, Route Length: 98.1871
0 20 35 36 3 28 22 1 0
```

Appendix C

Race Output

C.1 Parameter Selection for Construction Heuristics

```
R version 2.10.1 (2009-12-14)
Copyright (C) 2009 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> library("race")
> class <- "/home/ralph/Speciale/experiments/all-instances";
> O<-race("wrapper-construction.R",
          maxExp=10000,
          stat.test=c("friedman"),
          conf.level=0.95,
          first.test=5,
          interactive=TRUE,
          log.file="raceConstruction.log",
          no.slaves=0)

Racing methods for the selection of the best
Copyright (C) 2003 Mauro Birattari
This software comes with ABSOLUTELY NO WARRANTY

Race name.................................Model Selection:
Number of candidates.....................................630
Number of available tasks................................756
Max number of experiments..............................10000
Statistical test...............................Friedman test
Tasks seen before discarding...............................5
Initialization function...................................ok
Parallel Virtual Machine..................................no

                          Markers:
                             x No test is performed.
                             - The test is performed and
                               some candidates are discarded.
                             = The test is performed but
                               no candidate is discarded.

+-+-----------+-----------+-----------+-----------+-----------+
| |       Task|      Alive|       Best|  Mean best| Exp so far|
+-+-----------+-----------+-----------+-----------+-----------+
|x|          1|        630|        194|       2091|        630|
|x|          2|        630|        194|       1636|       1260|
|x|          3|        630|        383|       1837|       1890|
|x|          4|        630|        383|       1933|       2520|
```

−	5	145	383	1965	3150
=	6	145	383	1960	3295
=	7	145	383	1979	3440
=	8	145	383	2029	3585
−	9	84	572	2105	3730
=	10	84	572	2087	3814
=	11	84	598	2097	3898
=	12	84	569	2023	3982
=	13	84	569	1939	4066
−	14	67	569	1946	4150
=	15	67	569	1966	4217
−	16	38	569	1970	4284
=	17	38	569	1969	4322
=	18	38	569	1985	4360
=	19	38	569	1991	4398
=	20	38	569	1996	4436
=	21	38	569	2012	4474
=	22	38	569	2014	4512
=	23	38	569	1996	4550
=	24	38	569	1996	4588
=	25	38	569	2008	4626
−	26	26	569	1970	4664
=	27	26	569	1977	4690
=	28	26	569	1956	4716
=	29	26	569	1963	4742
=	30	26	569	1976	4768
=	31	26	569	1961	4794
=	32	26	569	1974	4820
=	33	26	569	1970	4846
=	34	26	569	1987	4872
=	35	26	569	1976	4898
=	36	26	569	1983	4924
=	37	26	569	1986	4950
=	38	26	569	2000	4976
=	39	26	569	2000	5002
−	40	14	569	2005	5028
=	41	14	569	2017	5042
=	42	14	569	2022	5056
=	43	14	569	2018	5070
=	44	14	569	2020	5084
=	45	14	569	2020	5098
=	46	14	569	2019	5112
=	47	14	569	2020	5126
=	48	14	569	2010	5140
=	49	14	569	2014	5154
=	50	14	569	2022	5168
=	51	14	569	2023	5182
=	52	14	569	2025	5196
−	53	6	569	2027	5210
=	54	6	569	2024	5216
=	55	6	569	2017	5222
=	56	6	569	2034	5228
=	57	6	506	2030	5234
=	58	6	506	2020	5240
=	59	6	506	2019	5246
=	60	6	506	2019	5252
=	61	6	506	2024	5258
=	62	6	506	2029	5264
=	63	6	506	2013	5270
=	64	6	506	2011	5276
=	65	6	506	2003	5282
=	66	6	506	2001	5288
=	67	6	506	1995	5294
=	68	6	506	1997	5300
=	69	6	506	2000	5306
=	70	6	506	1994	5312
=	71	6	506	1997	5318
=	72	6	506	1990	5324
=	73	6	569	1990	5330
=	74	6	569	1987	5336
=	75	6	569	1981	5342
=	76	6	569	1982	5348
=	77	6	569	1983	5354
=	78	6	569	1981	5360
=	79	6	569	1982	5366
−	80	5	569	1985	5372
=	81	5	569	1987	5377
=	82	5	569	1993	5382
=	83	5	569	1994	5387
=	84	5	569	1994	5392
=	85	5	569	2001	5397
=	86	5	569	2002	5402
=	87	5	569	1998	5407
=	88	5	569	1998	5412
=	89	5	569	1998	5417
=	90	5	569	1996	5422
=	91	5	569	1998	5427

=	92	5	569	1994	5432
=	93	5	569	1996	5437
=	94	5	569	1999	5442
=	95	5	569	2001	5447
=	96	5	569	2001	5452
=	97	5	569	2005	5457
=	98	5	569	2010	5462
=	99	5	569	2014	5467
=	100	5	569	2016	5472
=	101	5	569	2017	5477
=	102	5	569	2019	5482
=	103	5	569	2018	5487
=	104	5	569	2015	5492
=	105	5	569	2017	5497
=	106	5	569	2019	5502
=	107	5	569	2016	5507
=	108	5	569	2016	5512
=	109	5	569	2015	5517
=	110	5	569	2016	5522
=	111	5	569	2016	5527
=	112	5	569	2019	5532
=	113	5	569	2019	5537
=	114	5	569	2020	5542
=	115	5	569	2021	5547
=	116	5	569	2024	5552
=	117	5	569	2019	5557
=	118	5	569	2017	5562
=	119	5	569	2013	5567
=	120	5	569	2016	5572
=	121	5	569	2016	5577
=	122	5	569	2018	5582
=	123	5	569	2017	5587
=	124	5	569	2015	5592
=	125	5	569	2010	5597
=	126	5	569	2010	5602
=	127	5	569	2010	5607
=	128	5	569	2009	5612
=	129	5	569	2001	5617
=	130	5	569	1998	5622
=	131	5	569	1998	5627
=	132	5	569	2001	5632
=	133	5	569	2001	5637
=	134	5	569	2000	5642
=	135	5	569	2000	5647
=	136	5	569	2003	5652
=	137	5	569	2002	5657
=	138	5	569	2005	5662
=	139	5	569	2006	5667
=	140	5	569	2014	5672
=	141	5	569	2013	5677
=	142	5	569	2015	5682
=	143	5	569	2015	5687
=	144	5	569	2019	5692
=	145	5	569	2019	5697
=	146	5	569	2018	5702
=	147	5	569	2019	5707
=	148	5	569	2021	5712
=	149	5	569	2020	5717
=	150	5	569	2017	5722
=	151	5	569	2019	5727
=	152	5	569	2019	5732
=	153	5	569	2019	5737
=	154	5	569	2012	5742
=	155	5	569	2013	5747
=	156	5	569	2015	5752
=	157	5	569	2008	5757
=	158	5	569	2007	5762
=	159	5	569	2007	5767
=	160	5	569	2004	5772
=	161	5	569	1998	5777
=	162	5	569	1995	5782
=	163	5	569	1997	5787
=	164	5	569	1998	5792
=	165	5	569	1998	5797
=	166	5	569	1999	5802
=	167	5	569	1996	5807
=	168	5	569	1997	5812
=	169	5	569	1992	5817
=	170	5	569	1993	5822
=	171	5	569	1995	5827
=	172	5	569	1994	5832
=	173	5	569	1995	5837
=	174	5	569	1995	5842
=	175	5	569	1997	5847
=	176	5	569	2000	5852
=	177	5	569	2001	5857
=	178	5	569	2001	5862

```
|=|       179|         5|       569|      2002|      5867|
|=|       180|         5|       569|      2003|      5872|
|=|       181|         5|       569|      2005|      5877|
|=|       182|         5|       569|      2003|      5882|
|=|       183|         5|       569|      2005|      5887|
|=|       184|         5|       569|      2006|      5892|
|=|       185|         5|       569|      2007|      5897|
|=|       186|         5|       569|      2007|      5902|
|-|       187|         3|       569|      2009|      5907|
|-|       188|         2|       569|      2009|      5910|
|-|       189|         1|       569|      2011|      5912|
+-+----------+----------+----------+----------+----------+

Selected candidate:        569 mean value:      2011

Description of the selected candidate:
     label path
569  CONS200
                                                                command
569 --fill-ratio 100 --paramfiles /home/ralph/Speciale/settings/i1-2.cfg


>
```

C.2 Subtour Optimization for Construction Heuristics

```
R version 2.10.1 (2009-12-14)
Copyright (C) 2009 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> library("race")
> class <- "all-instances";
> O <- race("wrapper-suboptimizer.R",
           maxExp=5000,
           stat.test=c("friedman"),
           conf.level=0.95,
           first.test=5,
           interactive=TRUE,
           log.file="raceSuboptimizer.log",
           no.slaves=0)

Racing methods for the selection of the best
Copyright (C) 2003 Mauro Birattari
This software comes with ABSOLUTELY NO WARRANTY

Race name.................................Model Selection:
Number of candidates...................................220
Number of available tasks..............................756
Max number of experiments.............................5000
Statistical test...............................Friedman test
Tasks seen before discarding.............................5
Initialization function.................................ok
Parallel Virtual Machine................................no

                         Markers:
                            x No test is performed.
                            - The test is performed and
                              some candidates are discarded.
                            = The test is performed but
                              no candidate is discarded.


+-+----------+----------+----------+----------+----------+
| |      Task|     Alive|      Best| Mean best|Exp so far|
+-+----------+----------+----------+----------+----------+
|x|         1|       220|        49|      2163|       220|
|x|         2|       220|        49|      1509|       440|
|x|         3|       220|        49|      1713|       660|
|x|         4|       220|       137|      1882|       880|
|-|         5|        66|        27|      1937|      1100|
```

=	6	66	27	1932	1166
=	7	66	68	1997	1232
=	8	66	93	2053	1298
=	9	66	200	2095	1364
=	10	66	199	2076	1430
=	11	66	199	2078	1496
=	12	66	199	2029	1562
=	13	66	93	1953	1628
=	14	66	199	1945	1694
=	15	66	177	1984	1760
=	16	66	177	1973	1826
−	17	50	177	1976	1892
=	18	50	177	1989	1942
=	19	50	178	1999	1992
=	20	50	178	2009	2042
=	21	50	178	2019	2092
=	22	50	178	2028	2142
=	23	50	199	2003	2192
=	24	50	199	1995	2242
=	25	50	177	2014	2292
=	26	50	200	1964	2342
=	27	50	177	1985	2392
=	28	50	177	1961	2442
=	29	50	177	1971	2492
=	30	50	177	1981	2542
=	31	50	177	1964	2592
=	32	50	200	1956	2642
=	33	50	177	1963	2692
−	34	38	177	1981	2742
=	35	38	177	1967	2780
=	36	38	177	1972	2818
=	37	38	177	1976	2856
=	38	38	177	1990	2894
=	39	38	177	1987	2932
=	40	38	177	1996	2970
−	41	20	177	2009	3008
=	42	20	199	2017	3028
=	43	20	199	2009	3048
=	44	20	199	2007	3068
=	45	20	199	2006	3088
=	46	20	199	2007	3108
−	47	16	199	2006	3128
=	48	16	199	2000	3144
=	49	16	199	2003	3160
=	50	16	199	2013	3176
=	51	16	199	2016	3192
=	52	16	199	2020	3208
=	53	16	199	2019	3224
=	54	16	199	2014	3240
=	55	16	199	2022	3256
=	56	16	199	2046	3272
=	57	16	199	2039	3288
=	58	16	199	2029	3304
=	59	16	200	2025	3320
=	60	16	199	2025	3336
=	61	16	199	2030	3352
=	62	16	199	2033	3368
=	63	16	199	2018	3384
=	64	16	200	2011	3400
=	65	16	200	2001	3416
=	66	16	200	1998	3432
=	67	16	200	2004	3448
=	68	16	200	2007	3464
=	69	16	200	2011	3480
=	70	16	200	2004	3496
=	71	16	199	2006	3512
=	72	16	200	1997	3528
=	73	16	200	1998	3544
=	74	16	199	1994	3560
=	75	16	200	1986	3576
=	76	16	200	1988	3592
=	77	16	200	1989	3608
=	78	16	200	1988	3624
=	79	16	200	1990	3640
=	80	16	200	1993	3656
=	81	16	200	1996	3672
=	82	16	200	2003	3688
=	83	16	200	2004	3704
=	84	16	200	2006	3720
=	85	16	199	2016	3736
=	86	16	200	2014	3752
=	87	16	199	2010	3768
=	88	16	199	2008	3784
=	89	16	199	2008	3800
=	90	16	199	2010	3816
=	91	16	199	2009	3832
=	92	16	199	2013	3848

=	93	16	199	2020	3864
=	94	16	199	2022	3880
=	95	16	199	2023	3896
=	96	16	199	2023	3912
=	97	16	199	2027	3928
=	98	16	199	2033	3944
=	99	16	199	2036	3960
=	100	16	178	2040	3976
=	101	16	178	2041	3992
=	102	16	178	2041	4008
=	103	16	178	2038	4024
=	104	16	178	2042	4040
=	105	16	178	2045	4056
=	106	16	178	2045	4072
=	107	16	200	2040	4088
=	108	16	178	2040	4104
=	109	16	178	2038	4120
=	110	16	178	2038	4136
-	111	14	178	2039	4152
=	112	14	178	2041	4166
=	113	14	178	2040	4180
=	114	14	200	2042	4194
=	115	14	200	2044	4208
=	116	14	200	2047	4222
=	117	14	200	2041	4236
=	118	14	178	2039	4250
=	119	14	178	2034	4264
=	120	14	178	2037	4278
=	121	14	178	2038	4292
=	122	14	178	2039	4306
=	123	14	178	2037	4320
=	124	14	178	2035	4334
=	125	14	178	2031	4348
=	126	14	178	2030	4362
=	127	14	178	2030	4376
=	128	14	178	2028	4390
=	129	14	178	2020	4404
=	130	14	178	2016	4418
=	131	14	178	2015	4432
=	132	14	178	2018	4446
=	133	14	178	2019	4460
=	134	14	178	2018	4474
=	135	14	178	2019	4488
=	136	14	178	2021	4502
=	137	14	178	2020	4516
=	138	14	178	2021	4530
=	139	14	178	2022	4544
=	140	14	178	2033	4558
=	141	14	178	2031	4572
=	142	14	178	2031	4586
=	143	14	178	2031	4600
=	144	14	178	2035	4614
=	145	14	178	2038	4628
=	146	14	178	2036	4642
=	147	14	178	2038	4656
=	148	14	178	2043	4670
=	149	14	178	2041	4684
=	150	14	178	2037	4698
=	151	14	178	2039	4712
=	152	14	178	2039	4726
=	153	14	178	2039	4740
=	154	14	178	2032	4754
=	155	14	178	2033	4768
=	156	14	178	2035	4782
=	157	14	178	2028	4796
=	158	14	178	2027	4810
=	159	14	178	2027	4824
=	160	14	178	2026	4838
=	161	14	178	2019	4852
=	162	14	178	2021	4866
=	163	14	178	2022	4880
=	164	14	178	2023	4894
=	165	14	178	2023	4908
=	166	14	178	2025	4922
=	167	14	178	2021	4936
=	168	14	178	2029	4950
=	169	14	178	2023	4964
-	170	10	178	2023	4978
=	171	10	178	2024	4988
=	172	10	178	2024	4998

Selected candidate: 178 mean value: 2024

Description of the selected candidate:
label path
178 CONS178

```
                                                             command
178 --fill-ratio 90 --paramfiles /home/ralph/Speciale/settings/i1-2.cfg

>
```

C.3 Final Selection of Construction Heuristic

```
R version 2.10.1 (2009-12-14)
Copyright (C) 2009 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> library("race")
> class <- "/home/ralph/Speciale/experiments/all-instances";
> O<-race("wrapper-final_construction.R",
         maxExp=5000,
         stat.test=c("friedman"),
         conf.level=0.95,
         first.test=5,
         interactive=TRUE,
         log.file="raceFinalConstruction.log",
         no.slaves=0)

Racing methods for the selection of the best
Copyright (C) 2003 Mauro Birattari
This software comes with ABSOLUTELY NO WARRANTY

Race name.....................................Model Selection:
Number of candidates..........................................2
Number of available tasks...................................756
Max number of experiments..................................5000
Statistical test.................................Friedman test
Tasks seen before discarding..................................5
Initialization function......................................ok
Parallel Virtual Machine.....................................no

                              Markers:
                                 x No test is performed.
                                 - The test is performed and
                                   some candidates are discarded.
                                 = The test is performed but
                                   no candidate is discarded.

+-+-----------+-----------+-----------+-----------+-----------+
| |       Task|      Alive|       Best|  Mean best| Exp so far|
+-+-----------+-----------+-----------+-----------+-----------+
|x|          1|          2|          1|       2373|          2|
|x|          2|          2|          1|       1701|          4|
|x|          3|          2|          1|       1848|          6|
|x|          4|          2|          1|       1956|          8|
|=|          5|          2|          1|       2040|         10|
|=|          6|          2|          2|       1957|         12|
|=|          7|          2|          1|       2025|         14|
|=|          8|          2|          1|       2030|         16|
|=|          9|          2|          1|       2075|         18|
|=|         10|          2|          1|       2068|         20|
|=|         11|          2|          1|       2061|         22|
|=|         12|          2|          1|       2023|         24|
|=|         13|          2|          1|       1939|         26|
|=|         14|          2|          1|       1946|         28|
|=|         15|          2|          1|       1966|         30|
|=|         16|          2|          2|       1978|         32|
|=|         17|          2|          2|       1975|         34|
|=|         18|          2|          2|       1988|         36|
|=|         19|          2|          2|       1999|         38|
|=|         20|          2|          1|       1996|         40|
|=|         21|          2|          2|       2019|         42|
|=|         22|          2|          1|       2014|         44|
|=|         23|          2|          1|       1996|         46|
```

```
|=|    24|    2|    2|    2003|    48|
|=|    25|    2|    2|    2012|    50|
|=|    26|    2|    2|    1974|    52|
|=|    27|    2|    2|    1984|    54|
|=|    28|    2|    2|    1960|    56|
|=|    29|    2|    1|    1963|    58|
|=|    30|    2|    2|    1979|    60|
|=|    31|    2|    2|    1959|    62|
|=|    32|    2|    2|    1964|    64|
|=|    33|    2|    2|    1959|    66|
|=|    34|    2|    2|    1978|    68|
|=|    35|    2|    2|    1966|    70|
|=|    36|    2|    2|    1971|    72|
|=|    37|    2|    2|    1982|    74|
|=|    38|    2|    2|    1994|    76|
|=|    39|    2|    2|    1994|    78|
|=|    40|    2|    2|    2002|    80|
|=|    41|    2|    2|    2015|    82|
|=|    42|    2|    2|    2016|    84|
|=|    43|    2|    2|    2012|    86|
|=|    44|    2|    2|    2014|    88|
|=|    45|    2|    2|    2017|    90|
|=|    46|    2|    2|    2016|    92|
|=|    47|    2|    2|    2016|    94|
|=|    48|    2|    2|    2007|    96|
|=|    49|    2|    2|    2010|    98|
|=|    50|    2|    2|    2020|   100|
|=|    51|    2|    2|    2021|   102|
|=|    52|    2|    2|    2019|   104|
|=|    53|    2|    2|    2021|   106|
|=|    54|    2|    2|    2018|   108|
|=|    55|    2|    2|    2025|   110|
|=|    56|    2|    2|    2053|   112|
|=|    57|    2|    2|    2045|   114|
|=|    58|    2|    2|    2035|   116|
|=|    59|    2|    2|    2033|   118|
|=|    60|    2|    2|    2031|   120|
|=|    61|    2|    2|    2039|   122|
|=|    62|    2|    2|    2042|   124|
|=|    63|    2|    2|    2026|   126|
|=|    64|    2|    2|    2022|   128|
|=|    65|    2|    2|    2012|   130|
|=|    66|    2|    2|    2009|   132|
|=|    67|    2|    2|    2015|   134|
|=|    68|    2|    2|    2018|   136|
|=|    69|    2|    2|    2020|   138|
|=|    70|    2|    2|    2011|   140|
|=|    71|    2|    2|    2013|   142|
|=|    72|    2|    2|    2004|   144|
|=|    73|    2|    2|    2007|   146|
|=|    74|    2|    2|    2002|   148|
|=|    75|    2|    2|    1995|   150|
|=|    76|    2|    2|    1994|   152|
|=|    77|    2|    2|    1995|   154|
|=|    78|    2|    2|    1998|   156|
|=|    79|    2|    2|    2000|   158|
|=|    80|    2|    2|    2002|   160|
|=|    81|    2|    2|    2006|   162|
|=|    82|    2|    2|    2012|   164|
|=|    83|    2|    2|    2013|   166|
|=|    84|    2|    2|    2014|   168|
|=|    85|    2|    2|    2022|   170|
|=|    86|    2|    2|    2022|   172|
|=|    87|    2|    2|    2017|   174|
|=|    88|    2|    2|    2015|   176|
|=|    89|    2|    2|    2014|   178|
|=|    90|    2|    2|    2015|   180|
|=|    91|    2|    2|    2013|   182|
|=|    92|    2|    2|    2017|   184|
|=|    93|    2|    2|    2024|   186|
|=|    94|    2|    2|    2025|   188|
|=|    95|    2|    2|    2026|   190|
|=|    96|    2|    2|    2026|   192|
|=|    97|    2|    2|    2030|   194|
|=|    98|    2|    2|    2035|   196|
|=|    99|    2|    2|    2038|   198|
|=|   100|    2|    2|    2040|   200|
|=|   101|    2|    2|    2041|   202|
|=|   102|    2|    2|    2041|   204|
|=|   103|    2|    2|    2038|   206|
|=|   104|    2|    2|    2042|   208|
|=|   105|    2|    2|    2045|   210|
|=|   106|    2|    2|    2045|   212|
|=|   107|    2|    2|    2043|   214|
|=|   108|    2|    2|    2040|   216|
|=|   109|    2|    2|    2038|   218|
|=|   110|    2|    2|    2038|   220|
```

=	111	2	2	2039	222
=	112	2	2	2041	224
=	113	2	2	2040	226
=	114	2	2	2044	228
=	115	2	2	2046	230
=	116	2	2	2049	232
=	117	2	2	2042	234
=	118	2	2	2039	236
=	119	2	2	2034	238
=	120	2	2	2037	240
=	121	2	2	2038	242
=	122	2	2	2039	244
=	123	2	2	2037	246
=	124	2	2	2035	248
=	125	2	2	2031	250
=	126	2	2	2030	252
=	127	2	2	2030	254
=	128	2	2	2028	256
=	129	2	2	2020	258
=	130	2	2	2016	260
=	131	2	2	2015	262
=	132	2	2	2018	264
=	133	2	2	2019	266
=	134	2	2	2018	268
=	135	2	2	2019	270
=	136	2	2	2021	272
=	137	2	2	2020	274
=	138	2	2	2021	276
=	139	2	2	2022	278
=	140	2	2	2033	280
=	141	2	2	2031	282
=	142	2	2	2031	284
=	143	2	2	2031	286
=	144	2	2	2035	288
=	145	2	2	2038	290
=	146	2	2	2036	292
=	147	2	2	2038	294
=	148	2	2	2043	296
=	149	2	2	2041	298
=	150	2	2	2037	300
=	151	2	2	2039	302
=	152	2	2	2039	304
=	153	2	2	2039	306
=	154	2	2	2032	308
=	155	2	2	2033	310
=	156	2	2	2035	312
=	157	2	2	2028	314
=	158	2	2	2027	316
=	159	2	2	2027	318
=	160	2	2	2026	320
=	161	2	2	2019	322
=	162	2	2	2021	324
=	163	2	2	2022	326
=	164	2	2	2023	328
=	165	2	2	2023	330
=	166	2	2	2025	332
=	167	2	2	2021	334
=	168	2	2	2029	336
=	169	2	2	2023	338
=	170	2	2	2023	340
=	171	2	2	2024	342
=	172	2	2	2024	344
=	173	2	2	2024	346
=	174	2	2	2024	348
=	175	2	2	2026	350
=	176	2	2	2029	352
=	177	2	2	2030	354
=	178	2	2	2030	356
=	179	2	2	2032	358
=	180	2	2	2033	360
=	181	2	2	2036	362
=	182	2	1	2003	364
=	183	2	2	2039	366
=	184	2	2	2039	368
=	185	2	2	2040	370
=	186	2	2	2041	372
=	187	2	2	2041	374
=	188	2	2	2042	376
=	189	2	2	2043	378
=	190	2	2	2048	380
=	191	2	1	2017	382
=	192	2	1	2017	384
=	193	2	1	2014	386
=	194	2	1	2012	388
=	195	2	1	2012	390
=	196	2	1	2010	392
=	197	2	1	2011	394

=	198	2	1	2008	396
=	199	2	1	2014	398
=	200	2	1	2012	400
=	201	2	1	2011	402
=	202	2	1	2009	404
=	203	2	1	2004	406
=	204	2	1	2004	408
=	205	2	1	2008	410
=	206	2	1	2009	412
=	207	2	1	2011	414
=	208	2	1	2010	416
=	209	2	1	2009	418
=	210	2	1	2007	420
=	211	2	1	2008	422
=	212	2	1	2008	424
=	213	2	1	2006	426
=	214	2	1	2007	428
=	215	2	1	2007	430
=	216	2	1	2006	432
=	217	2	1	2005	434
=	218	2	1	2007	436
=	219	2	1	2007	438
=	220	2	1	2010	440
=	221	2	1	2005	442
=	222	2	1	2007	444
=	223	2	1	2008	446
=	224	2	1	2008	448
=	225	2	1	2008	450
=	226	2	1	2009	452
=	227	2	1	2010	454
=	228	2	1	2008	456
=	229	2	1	2008	458
=	230	2	1	2005	460
=	231	2	1	2004	462
=	232	2	1	2005	464
=	233	2	1	2003	466
=	234	2	1	2002	468
=	235	2	1	2003	470
=	236	2	1	2003	472
=	237	2	1	2005	474
=	238	2	1	2007	476
=	239	2	1	2008	478
=	240	2	1	2005	480
=	241	2	1	2003	482
=	242	2	1	2005	484
=	243	2	1	2007	486
=	244	2	1	2007	488
=	245	2	1	2007	490
=	246	2	1	2005	492
=	247	2	1	2005	494
=	248	2	1	2007	496
=	249	2	1	2006	498
=	250	2	1	2006	500
=	251	2	1	2006	502
=	252	2	1	2007	504
=	253	2	1	2007	506
=	254	2	1	2008	508
=	255	2	1	2010	510
=	256	2	1	2014	512
=	257	2	1	2016	514
=	258	2	1	2011	516
=	259	2	1	2010	518
=	260	2	1	2006	520
=	261	2	1	2005	522
=	262	2	1	2005	524
=	263	2	1	2004	526
=	264	2	1	2002	528
=	265	2	1	2002	530
=	266	2	1	2004	532
=	267	2	1	2004	534
=	268	2	1	2006	536
=	269	2	1	2007	538
=	270	2	1	2009	540
=	271	2	1	2009	542
=	272	2	1	2009	544
=	273	2	1	2010	546
=	274	2	1	2007	548
=	275	2	1	2007	550
=	276	2	1	2009	552
=	277	2	1	2010	554
=	278	2	1	2010	556
=	279	2	1	2011	558
=	280	2	1	2012	560
=	281	2	1	2012	562
=	282	2	1	2011	564
=	283	2	1	2009	566
=	284	2	1	2010	568

=	285	2	1	2011	570
=	286	2	1	2012	572
=	287	2	1	2013	574
=	288	2	1	2011	576
=	289	2	1	2011	578
=	290	2	1	2010	580
=	291	2	1	2014	582
=	292	2	1	2016	584
=	293	2	1	2014	586
=	294	2	1	2014	588
=	295	2	1	2014	590
=	296	2	1	2014	592
=	297	2	1	2013	594
=	298	2	1	2014	596
=	299	2	1	2014	598
=	300	2	1	2014	600
=	301	2	1	2015	602
=	302	2	1	2015	604
=	303	2	1	2015	606
=	304	2	1	2013	608
=	305	2	1	2012	610
=	306	2	1	2013	612
=	307	2	1	2015	614
=	308	2	1	2015	616
=	309	2	1	2014	618
=	310	2	1	2014	620
=	311	2	1	2015	622
=	312	2	1	2013	624
=	313	2	1	2015	626
=	314	2	1	2016	628
=	315	2	1	2016	630
=	316	2	1	2013	632
=	317	2	1	2011	634
=	318	2	1	2012	636
=	319	2	1	2013	638
=	320	2	1	2013	640
=	321	2	1	2013	642
=	322	2	1	2015	644
=	323	2	1	2016	646
=	324	2	1	2015	648
=	325	2	1	2015	650
=	326	2	1	2016	652
=	327	2	1	2016	654
=	328	2	1	2017	656
=	329	2	1	2018	658
=	330	2	1	2015	660
=	331	2	1	2016	662
=	332	2	1	2017	664
=	333	2	1	2014	666
=	334	2	1	2015	668
=	335	2	1	2016	670
=	336	2	1	2017	672
=	337	2	1	2016	674
=	338	2	1	2016	676
=	339	2	1	2018	678
=	340	2	1	2018	680
=	341	2	1	2018	682
=	342	2	1	2019	684
=	343	2	1	2019	686
=	344	2	1	2019	688
=	345	2	1	2018	690
=	346	2	1	2019	692
=	347	2	1	2016	694
=	348	2	1	2013	696
=	349	2	1	2014	698
=	350	2	1	2014	700
=	351	2	1	2013	702
=	352	2	1	2010	704
=	353	2	1	2010	706
=	354	2	1	2009	708
=	355	2	1	2011	710
=	356	2	1	2011	712
=	357	2	1	2008	714
=	358	2	1	2009	716
=	359	2	1	2009	718
=	360	2	1	2010	720
=	361	2	1	2008	722
=	362	2	1	2009	724
=	363	2	1	2010	726
=	364	2	1	2010	728
=	365	2	1	2010	730
=	366	2	1	2011	732
=	367	2	1	2011	734
=	368	2	1	2011	736
=	369	2	1	2013	738
=	370	2	1	2013	740
=	371	2	1	2012	742

```
|=|   372|   2|   1|   2012|   744|
|=|   373|   2|   1|   2011|   746|
|=|   374|   2|   1|   2013|   748|
|=|   375|   2|   1|   2013|   750|
|=|   376|   2|   1|   2013|   752|
|=|   377|   2|   1|   2014|   754|
|=|   378|   2|   1|   2015|   756|
|=|   379|   2|   1|   2015|   758|
|=|   380|   2|   1|   2014|   760|
|=|   381|   2|   1|   2014|   762|
|=|   382|   2|   1|   2015|   764|
|=|   383|   2|   1|   2014|   766|
|=|   384|   2|   1|   2012|   768|
|=|   385|   2|   1|   2012|   770|
|=|   386|   2|   1|   2012|   772|
|=|   387|   2|   1|   2013|   774|
|=|   388|   2|   1|   2013|   776|
|=|   389|   2|   1|   2012|   778|
|=|   390|   2|   1|   2012|   780|
|=|   391|   2|   1|   2013|   782|
|=|   392|   2|   1|   2014|   784|
|=|   393|   2|   1|   2014|   786|
|=|   394|   2|   1|   2014|   788|
|=|   395|   2|   1|   2012|   790|
|=|   396|   2|   1|   2011|   792|
|=|   397|   2|   1|   2008|   794|
|=|   398|   2|   1|   2008|   796|
|=|   399|   2|   1|   2005|   798|
|=|   400|   2|   1|   2006|   800|
|=|   401|   2|   1|   2006|   802|
|=|   402|   2|   1|   2006|   804|
|=|   403|   2|   1|   2007|   806|
|=|   404|   2|   1|   2007|   808|
|=|   405|   2|   1|   2007|   810|
|=|   406|   2|   1|   2007|   812|
|=|   407|   2|   1|   2007|   814|
|=|   408|   2|   1|   2005|   816|
|=|   409|   2|   1|   2005|   818|
|=|   410|   2|   1|   2004|   820|
|=|   411|   2|   1|   2005|   822|
|=|   412|   2|   1|   2005|   824|
|=|   413|   2|   1|   2006|   826|
|=|   414|   2|   1|   2006|   828|
|=|   415|   2|   1|   2006|   830|
|=|   416|   2|   1|   2005|   832|
|=|   417|   2|   1|   2005|   834|
|=|   418|   2|   1|   2004|   836|
|=|   419|   2|   1|   2006|   838|
|=|   420|   2|   1|   2006|   840|
|=|   421|   2|   1|   2007|   842|
|=|   422|   2|   1|   2006|   844|
|=|   423|   2|   1|   2006|   846|
|=|   424|   2|   1|   2007|   848|
|=|   425|   2|   1|   2007|   850|
|=|   426|   2|   1|   2007|   852|
|=|   427|   2|   1|   2007|   854|
|=|   428|   2|   1|   2006|   856|
|=|   429|   2|   1|   2008|   858|
|=|   430|   2|   1|   2008|   860|
|=|   431|   2|   1|   2009|   862|
|=|   432|   2|   1|   2010|   864|
|=|   433|   2|   1|   2010|   866|
|=|   434|   2|   1|   2011|   868|
|=|   435|   2|   1|   2010|   870|
|=|   436|   2|   1|   2010|   872|
|=|   437|   2|   1|   2010|   874|
|=|   438|   2|   1|   2009|   876|
|=|   439|   2|   1|   2009|   878|
|=|   440|   2|   1|   2010|   880|
|=|   441|   2|   1|   2011|   882|
|=|   442|   2|   1|   2011|   884|
|=|   443|   2|   1|   2011|   886|
|=|   444|   2|   1|   2011|   888|
|=|   445|   2|   1|   2009|   890|
|=|   446|   2|   1|   2009|   892|
|=|   447|   2|   1|   2009|   894|
|=|   448|   2|   1|   2010|   896|
|=|   449|   2|   1|   2010|   898|
|=|   450|   2|   1|   2010|   900|
|=|   451|   2|   1|   2010|   902|
|=|   452|   2|   1|   2010|   904|
|=|   453|   2|   1|   2009|   906|
|=|   454|   2|   1|   2007|   908|
|=|   455|   2|   1|   2007|   910|
|=|   456|   2|   1|   2007|   912|
|=|   457|   2|   1|   2008|   914|
|=|   458|   2|   1|   2007|   916|
```

```
|=|       459|       2|       1|    2008|     918|
|=|       460|       2|       1|    2008|     920|
|=|       461|       2|       1|    2008|     922|
|=|       462|       2|       1|    2009|     924|
|=|       463|       2|       1|    2008|     926|
|=|       464|       2|       1|    2007|     928|
|=|       465|       2|       1|    2005|     930|
|=|       466|       2|       1|    2005|     932|
|=|       467|       2|       1|    2005|     934|
|=|       468|       2|       1|    2004|     936|
|=|       469|       2|       1|    2004|     938|
|=|       470|       2|       1|    2003|     940|
|=|       471|       2|       1|    2004|     942|
|=|       472|       2|       1|    2004|     944|
|=|       473|       2|       1|    2004|     946|
|=|       474|       2|       1|    2002|     948|
|=|       475|       2|       1|    2003|     950|
|=|       476|       2|       1|    2002|     952|
|=|       477|       2|       1|    2003|     954|
|=|       478|       2|       1|    2003|     956|
|=|       479|       2|       1|    2005|     958|
|=|       480|       2|       1|    2006|     960|
|=|       481|       2|       1|    2005|     962|
|=|       482|       2|       1|    2004|     964|
|=|       483|       2|       1|    2005|     966|
|=|       484|       2|       1|    2004|     968|
|=|       485|       2|       1|    2004|     970|
|=|       486|       2|       1|    2004|     972|
|=|       487|       2|       1|    2004|     974|
|=|       488|       2|       1|    2005|     976|
|=|       489|       2|       1|    2006|     978|
|=|       490|       2|       1|    2006|     980|
|=|       491|       2|       1|    2006|     982|
|=|       492|       2|       1|    2004|     984|
|=|       493|       2|       1|    2005|     986|
|=|       494|       2|       1|    2005|     988|
|=|       495|       2|       1|    2005|     990|
|=|       496|       2|       1|    2005|     992|
|=|       497|       2|       1|    2003|     994|
|=|       498|       2|       1|    2003|     996|
|=|       499|       2|       1|    2003|     998|
|=|       500|       2|       1|    2003|    1000|
|-|       501|       1|       1|    2002|    1002|
+-+----------+--------+--------+--------+--------+

Selected candidate:        1 mean value:     2002

Description of the selected candidate:
  label path
1 CONS1
                                                                 command
1 --fill-ratio 100 --paramfiles /home/ralph/Speciale/settings/i1-2.cfg


>
```

C.4 Parameter Selection for the ABHC

```
R version 2.9.2 (2009-08-24)
Copyright (C) 2009 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

REvolution R enhancements not installed.  For improved
performance and other extensions: apt-get install revolution-r

> library("race")
> class <- "/home/ralph/Speciale/experiments/all-instances";
> O<-race("wrapper-race.R",
          maxExp=5000,
          stat.test=c("friedman"),
          conf.level=0.95,
          first.test=5,
          interactive=TRUE,
          log.file="race.log",
          no.slaves=34)
```

```
Racing methods for the selection of the best
Copyright (C) 2003 Mauro Birattari
This software comes with ABSOLUTELY NO WARRANTY

Race name.....................................Model Selection:
Number of candidates.........................................32
Number of available tasks...................................756
Max number of experiments..................................5000
Statistical test...................................Friedman test
Tasks seen before discarding..................................5
Initialization function......................................ok
Parallel Virtual Machine....................................yes

                         Markers:
                            x No test is performed.
                            - The test is performed and
                              some candidates are discarded.
                            = The test is performed but
                              no candidate is discarded.

+-+-----------+-----------+-----------+-----------+-----------+
| |       Task|      Alive|       Best|  Mean best| Exp so far|
+-+-----------+-----------+-----------+-----------+-----------+
|x|          1|         32|         28|       1285|         32|
|x|          2|         32|          7|        954|         64|
|x|          3|         32|          7|       1204|         96|
|x|          4|         32|         15|       1218|        128|
|-|          5|         14|          7|       1238|        160|
|-|          6|          9|          8|       1139|        174|
|=|          7|          9|          7|       1168|        183|
|=|          8|          9|          8|       1162|        192|
|=|          9|          9|          7|       1247|        201|
|=|         10|          9|          8|       1239|        210|
|=|         11|          9|          8|       1206|        219|
|-|         12|          7|         11|       1154|        228|
|=|         13|          7|         15|       1106|        235|
|=|         14|          7|         15|       1150|        242|
|-|         15|          5|         15|       1190|        249|
|=|         16|          5|         15|       1214|        254|
|=|         17|          5|         15|       1212|        259|
|=|         18|          5|         15|       1239|        264|
|=|         19|          5|         15|       1237|        269|
|=|         20|          5|         15|       1236|        274|
|=|         21|          5|         15|       1268|        279|
|=|         22|          5|         15|       1263|        284|
|=|         23|          5|         15|       1234|        289|
|=|         24|          5|         15|       1233|        294|
|=|         25|          5|         15|       1236|        299|
|=|         26|          5|         15|       1215|        304|
|=|         27|          5|         15|       1232|        309|
|=|         28|          5|         15|       1219|        314|
|=|         29|          5|         15|       1236|        319|
|=|         30|          5|         15|       1238|        324|
|=|         31|          5|         15|       1221|        329|
|=|         32|          5|         15|       1234|        334|
|=|         33|          5|         15|       1230|        339|
|=|         34|          5|         15|       1247|        344|
|=|         35|          5|         15|       1230|        349|
|=|         36|          5|         15|       1241|        354|
|=|         37|          5|         15|       1246|        359|
|=|         38|          5|         15|       1269|        364|
|=|         39|          5|         15|       1262|        369|
|=|         40|          5|         15|       1283|        374|
|=|         41|          5|         15|       1294|        379|
|=|         42|          5|         15|       1300|        384|
|=|         43|          5|         15|       1298|        389|
|=|         44|          5|         15|       1306|        394|
|=|         45|          5|         15|       1304|        399|
|=|         46|          5|         15|       1302|        404|
|=|         47|          5|         15|       1308|        409|
|=|         48|          5|         15|       1306|        414|
|=|         49|          5|         15|       1315|        419|
|-|         50|          2|         15|       1333|        424|
|=|         51|          2|         15|       1331|        426|
|=|         52|          2|         15|       1336|        428|
|=|         53|          2|         15|       1340|        430|
|=|         54|          2|         15|       1344|        432|
|=|         55|          2|         15|       1332|        434|
|=|         56|          2|         15|       1350|        436|
|=|         57|          2|         15|       1349|        438|
|=|         58|          2|         15|       1337|        440|
|=|         59|          2|         15|       1335|        442|
|=|         60|          2|         15|       1332|        444|
|=|         61|          2|         15|       1341|        446|
|=|         62|          2|         15|       1341|        448|
|=|         63|          2|         15|       1330|        450|
```

=	64	2	15	1333	452
=	65	2	15	1325	454
=	66	2	15	1321	456
=	67	2	15	1310	458
=	68	2	15	1315	460
=	69	2	15	1321	462
=	70	2	15	1315	464
=	71	2	15	1319	466
=	72	2	15	1310	468
=	73	2	15	1316	470
=	74	2	15	1311	472
=	75	2	15	1303	474
=	76	2	15	1306	476
=	77	2	15	1305	478
=	78	2	15	1305	480
=	79	2	15	1310	482
=	80	2	15	1315	484
=	81	2	15	1315	486
=	82	2	15	1320	488
=	83	2	15	1318	490
=	84	2	15	1322	492
=	85	2	15	1331	494
=	86	2	15	1336	496
=	87	2	15	1327	498
=	88	2	15	1326	500
=	89	2	15	1321	502
=	90	2	15	1317	504
=	91	2	15	1315	506
=	92	2	15	1307	508
=	93	2	15	1305	510
=	94	2	15	1304	512
=	95	2	15	1308	514
=	96	2	15	1312	516
=	97	2	15	1311	518
=	98	2	15	1316	520
=	99	2	15	1316	522
=	100	2	15	1319	524
=	101	2	15	1319	526
=	102	2	15	1318	528
=	103	2	15	1316	530
=	104	2	15	1309	532
=	105	2	15	1309	534
=	106	2	15	1309	536
=	107	2	15	1307	538
=	108	2	15	1302	540
=	109	2	15	1304	542
=	110	2	15	1307	544
=	111	2	15	1306	546
=	112	2	15	1306	548
=	113	2	15	1305	550
=	114	2	15	1308	552
=	115	2	15	1311	554
=	116	2	15	1311	556
=	117	2	15	1308	558
=	118	2	15	1307	560
=	119	2	15	1304	562
=	120	2	15	1308	564
=	121	2	15	1307	566
=	122	2	15	1307	568
=	123	2	15	1309	570

Appendix D

Results for Construction Heuristics

D.1 ETTRP instances R101 and R102

ETTRP Instances R101 and R102									
25% Truck customers, 50-220 Truck Capacity, 50-220 Trailer Capacity									
Problem ID	Truck Value	NV	Compl. Veh.	NV	Problem ID	Truck Value	NV	Compl. Veh.	NV
R101_25_50_50	2388.81	31	2142.66	24	R102_25_50_50	2238.05	31	1900.53	21
R101_25_60_60	2183.65	26	2074.83	24	R102_25_60_60	2051.09	27	1872.67	20
R101_25_70_70	1932.93	24	2068.17	23	R102_25_70_70	1833.89	23	1860.27	19
R101_25_80_80	1942.29	23	2002.59	22	R102_25_80_80	1891.35	22	1839.59	19
R101_25_90_90	1895.05	22	2002.59	22	R102_25_90_90	1792.40	20	1816.57	19
R101_25_100_100	1815.25	20	2002.59	22	R102_25_100_100	1787.34	19	1816.57	19
R101_25_110_110	1814.91	21	2002.59	22	R102_25_110_110	1741.41	19	1854.12	19
R101_25_120_120	1872.80	21	2002.59	22	R102_25_120_120	1694.50	19	1854.12	19
R101_25_130_130	1872.80	21	2002.59	22	R102_25_120_130	1710.25	19	1830.43	19
R101_25_140_140	1872.80	21	2002.59	22	R102_25_140_140	1721.25	19	1830.43	19
R101_25_150_150	1872.80	21	2002.59	22	R102_25_150_150	1718.14	19	1830.43	19
R101_25_160_160	1872.80	21	2002.59	22	R102_25_160_160	1722.97	19	1830.43	19
R101_25_170_170	1872.80	21	2002.59	22	R102_25_170_170	1709.27	19	1830.43	19
R101_25_180_180	1872.80	21	2002.59	22	R102_25_180_180	1709.27	19	1830.43	19
R101_25_190_190	1872.80	21	2002.59	22	R102_25_190_190	1709.27	19	1830.43	19
R101_25_200_200	1872.80	21	2002.59	22	R102_25_200_200	1704.70	19	1830.43	19
R101_25_210_210	1872.80	21	2002.59	22	R102_25_210_210	1704.70	19	1830.43	19
R101_25_220_220	1872.80	21	2002.59	22	R102_25_220_220	1704.70	19	1830.43	19
50% Truck customers, 50-220 Truck Capacity, 50-220 Trailer Capacity									
R101_50_50_50	2388.81	31	2318.20	27	R102_50_50_50	2238.05	31	2232.37	25
R101_50_60_60	2183.65	26	2331.29	27	R102_50_60_60	2051.09	27	2241.52	25
R101_50_70_70	1932.93	24	2249.60	24	R102_50_70_70	1833.89	23	2177.55	24
R101_50_80_80	1942.29	23	2242.56	24	R102_50_80_80	1891.35	22	2130.40	23
R101_50_90_90	1895.05	22	2221.75	25	R102_50_90_90	1792.40	20	2115.95	23
R101_50_100_100	1815.25	20	2170.50	24	R102_50_100_100	1787.34	19	2115.95	23
R101_50_110_110	1814.91	21	2149.42	24	R102_50_110_110	1741.41	19	2121.20	24
R101_50_120_120	1872.80	21	2149.42	24	R102_50_120_120	1694.50	19	2106.06	23
R101_50_120_130	1872.80	21	2149.42	24	R102_50_120_130	1710.25	19	2106.06	23
R101_50_140_140	1872.80	21	2149.42	24	R102_50_140_140	1721.25	19	2106.06	23
R101_50_150_150	1872.80	21	2149.42	24	R102_50_150_150	1718.14	19	2106.06	23
R101_50_160_160	1872.80	21	2149.42	24	R102_50_160_160	1722.97	19	2106.06	23
R101_50_170_170	1872.80	21	2149.42	24	R102_50_170_170	1709.27	19	2106.06	23
R101_50_180_180	1872.80	21	2149.42	24	R102_50_180_180	1709.27	19	2106.06	23
R101_50_190_190	1872.80	21	2149.42	24	R102_50_190_190	1709.27	19	2106.06	23
R101_50_200_200	1872.80	21	2149.42	24	R102_50_200_200	1704.70	19	2106.06	23
R101_50_210_210	1872.80	21	2149.42	24	R102_50_210_210	1704.70	19	2106.06	23
R101_50_220_220	1872.80	21	2149.42	24	R102_50_220_220	1704.70	19	2106.06	23
75% Truck customers, 50-220 Truck Capacity, 50-220 Trailer Capacity									
R101_75_50_50	2388.81	31	2646.06	33	R102_75_50_50	2238.05	31	2652.19	31
R101_75_60_60	2183.65	26	2620.78	31	R102_75_60_60	2051.09	27	2525.43	28
R101_75_70_70	1932.93	24	2623.51	29	R102_75_70_70	1833.89	23	2483.25	27
R101_75_80_80	1942.29	23	2526.22	28	R102_75_80_80	1891.35	22	2519.19	28
R101_75_90_90	1895.05	22	2530.92	28	R102_75_90_90	1792.40	20	2409.29	27
R101_75_100_100	1815.25	20	2514,87	28	R102_75_100_100	1787.34	19	2381.02	27
R101_75_110_110	1814.91	21	2514,87	28	R102_75_110_110	1741.41	19	2381.02	27
R101_75_120_120	1872.80	21	2514,87	28	R102_75_120_120	1694.50	19	2381.02	27
R101_75_120_130	1872.80	21	2514,87	28	R102_75_120_130	1710.25	19	2381.28	27
R101_75_140_140	1872.80	21	2514,87	28	R102_75_140_140	1721.25	19	2381.28	27
R101_75_150_150	1872.80	21	2514,87	28	R102_75_150_150	1718.14	19	2381.28	27
R101_75_160_160	1872.80	21	2514,87	28	R102_75_160_160	1722.97	19	2381.28	27
R101_75_170_170	1872.80	21	2514,87	28	R102_75_170_170	1709.27	19	2381.28	27
R101_75_180_180	1872.80	21	2514,87	28	R102_75_180_180	1709.27	19	2381.28	27
R101_75_190_190	1872.80	21	2514,87	28	R102_75_190_190	1709.27	19	2381.28	27
R101_75_200_200	1872.80	21	2514,87	28	R102_75_200_200	1704.70	19	2381.28	27
R101_75_210_210	1872.80	21	2514,87	28	R102_75_210_210	1704.70	19	2381.28	27
R101_75_220_220	1872.80	21	2514,87	28	R102_75_220_220	1704.70	19	2381.28	27

Table D.1: Compilation of results for construction heuristic runs on ETTRP instances R101_25, R101_50, R101_75, and R102_25, R102_50, R102_75. Truck and Trailer capacities vary from 50 to 220. The column denoted Truck Value contains values for solving the instance without the use of a trailer. The columns Compl. Veh. denote the values obtained by use of a complete vehicle. The columns NV denote the number of vehicles used.

D.2 ETTRP instances C101 and C102

ETTRP Instances C101 and C102									
25% Truck customers, 50-220 Truck Capacity, 50-220 Trailer Capacity									
Problem ID	Truck Value	NV	Compl. Veh.	NV	Problem ID	Truck Value	NV	Compl. Veh.	NV
C101_25_50_50	2636.21	37	1957.48	22	C102_25_50_50	2579.36	37	1841.72	21
C101_25_60_60	2223.88	31	1647.10	18	C102_25_60_60	2270.48	31	1677.95	18
C101_25_70_70	1954.15	26	1841.95	17	C102_25_70_70	1962.67	27	1651.38	15
C101_25_80_80	1756.65	23	1558.39	15	C102_25_80_80	1777.10	23	1687.57	14
C101_25_90_90	1765.54	21	1540.14	13	C102_25_90_90	1729.24	21	1624.43	13
C101_25_100_100	1508.51	19	1476.77	13	C102_25_100_100	1497.43	19	1451.89	13
C101_25_110_110	1480.63	18	1403.05	13	C102_25_110_110	1509.01	17	1506.23	13
C101_25_120_120	1443.13	17	1401.79	13	C102_25_120_120	1327.23	16	1548.14	12
C101_25_130_130	1503.92	15	1401.79	13	C102_25_120_130	1373.95	15	1571.06	12
C101_25_140_140	1366.12	15	1401.79	13	C102_25_140_140	1248.01	14	1571.06	12
C101_25_150_150	1382.73	14	1401.79	13	C102_25_150_150	1350.95	14	1571.06	12
C101_25_160_160	1328.52	15	1401.79	13	C102_25_160_160	1252.65	13	1571.06	12
C101_25_170_170	1230.38	14	1401.79	13	C102_25_170_170	1159.43	13	1571.06	12
C101_25_180_180	1246.31	14	1401.79	13	C102_25_180_180	1166.28	12	1571.06	12
C101_25_190_190	1073.67	12	1401.79	13	C102_25_190_190	1144.05	11	1571.06	12
C101_25_200_200	923.71	10	1401.79	13	C102_25_200_200	1029.44	10	1571.06	12
C101_25_210_210	905.47	10	1401.79	13	C102_25_210_210	1080.05	10	1571.06	12
C101_25_220_220	894.12	10	1401.79	13	C102_25_220_220	1120.91	10	1571.06	12
50% Truck customers, 50-220 Truck Capacity, 50-220 Trailer Capacity									
C101_50_50_50	2636.21	37	2619.35	26	C102_50_50_50	2579.36	37	2510.42	26
C101_50_60_60	2223.88	31	2267.14	22	C102_50_60_60	2270.48	31	2253.52	22
C101_50_70_70	1954.15	26	1991.91	19	C102_50_70_70	1962.67	27	2061.30	19
C101_50_80_80	1756.65	23	2008.85	17	C102_50_80_80	1777.10	23	2133.64	18
C101_50_90_90	1765.54	21	1956.48	16	C102_50_90_90	1729.24	21	2205.44	16
C101_50_100_100	1508.51	19	1897.02	16	C102_50_100_100	1497.43	19	1930.85	15
C101_50_110_110	1480.63	18	1881.01	15	C102_50_110_110	1509.01	17	2023.24	15
C101_50_120_120	1443.13	17	1881.01	15	C102_50_120_120	1327.23	16	2022.38	15
C101_50_120_130	1503.92	15	1884.42	15	C102_50_120_130	1373.95	15	2022.38	15
C101_50_140_140	1366.12	15	1884.33	15	C102_50_140_140	1248.01	14	2025.78	15
C101_50_150_150	1382.73	14	1884.33	15	C102_50_150_150	1350.95	14	2025.69	15
C101_50_160_160	1328.52	15	1884.33	15	C102_50_160_160	1252.65	13	2025.69	15
C101_50_170_170	1230.38	14	1884.33	15	C102_50_170_170	1159.43	13	2025.69	15
C101_50_180_180	1246.31	14	1884.33	15	C102_50_180_180	1166.28	12	2025.69	15
C101_50_190_190	1073.67	12	1884.33	15	C102_50_190_190	1144.05	11	2025.69	15
C101_50_200_200	923.71	10	1884.33	15	C102_50_200_200	1029.44	10	2025.69	15
C101_50_210_210	905.47	10	1884.33	15	C102_50_210_210	1080.05	10	2025.69	15
C101_50_220_220	894.12	10	1884.33	15	C102_50_220_220	1120.91	10	2025.69	15
75% Truck customers, 50-220 Truck Capacity, 50-220 Trailer Capacity									
C101_75_50_50	2636.21	37	2996.35	32	C102_75_50_50	2579.36	37	2908.86	32
C101_75_60_60	2223.88	31	2673.10	27	C102_75_60_60	2270.48	31	2670.87	27
C101_75_70_70	1954.15	26	2592.24	25	C102_75_70_70	1962.67	27	2381.65	25
C101_75_80_80	1756.65	23	2391.18	21	C102_75_80_80	1777.10	23	2351.00	20
C101_75_90_90	1765.54	21	2417.65	21	C102_75_90_90	1729.24	21	2344.58	19
C101_75_100_100	1508.51	19	2380.38	22	C102_75_100_100	1497.43	19	2379.51	19
C101_75_110_110	1480.63	18	2259.27	20	C102_75_110_110	1509.01	17	2285.23	17
C101_75_120_120	1443.13	17	2230.68	19	C102_75_120_120	1327.23	16	2243.73	17
C101_75_120_130	1503.92	15	2159.61	19	C102_75_120_130	1373.95	15	2251.21	17
C101_75_140_140	1366.12	15	2018.20	19	C102_75_140_140	1248.01	14	2060.11	16
C101_75_150_150	1382.73	14	2017.19	19	C102_75_150_150	1350.95	14	2147.80	16
C101_75_160_160	1328.52	15	1994.17	19	C102_75_160_160	1252.65	13	2147.80	16
C101_75_170_170	1230.38	14	1994.17	19	C102_75_170_170	1159.43	13	2147.80	16
C101_75_180_180	1246.31	14	1994.17	19	C102_75_180_180	1166.28	12	2147.80	16
C101_75_190_190	1073.67	12	1994.17	19	C102_75_190_190	1144.05	11	2147.80	16
C101_75_200_200	923.71	10	1994.17	19	C102_75_200_200	1029.44	10	2147.80	16
C101_75_210_210	905.47	10	1994.17	19	C102_75_210_210	1080.05	10	2147.80	16
C101_75_220_220	894.12	10	1994.17	19	C102_75_220_220	1120.91	10	2147.80	16

Table D.2: Compilation of results for construction heuristic runs on ETTRP instances C101_25, C101_50, C101_75, and C102_25, C102_50, C102_75. Truck and Trailer capacities vary from 50 to 220. The column denoted Truck Value contains values for solving the instance without the use of a trailer. The columns Compl. Veh. denote the values obtained by use of a complete vehicle. The columns NV denote the number of vehicles used.

D.3 ETTRP instances RC101 and RC102

ETTRP Instances RC101 and RC102									
25% Truck customers, 50-220 Truck Capacity, 50-220 Trailer Capacity									
Problem ID	Truck Value	NV	Compl. Veh.	NV	Problem ID	Truck Value	NV	Compl. Veh.	NV
RC101_25_50_50	3146.94	36	2455.33	23	RC102_25_50_50	3114.79	36	2384.09	23
RC101_25_60_60	2747.87	30	2186.59	21	RC102_25_60_60	2700.03	30	2320.48	20
RC101_25_70_70	2481.43	26	2254.25	21	RC102_25_70_70	2479.50	26	2273.05	21
RC101_25_80_80	2276.91	23	2111.86	20	RC102_25_80_80	2212.71	23	2128.86	18
RC101_25_90_90	2215.08	21	2106.12	20	RC102_25_90_90	2058.92	20	2116.75	18
RC101_25_100_100	1987.87	19	2104.07	20	RC102_25_100_100	2002.80	18	2077.98	18
RC101_25_110_110	1988.66	19	2097.39	20	RC102_25_110_110	1861.02	17	2077.98	18
RC101_25_120_120	1952.01	18	2097.39	20	RC102_25_120_120	1872.33	16	2077.98	18
RC101_25_130_130	1889.36	17	2097.39	20	RC102_25_120_130	1878.66	16	2077.98	18
RC101_25_140_140	1964.06	18	2097.39	20	RC102_25_140_140	1852.57	17	2031.31	17
RC101_25_150_150	1927.96	18	2039.63	19	RC102_25_150_150	1866.81	16	2022.61	17
RC101_25_160_160	1973.72	18	2032.95	19	RC102_25_160_160	1858.39	16	2022.61	17
RC101_25_170_170	1901.80	17	2032.95	19	RC102_25_170_170	1866.93	16	2022.61	17
RC101_25_180_180	1944.30	17	2032.95	19	RC102_25_180_180	1861.46	16	2022.61	17
RC101_25_190_190	1944.30	17	2032.95	19	RC102_25_190_190	1898.41	16	2022.61	17
RC101_25_200_200	1925.80	17	2032.95	19	RC102_25_200_200	1898.41	16	2022.61	17
RC101_25_210_210	1925.80	17	2032.95	19	RC102_25_210_210	1898.41	16	2022.61	17
RC101_25_220_220	1925.80	17	2032.95	19	RC102_25_220_220	1898.41	16	2022.61	17
50% Truck customers, 50-220 Truck Capacity, 50-220 Trailer Capacity									
RC101_50_50_50	3146.94	36	2899.86	28	RC102_50_50_50	3114.79	36	2713.91	26
RC101_50_60_60	2747.87	30	2593.20	25	RC102_50_60_60	2700.03	30	2465.33	23
RC101_50_70_70	2481.43	26	2523.06	24	RC102_50_70_70	2479.50	26	2400.15	22
RC101_50_80_80	2276.91	23	2579.40	24	RC102_50_80_80	2212.71	23	2309.81	21
RC101_50_90_90	2215.08	21	2374.42	22	RC102_50_90_90	2058.92	20	2242.10	19
RC101_50_100_100	1987.87	19	2297.01	22	RC102_50_100_100	2002.80	18	2342.61	20
RC101_50_110_110	1988.66	19	2312.59	22	RC102_50_110_110	1861.02	17	2338.54	20
RC101_50_120_120	1952.01	18	2322.99	22	RC102_50_120_120	1872.33	16	2197.59	19
RC101_50_120_130	1889.36	17	2319.62	22	RC102_50_120_130	1878.66	16	2197.59	19
RC101_50_140_140	1964.06	18	2257.45	21	RC102_50_140_140	1852.57	17	2197.59	19
RC101_50_150_150	1927.96	18	2257.45	21	RC102_50_150_150	1866.81	16	2197.59	19
RC101_50_160_160	1973.72	18	2257.45	21	RC102_50_160_160	1858.39	16	2197.59	19
RC101_50_170_170	1901.80	17	2257.45	21	RC102_50_170_170	1866.93	16	2197.59	19
RC101_50_180_180	1944.30	17	2257.45	21	RC102_50_180_180	1861.46	16	2197.59	19
RC101_50_190_190	1944.30	17	2257.45	21	RC102_50_190_190	1898.41	16	2197.59	19
RC101_50_200_200	1925.80	17	2257.45	21	RC102_50_200_200	1898.41	16	2197.59	19
RC101_50_210_210	1925.80	17	2257.45	21	RC102_50_210_210	1898.41	16	2197.59	19
RC101_50_220_220	1925.80	17	2257.45	21	RC102_50_220_220	1898.41	16	2197.59	19
75% Truck customers, 50-220 Truck Capacity, 50-220 Trailer Capacity									
RC101_75_50_50	3146.94	36	3299.67	33	RC102_75_50_50	3114.79	36	3209.84	31
RC101_75_60_60	2747.87	30	2884.75	29	RC102_75_60_60	2700.03	30	2843.05	28
RC101_75_70_70	2481.43	26	2700.56	27	RC102_75_70_70	2479.50	26	2690.26	26
RC101_75_80_80	2276.91	23	2615.84	26	RC102_75_80_80	2212.71	23	2624.30	23
RC101_75_90_90	2215.08	21	2627.61	24	RC102_75_90_90	2058.92	20	2453.95	22
RC101_75_100_100	1987.87	19	2459.72	23	RC102_75_100_100	2002.80	18	2299.10	21
RC101_75_110_110	1988.66	19	2461.11	23	RC102_75_110_110	1861.02	17	2246.19	20
RC101_75_120_120	1952.01	18	2459.22	23	RC102_75_120_120	1872.33	16	2242.77	20
RC101_75_120_130	1889.36	17	2395.78	22	RC102_75_120_130	1878.66	16	2187.42	20
RC101_75_140_140	1964.06	18	2403.45	22	RC102_75_140_140	1852.57	17	2188.26	20
RC101_75_150_150	1927.96	18	2408.54	22	RC102_75_150_150	1866.81	16	2163.27	20
RC101_75_160_160	1973.72	18	2408.54	22	RC102_75_160_160	1858.39	16	2163.27	20
RC101_75_170_170	1901.80	17	2408.54	22	RC102_75_170_170	1866.93	16	2166.36	20
RC101_75_180_180	1944.30	17	2408.54	22	RC102_75_180_180	1861.46	16	2166.36	20
RC101_75_190_190	1944.30	17	2408.54	22	RC102_75_190_190	1898.41	16	2166.36	20
RC101_75_200_200	1925.80	17	2408.54	22	RC102_75_200_200	1898.41	16	2166.36	20
RC101_75_210_210	1925.80	17	2408.54	22	RC102_75_210_210	1898.41	16	2166.36	20
RC101_75_220_220	1925.80	17	2408.54	22	RC102_75_220_220	1898.41	16	2166.36	20

Table D.3: Compilation of results for construction heuristic runs on ETTRP instances RC101_25, RC101_50, RC101_75, and RC102_25, RC102_50, RC102_75. Truck and Trailer capacities vary from 50 to 220. The column denoted Truck Value contains values for solving the instance without the use of a trailer. The columns Compl. Veh. denote the values obtained by use of a complete vehicle. The columns NV denote the number of vehicles used.

D.4 ETTRP instances R201 and R202

ETTRP Instances R201 and R202									
25% Truck customers, 50-1200 Truck Capacity, 50-1200 Trailer Capacity									
Problem ID	Truck Value	NV	Compl. Veh.	NV	Problem ID	Truck Value	NV	Compl. Veh.	NV
R201_25_50_50	2357.71	31	1883.83	17	R202_25_50_50	2270.96	30	1802.83	17
R201_25_100_100	1709.21	15	1769.32	10	R202_25_100_100	1612.14	15	1648.16	9
R201_25_150_150	1467.57	10	1999.44	9	R202_25_150_150	1435.18	10	1681.32	8
R201_25_200_200	1592.83	8	1930.26	8	R202_25_200_200	1370.17	8	1857.81	7
R201_25_250_250	1478.49	7	1959.47	8	R202_25_250_250	1399.32	7	1854.69	7
R201_25_300_300	1613.88	5	1959.47	8	R202_25_300_300	1532.15	6	1843.71	6
R201_25_350_350	1748.73	5	1959.47	8	R202_25_350_350	1378.18	6	1843.71	6
R201_25_400_400	1864.75	5	1959.47	8	R202_25_400_400	1528.30	5	1843.71	6
R201_25_450_450	1757.54	5	1959.47	8	R202_25_450_450	1619.17	5	1843.71	6
R201_25_500_500	1829.73	5	1959.47	8	R202_25_500_500	1614.78	5	1843.71	6
R201_25_550_550	1809.49	5	1959.47	8	R202_25_550_550	1532.36	5	1843.71	6
R201_25_600_600	1809.49	5	1959.47	8	R202_25_600_600	1572.93	4	1843.71	6
R201_25_650_650	1809.49	5	1959.47	8	R202_25_650_650	1572.93	4	1843.71	6
R201_25_700_700	1809.49	5	1959.47	8	R202_25_700_700	1572.93	4	1843.71	6
R201_25_750_750	1809.49	5	1959.47	8	R202_25_750_750	1572.93	4	1843.71	6
R201_25_800_800	1809.49	5	1959.47	8	R202_25_800_800	1572.93	4	1843.71	6
R201_25_850_850	1809.49	5	1959.47	8	R202_25_850_850	1572.93	4	1843.71	6
R201_25_900_900	1809.49	5	1959.47	8	R202_25_900_900	1572.93	4	1843.71	6
R201_25_950_950	1809.49	5	1959.47	8	R202_25_950_950	1572.93	4	1843.71	6
R201_25_1000_1000	1809.49	5	1959.47	8	R202_25_1000_1000	1572.93	4	1843.71	6
R201_25_1050_1050	1809.49	5	1959.47	8	R202_25_1050_1050	1572.93	4	1843.71	6
R201_25_1100_1100	1809.49	5	1959.47	8	R202_25_1100_1100	1572.93	4	1843.71	6
R201_25_1150_1150	1809.49	5	1959.47	8	R202_25_1150_1150	1572.93	4	1843.71	6
R201_25_1200_1200	1809.49	5	1959.47	8	R202_25_1200_1200	1572.93	4	1843.71	6
50% Truck customers, 50-1200 Truck Capacity, 50-1200 Trailer Capacity									
R201_50_50_50	2357.71	31	2235.48	21	R202_50_50_50	2270.96	30	2202.29	20
R201_50_100_100	1709.21	15	2086.50	13	R202_50_100_100	1612.14	15	1838.29	11
R201_50_150_150	1467.57	10	2088.37	10	R202_50_150_150	1435.18	10	1883.15	7
R201_50_200_200	1592.83	8	2043.82	10	R202_50_200_200	1370.17	8	1873.63	7
R201_50_250_250	1478.49	7	2043.82	10	R202_50_250_250	1399.32	7	2001.01	7
R201_50_300_300	1613.88	5	2043.82	10	R202_50_300_300	1532.15	6	2001.01	7
R201_50_350_350	1748.73	5	2043.82	10	R202_50_350_350	1378.18	6	2001.01	7
R201_50_400_400	1864.75	5	2043.82	10	R202_50_400_400	1528.30	5	2001.01	7
R201_50_450_450	1757.54	5	2043.82	10	R202_50_450_450	1619.17	5	2001.01	7
R201_50_500_500	1829.73	5	2043.82	10	R202_50_500_500	1614.78	5	2001.01	7
R201_50_550_550	1809.49	5	2043.82	10	R202_50_550_550	1532.36	5	2001.01	7
R201_50_600_600	1809.49	5	2043.82	10	R202_50_600_600	1572.93	4	2001.01	7
R201_50_650_650	1809.49	5	2043.82	10	R202_50_650_650	1572.93	4	2001.01	7
R201_50_700_700	1809.49	5	2043.82	10	R202_50_700_700	1572.93	4	2001.01	7
R201_50_750_750	1809.49	5	2043.82	10	R202_50_750_750	1572.93	4	2001.01	7
R201_50_800_800	1809.49	5	2043.82	10	R202_50_800_800	1572.93	4	2001.01	7
R201_50_850_850	1809.49	5	2043.82	10	R202_50_850_850	1572.93	4	2001.01	7
R201_50_900_900	1809.49	5	2043.82	10	R202_50_900_900	1572.93	4	2001.01	7
R201_50_950_950	1809.49	5	2043.82	10	R202_50_950_950	1572.93	4	2001.01	7
R201_50_1000_1000	1809.49	5	2043.82	10	R202_50_1000_1000	1572.93	4	2001.01	7
R201_50_1050_1050	1809.49	5	2043.82	10	R202_50_1050_1050	1572.93	4	2001.01	7
R201_50_1100_1100	1809.49	5	2043.82	10	R202_50_1100_1100	1572.93	4	2001.01	7
R201_50_1150_1150	1809.49	5	2043.82	10	R202_50_1150_1150	1572.93	4	2001.01	7
R201_50_1200_1200	1809.49	5	2043.82	10	R202_50_1200_1200	1572.93	4	2001.01	7
75% Truck customers, 50-1200 Truck Capacity, 50-1200 Trailer Capacity									
R201_75_50_50	2357.71	31	2502.24	28	R202_75_50_50	2270.96	30	2402.85	27
R201_75_100_100	1709.21	15	2280.95	15	R202_75_100_100	1612.14	15	2105.41	15
R201_75_150_150	1467.57	10	2235.47	13	R202_75_150_150	1435.18	10	2027.18	10
R201_75_200_200	1592.83	8	2280.65	13	R202_75_200_200	1370.17	8	2057.21	9
R201_75_250_250	1478.49	7	2280.65	13	R202_75_250_250	1399.32	7	2057.21	9
R201_75_300_300	1613.88	5	2280.65	13	R202_75_300_300	1532.15	6	2057.21	9
R201_75_350_350	1748.73	5	2280.65	13	R202_75_350_350	1378.18	6	2057.21	9
R201_75_400_400	1864.75	5	2280.65	13	R202_75_400_400	1528.30	5	2057.21	9
R201_75_450_450	1757.54	5	2280.65	13	R202_75_450_450	1619.17	5	2057.21	9
R201_75_500_500	1829.73	5	2280.65	13	R202_75_500_500	1614.78	5	2057.21	9
R201_75_550_550	1809.49	5	2280.65	13	R202_75_550_550	1532.36	5	2057.21	9
R201_75_600_600	1809.49	5	2280.65	13	R202_75_600_600	1572.93	4	2057.21	9
R201_75_650_650	1809.49	5	2280.65	13	R202_75_650_650	1572.93	4	2057.21	9
R201_75_700_700	1809.49	5	2280.65	13	R202_75_700_700	1572.93	4	2057.21	9
R201_75_750_750	1809.49	5	2280.65	13	R202_75_750_750	1572.93	4	2057.21	9
R201_75_800_800	1809.49	5	2280.65	13	R202_75_800_800	1572.93	4	2057.21	9
R201_75_850_850	1809.49	5	2280.65	13	R202_75_850_850	1572.93	4	2057.21	9
R201_75_900_900	1809.49	5	2280.65	13	R202_75_900_900	1572.93	4	2057.21	9
R201_75_950_950	1809.49	5	2280.65	13	R202_75_950_950	1572.93	4	2057.21	9
R201_75_1000_1000	1809.49	5	2280.65	13	R202_75_1000_1000	1572.93	4	2057.21	9
R201_75_1050_1050	1809.49	5	2280.65	13	R202_75_1050_1050	1572.93	4	2057.21	9
R201_75_1100_1100	1809.49	5	2280.65	13	R202_75_1100_1100	1572.93	4	2057.21	9
R201_75_1150_1150	1809.49	5	2280.65	13	R202_75_1150_1150	1572.93	4	2057.21	9
R201_75_1200_1200	1809.49	5	2280.65	13	R202_75_1200_1200	1572.93	4	2057.21	9

Table D.4: Compilation of results for construction heuristic runs on ETTRP instances R201_25, R201_50, R201_75, and R202_25, R202_50, R202_75. Truck and Trailer capacities vary from 50 to 220. The column denoted Truck Value contains values for solving the instance without the use of a trailer. The columns Compl. Veh. denote the values obtained by use of a complete vehicle. The columns NV denote the number of vehicles used.

D.5 ETTRP instances C201 and C202

ETTRP Instances C201 and C202									
25% Truck customers, 50-1200 Truck Capacity, 50-1200 Trailer Capacity									
Problem ID	Truck Value	NV	Compl. Veh.	NV	Problem ID	Truck Value	NV	Compl. Veh.	NV
C201_25_50_50	2659.93	37	2087.75	21	C202_25_50_50	2689.69	37	2073.04	21
C201_25_100_100	1574.54	19	1385.96	10	C202_25_100_100	1675.50	19	1448.92	11
C201_25_150_150	1292.31	13	1215.89	7	C202_25_150_150	1237.80	13	1269.28	8
C201_25_200_200	1044.47	10	1176.01	6	C202_25_200_200	1081.90	10	1337.81	7
C201_25_250_250	1003.15	9	1080.34	6	C202_25_250_250	955.49	9	1151.79	6
C201_25_300_300	908.30	7	949.234	4	C202_25_300_300	953.14	7	997.26	5
C201_25_350_350	961.17	6	937.33	4	C202_25_350_350	1107.34	6	1029.18	4
C201_25_400_400	958.41	6	937.33	4	C202_25_400_400	1141.61	6	1029.18	4
C201_25_450_450	937.61	6	937.33	4	C202_25_450_450	1003.60	6	1029.18	4
C201_25_500_500	831.74	6	937.33	4	C202_25_500_500	936.55	6	1029.18	4
C201_25_550_550	690.60	5	937.33	4	C202_25_550_550	817.45	5	1029.18	4
C201_25_600_600	635.77	4	937.33	4	C202_25_600_600	835.91	5	1029.18	4
C201_25_650_650	603.88	3	937.33	4	C202_25_650_650	795.61	4	1029.18	4
C201_25_700_700	603.88	3	937.33	4	C202_25_700_700	795.61	4	1029.18	4
C201_25_750_750	603.88	3	937.33	4	C202_25_750_750	795.61	4	1029.18	4
C201_25_800_800	603.88	3	937.33	4	C202_25_800_800	795.61	4	1029.18	4
C201_25_850_850	603.88	3	937.33	4	C202_25_850_850	795.61	4	1029.18	4
C201_25_900_900	603.88	3	937.33	4	C202_25_900_900	795.61	4	1029.18	4
C201_25_950_950	603.88	3	937.33	4	C202_25_950_950	795.61	4	1029.18	4
C201_25_1000_1000	603.88	3	937.33	4	C202_25_1000_1000	795.61	4	1029.18	4
C201_25_1050_1050	603.88	3	937.33	4	C202_25_1050_1050	795.61	4	1029.18	4
C201_25_1100_1100	603.88	3	937.33	4	C202_25_1100_1100	795.61	4	1029.18	4
C201_25_1150_1150	603.88	3	937.33	4	C202_25_1150_1150	795.61	4	1029.18	4
C201_25_1200_1200	603.88	3	937.33	4	C202_25_1200_1200	795.61	4	1029.18	4
50% Truck customers, 50-1200 Truck Capacity, 50-1200 Trailer Capacity									
C201_50_50_50	2659.93	37	2627.75	26	C202_50_50_50	2689.69	37	2615.11	25
C201_50_100_100	1574.54	19	1772.10	13	C202_50_100_100	1675.50	19	1766.01	13
C201_50_150_150	1292.31	13	1685.75	9	C202_50_150_150	1237.80	13	1469.83	9
C201_50_200_200	1044.47	10	1668.41	7	C202_50_200_200	1081.90	10	1449.74	7
C201_50_250_250	1003.15	9	1740.54	6	C202_50_250_250	955.49	9	1422.07	6
C201_50_300_300	908.30	7	1639.38	6	C202_50_300_300	953.14	7	1507.98	6
C201_50_350_350	961.17	6	1605.66	6	C202_50_350_350	1107.34	6	1436.22	6
C201_50_400_400	958.41	6	1605.66	6	C202_50_400_400	1141.61	6	1491.69	6
C201_50_450_450	937.61	6	1605.66	6	C202_50_450_450	1003.60	6	1491.69	6
C201_50_500_500	831.74	6	1605.66	6	C202_50_500_500	936.55	6	1491.69	6
C201_50_550_550	690.60	5	1605.66	6	C202_50_550_550	817.45	5	1491.69	6
C201_50_600_600	635.77	4	1605.66	6	C202_50_600_600	835.91	5	1491.69	6
C201_50_650_650	603.88	3	1605.66	6	C202_50_650_650	795.61	4	1491.69	6
C201_50_700_700	603.88	3	1605.66	6	C202_50_700_700	795.61	4	1491.69	6
C201_50_750_750	603.88	3	1605.66	6	C202_50_750_750	795.61	4	1491.69	6
C201_50_800_800	603.88	3	1605.66	6	C202_50_800_800	795.61	4	1491.69	6
C201_50_850_850	603.88	3	1605.66	6	C202_50_850_850	795.61	4	1491.69	6
C201_50_900_900	603.88	3	1605.66	6	C202_50_900_900	795.61	4	1491.69	6
C201_50_950_950	603.88	3	1605.66	6	C202_50_950_950	795.61	4	1491.69	6
C201_50_1000_1000	603.88	3	1605.66	6	C202_50_1000_1000	795.61	4	1491.69	6
C201_50_1050_1050	603.88	3	1605.66	6	C202_50_1050_1050	795.61	4	1491.69	6
C201_50_1100_1100	603.88	3	1605.66	6	C202_50_1100_1100	795.61	4	1491.69	6
C201_50_1150_1150	603.88	3	1605.66	6	C202_50_1150_1150	795.61	4	1491.69	6
C201_50_1200_1200	603.88	3	1605.66	6	C202_50_1200_1200	795.61	4	1491.69	6
75% Truck customers, 50-1200 Truck Capacity, 50-1200 Trailer Capacity									
C201_75_50_50	2659.93	37	3156.08	31	C202_75_50_50	2689.69	37	3045.71	31
C201_75_100_100	1574.54	19	2089.54	16	C202_75_100_100	1675.50	19	2088.72	16
C201_75_150_150	1292.31	13	2087.23	13	C202_75_150_150	1237.80	13	2013.47	12
C201_75_200_200	1044.47	10	1843.45	11	C202_75_200_200	1081.90	10	1763.05	9
C201_75_250_250	1003.15	9	1740.76	10	C202_75_250_250	955.49	9	1759.14	8
C201_75_300_300	908.30	7	1650.41	9	C202_75_300_300	953.14	7	1683.06	7
C201_75_350_350	961.17	6	1788.92	8	C202_75_350_350	1107.34	6	1683.06	7
C201_75_400_400	958.41	6	1622.29	8	C202_75_400_400	1141.61	6	1531.23	6
C201_75_450_450	937.61	6	1626,25	8	C202_75_450_450	1003.60	6	1515.18	6
C201_75_500_500	831.74	6	1626,25	8	C202_75_500_500	936.55	6	1515.18	6
C201_75_550_550	690.60	5	1626,25	8	C202_75_550_550	817.45	5	1515.18	6
C201_75_600_600	635.77	4	1626,25	8	C202_75_600_600	835.91	5	1515.18	6
C201_75_650_650	603.88	3	1626,25	8	C202_75_650_650	795.61	4	1515.18	6
C201_75_700_700	603.88	3	1626,25	8	C202_75_700_700	795.61	4	1515.18	6
C201_75_750_750	603.88	3	1626,25	8	C202_75_750_750	795.61	4	1515.18	6
C201_75_800_800	603.88	3	1626,25	8	C202_75_800_800	795.61	4	1515.18	6
C201_75_850_850	603.88	3	1626,25	8	C202_75_850_850	795.61	4	1515.18	6
C201_75_900_900	603.88	3	1626,25	8	C202_75_900_900	795.61	4	1515.18	6
C201_75_950_950	603.88	3	1626,25	8	C202_75_950_950	795.61	4	1515.18	6
C201_75_1000_1000	603.88	3	1626,25	8	C202_75_1000_1000	795.61	4	1515.18	6
C201_75_1050_1050	603.88	3	1626,25	8	C202_75_1050_1050	795.61	4	1515.18	6
C201_75_1100_1100	603.88	3	1626,25	8	C202_75_1100_1100	795.61	4	1515.18	6
C201_75_1150_1150	603.88	3	1626,25	8	C202_75_1150_1150	795.61	4	1515.18	6
C201_75_1200_1200	603.88	3	1626,25	8	C202_75_1200_1200	795.61	4	1515.18	6

Table D.5: Compilation of results for construction heuristic runs on ETTRP instances C201_25, C201_50, C201_75, and C202_25, C202_50, C202_75. Truck and Trailer capacities vary from 50 to 220. The column denoted Truck Value contains values for solving the instance without the use of a trailer. The columns Compl. Veh. denote the values obtained by use of a complete vehicle. The columns NV denote the number of vehicles used.

D.6 ETTRP instances RC201 and RC202

ETTRP Instances RC201 and RC202									
25% Truck customers, 50-1200 Truck Capacity, 50-1200 Trailer Capacity									
Problem ID	Truck Value	NV	Compl. Veh.	NV	Problem ID	Truck Value	NV	Compl. Veh.	NV
RC201_25_50_50	3096.99	36	2474.23	21	RC202_25_50_50	3058.94	36	2312.83	22
RC201_25_100_100	2203.60	18	2266.00	12	RC202_25_100_100	1922.98	18	1717.86	11
RC201_25_150_150	1926.30	12	2302.44	10	RC202_25_150_150	1617.21	12	1761.62	9
RC201_25_200_200	1911.25	10	2383.39	10	RC202_25_200_200	1584.99	9	1889.52	8
RC201_25_250_250	2397.22	8	2360.34	9	RC202_25_250_250	1507.58	7	2066.08	8
RC201_25_300_300	2320.72	7	2360.34	9	RC202_25_300_300	1523.49	6	2064.98	8
RC201_25_350_350	2474.49	6	2360.34	9	RC202_25_350_350	1558.38	6	2064.98	8
RC201_25_400_400	2554.39	6	2360.34	9	RC202_25_400_400	1689.63	5	2064.98	8
RC201_25_450_450	2540.76	6	2360.34	9	RC202_25_450_450	1744.87	5	2064.98	8
RC201_25_500_500	2540.76	6	2360.34	9	RC202_25_500_500	1693.29	5	2064.98	8
RC201_25_550_550	2540.76	6	2360.34	9	RC202_25_550_550	1703.28	5	2064.98	8
RC201_25_600_600	2540.76	6	2360.34	9	RC202_25_600_600	1876.93	4	2064.98	8
RC201_25_650_650	2540.76	6	2360.34	9	RC202_25_650_650	1844.61	5	2064.98	8
RC201_25_700_700	2540.76	6	2360.34	9	RC202_25_700_700	1811.75	4	2064.98	8
RC201_25_750_750	2540.76	6	2360.34	9	RC202_25_750_750	1811.75	4	2064.98	8
RC201_25_800_800	2540.76	6	2360.34	9	RC202_25_800_800	1811.75	4	2064.98	8
RC201_25_850_850	2540.76	6	2360.34	9	RC202_25_850_850	1811.75	4	2064.98	8
RC201_25_900_900	2540.76	6	2360.34	9	RC202_25_900_900	1811.75	4	2064.98	8
RC201_25_950_950	2540.76	6	2360.34	9	RC202_25_950_950	1811.75	4	2064.98	8
RC201_25_1000_1000	2540.76	6	2360.34	9	RC202_25_1000_1000	1811.75	4	2064.98	8
RC201_25_1050_1050	2540.76	6	2360.34	9	RC202_25_1050_1050	1811.75	4	2064.98	8
RC201_25_1100_1100	2540.76	6	2360.34	9	RC202_25_1100_1100	1811.75	4	2064.98	8
RC201_25_1150_1150	2540.76	6	2360.34	9	RC202_25_1150_1150	1811.75	4	2064.98	8
RC201_25_1200_1200	2540.76	6	2360.34	9	RC202_25_1200_1200	1811.75	4	2064.98	8
50% Truck customers, 50-1200 Truck Capacity, 50-1200 Trailer Capacity									
RC201_50_50_50	3096.99	36	2960.34	26	RC202_50_50_50	3058.94	36	2723.69	25
RC201_50_100_100	2203.60	18	2196.60	14	RC202_50_100_100	1922.98	18	2003.19	12
RC201_50_150_150	1926.30	12	2222.52	12	RC202_50_150_150	1617.21	12	1841.84	9
RC201_50_200_200	1911.25	10	2246.17	11	RC202_50_200_200	1584.99	9	1880.35	9
RC201_50_250_250	2397.22	8	2246.17	11	RC202_50_250_250	1507.58	7	2065.23	8
RC201_50_300_300	2320.72	7	2246.17	11	RC202_50_300_300	1523.49	6	2065.23	8
RC201_50_350_350	2474.49	6	2246.17	11	RC202_50_350_350	1558.38	6	2065.23	8
RC201_50_400_400	2554.39	6	2246.17	11	RC202_50_400_400	1689.63	5	2065.23	8
RC201_50_450_450	2540.76	6	2246.17	11	RC202_50_450_450	1744.87	5	2065.23	8
RC201_50_500_500	2540.76	6	2246.17	11	RC202_50_500_500	1693.29	5	2065.23	8
RC201_50_550_550	2540.76	6	2246.17	11	RC202_50_550_550	1703.28	5	2065.23	8
RC201_50_600_600	2540.76	6	2246.17	11	RC202_50_600_600	1876.93	4	2065.23	8
RC201_50_650_650	2540.76	6	2246.17	11	RC202_50_650_650	1844.61	5	2065.23	8
RC201_50_700_700	2540.76	6	2246.17	11	RC202_50_700_700	1811.75	4	2065.23	8
RC201_50_750_750	2540.76	6	2246.17	11	RC202_50_750_750	1811.75	4	2065.23	8
RC201_50_800_800	2540.76	6	2246.17	11	RC202_50_800_800	1811.75	4	2065.23	8
RC201_50_850_850	2540.76	6	2246.17	11	RC202_50_850_850	1811.75	4	2065.23	8
RC201_50_900_900	2540.76	6	2246.17	11	RC202_50_900_900	1811.75	4	2065.23	8
RC201_50_950_950	2540.76	6	2246.17	11	RC202_50_950_950	1811.75	4	2065.23	8
RC201_50_1000_1000	2540.76	6	2246.17	11	RC202_50_1000_1000	1811.75	4	2065.23	8
RC201_50_1050_1050	2540.76	6	2246.17	11	RC202_50_1050_1050	1811.75	4	2065.23	8
RC201_50_1100_1100	2540.76	6	2246.17	11	RC202_50_1100_1100	1811.75	4	2065.23	8
RC201_50_1150_1150	2540.76	6	2246.17	11	RC202_50_1150_1150	1811.75	4	2065.23	8
RC201_50_1200_1200	2540.76	6	2246.17	11	RC202_50_1200_1200	1811.75	4	2065.23	8
75% Truck customers, 50-1200 Truck Capacity, 50-1200 Trailer Capacity									
RC201_75_50_50	3096.99	36	3243.47	31	RC202_75_50_50	3058.94	36	3148.84	31
RC201_75_100_100	2203.60	18	2394.85	17	RC202_75_100_100	1922.98	18	2244.54	17
RC201_75_150_150	1926.30	12	2421.00	14	RC202_75_150_150	1617.21	12	2065.73	12
RC201_75_200_200	1911.25	10	2207.82	12	RC202_75_200_200	1584.99	9	1936.54	11
RC201_75_250_250	2397.22	8	2373.89	12	RC202_75_250_250	1507.58	7	2094.23	10
RC201_75_300_300	2320.72	7	2373.89	12	RC202_75_300_300	1523.49	6	2094.23	10
RC201_75_350_350	2474.49	6	2373.89	12	RC202_75_350_350	1558.38	6	2094.23	10
RC201_75_400_400	2554.39	6	2373.89	12	RC202_75_400_400	1689.63	5	2094.23	10
RC201_75_450_450	2540.76	6	2373.89	12	RC202_75_450_450	1744.87	5	2094.23	10
RC201_75_500_500	2540.76	6	2373.89	12	RC202_75_500_500	1693.29	5	2094.23	10
RC201_75_550_550	2540.76	6	2373.89	12	RC202_75_550_550	1703.28	5	2094.23	10
RC201_75_600_600	2540.76	6	2373.89	12	RC202_75_600_600	1876.93	4	2094.23	10
RC201_75_650_650	2540.76	6	2373.89	12	RC202_75_650_650	1844.61	5	2094.23	10
RC201_75_700_700	2540.76	6	2373.89	12	RC202_75_700_700	1811.75	4	2094.23	10
RC201_75_750_750	2540.76	6	2373.89	12	RC202_75_750_750	1811.75	4	2094.23	10
RC201_75_800_800	2540.76	6	2373.89	12	RC202_75_800_800	1811.75	4	2094.23	10
RC201_75_850_850	2540.76	6	2373.89	12	RC202_75_850_850	1811.75	4	2094.23	10
RC201_75_900_900	2540.76	6	2373.89	12	RC202_75_900_900	1811.75	4	2094.23	10
RC201_75_950_950	2540.76	6	2373.89	12	RC202_75_950_950	1811.75	4	2094.23	10
RC201_75_1000_1000	2540.76	6	2373.89	12	RC202_75_1000_1000	1811.75	4	2094.23	10
RC201_75_1050_1050	2540.76	6	2373.89	12	RC202_75_1050_1050	1811.75	4	2094.23	10
RC201_75_1100_1100	2540.76	6	2373.89	12	RC202_75_1100_1100	1811.75	4	2094.23	10
RC201_75_1150_1150	2540.76	6	2373.89	12	RC202_75_1150_1150	1811.75	4	2094.23	10
RC201_75_1200_1200	2540.76	6	2373.89	12	RC202_75_1200_1200	1811.75	4	2094.23	10

Table D.6: Compilation of results for construction heuristic runs on ETTRP instances RC201_25, RC201_50, RC201_75, and RC202_25, RC202_50, RC202_75. Truck and Trailer capacities vary from 50 to 220. The column denoted Truck Value contains values for solving the instance without the use of a trailer. The columns Compl. Veh. denote the values obtained by use of a complete vehicle. The columns NV denote the number of vehicles used.

Appendix E

Search Progression

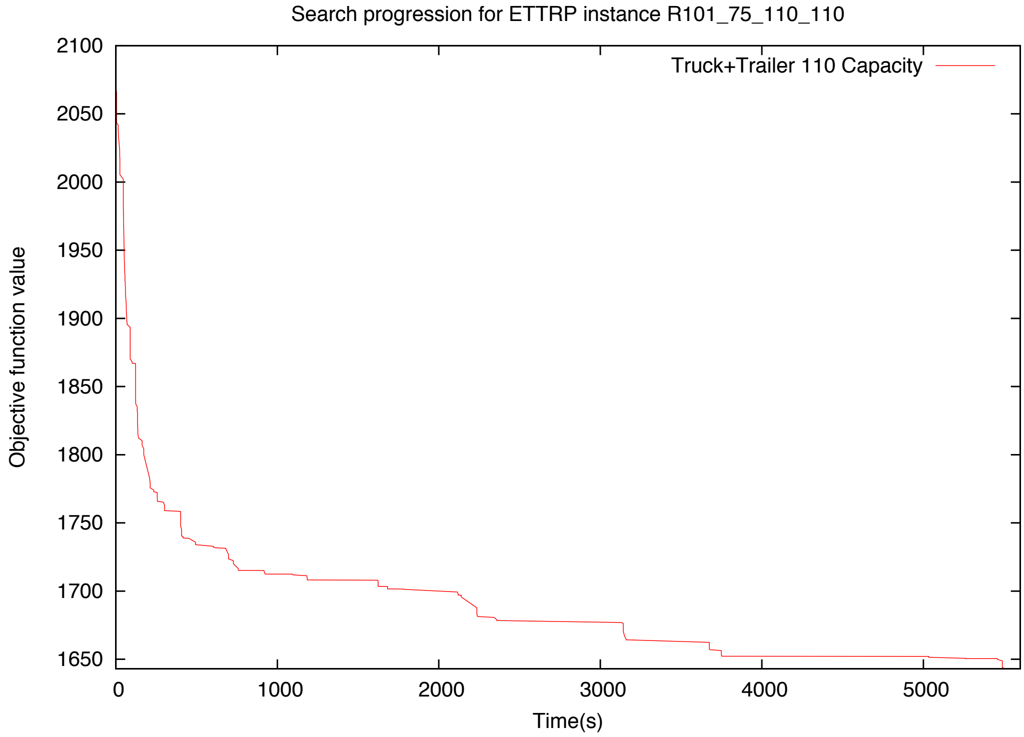


Figure E.1: Extended search progression for ETTRP instance R101_75_110_110. The figure shows the algorithms ability to produce results of similar quality for problems where the use of trailers is unattractive. The result of the search was 1643.52 using 20 vehicles and zero trailers. The best solution obtained without the use of trailers by the implemented algorithm is 1642.88 using 20 vehicles.

List of Figures

List of Tables

List of Algorithms

Notation

A: is the set of arcs, 6
C: is the capacity of a truck (CVRP), 6
$C_{trailer}$: is the capacity of a trailer, 9
$C^{i}_{trailer}$: is the capacity of the trailer immediately before visiting customer i, 9
$C^{\alpha(e)}_{truckSum}$: is the sum of demands serviced from the truck at vehicle customers from 0_s forwards until $\alpha(e)$, 32
C_{truck}: is the capacity of a truck, 9
C^{i}_{truck}: is the capacity of the truck immediately before visiting customer i, 9
E: is the set of edges, 6
G: is a complete directed graph, 6
K: is the number of vehicles used to solve the problem, 6
K_{min}: is the minimum number of vehicles needed to solve the problem, 6
$L(R)$: is the total length of the routing plan R, 7
$L(R_k)$: is the total length of route k, 7
P: is the set of available parking places, 13
$P_{\cap used}$: is the set of parking places used in several routes, 13
P_{used}: is the set of parking places used in the current routing plan, 13
R: is the set of routes, 6
V: is the set of vertices, 6
V_T: is the set of all truck customers, 10
V_V: is the set of all vehicle customers, 10
$[e_i, l_i]$: is the time window associated with customer i, 8
$\alpha(e)$: is the start vertex associated with the edge e, 6
β: is the sparsification parameter, 61
$\beta(e)$: is the end vertex associated with the edge e, 6
δ_j: is the earliest time of departure at customer j, 27
$\epsilon(i)$: is the customer type of customer i, 9
γ: is the load balance, 29
γ_σ: is the load balance for route σ, 30
γ^{i}_{σ}: is the load balance when visiting customer i in route σ, 31
$\langle c, e\rangle$: is a customer-edge pair, 46
$\mathcal{C}(R_k)$: is the set of customers of route k, 7
$\mathcal{D}(S)$: is the demand for the set of vertices S, 6
$\mathcal{E}(R_k)$: is the set of edges of route k, 7
$\mathcal{N}^{op}$: is a neighbourhood operator, 58
$\mathcal{W}(R_k)$: is the total waiting time of route k, 8
$\omega(R)$: is the objective function over the routing plan R, 7
$\otimes$: is the concatenation operator, 28
ψ_i: is the latest allowable arrival time at customer i, 27
σ_k: is an ordered sequence of vertices of route k, 6
ς: is a subsequence of customers, 28
$\varsigma^{i,j}_{k}$: is a subsequence of customers from index i to j of route k, 28
ϑ: is the granularity threshold, 61

a_i: is the arrival time at customer i, 8
b_i: is the beginning of service at customer i, 8
c_{ij}: is the non-negative cost (length) associated with arc (i, j), 6
$d(i, j)$: is the length of the arc (i, j), 6
d_i: is the demand of vertex i, 6
e_i: is the earliest start of service for customer i, 8
i^+: is the customer immediately following i, 7
i^-: is the customer immediately preceding i, 7
i_e: is used to indicate the "end" vertex of i, 7
i_s: is used to indicate the "start" vertex of i, 7
l_i: is the latest start of service for customer l, 8
w_i: is the waiting time at customer i, 8
$z^{'}$: is the value of a heuristic solution, 61
$|e|$: is the length of edge e, 6

TC: is an abbreviation of Truck Customer, 9

VC: is an abbreviation of Vehicle Customer, 9
VR: is an abbreviation of Vehicle Route, 10

Bibliography

[A/S75] Transvision A/S. Planlægnings- og optimeringssystemer til transport og distribution. `http://www.transvision.dk`, 1975.

[BBMR] R. Baldacci, E. Bartolini, A. Mingozzi, and R. Roberti. An exact solution framework for a broad class of vehicle routing problems. *Computational Management Science*, pages 1–40.

[BG05] O. Bräysy and M. Gendreau. Vehicle routing problem with time windows, Part I: Route construction and local search algorithms. *Transportation Science*, 39(1):104–118, 2005.

[Bir03] M. Birattari. The race package for R. Racing methods for the selection of the best. Technical report, Technical Report TR/IRIDIA/2003-37, IRIDIA, Universite Libre de Bruxelles, Brussels, Belgium, 2003.

[BM98] JCS Brandao and A. Mercer. The multi-trip vehicle routing problem. *Journal of the Operational research society*, pages 799–805, 1998.

[Cha02] I.M. Chao. A tabu search method for the truck and trailer routing problem. *Computers and Operations Research*, 29(1):33–51, 2002.

[CLRS01] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 2001.

[CMT79] N. Christofides, A. Mingozzi, and P. Toth. The vehicle routing problem. *Combinatorial optimization*, 11:315338, 1979.

[DB03] W. Dullaert and O. Bräysy. Routing relatively few customers per route. *Top*, 11(2):325–336, 2003.

[DK07] U. Derigs and R. Kaiser. Applying the attribute based hill climber heuristic to the vehicle routing problem. *European Journal of Operational Research*, 177(2):719–732, 2007.

[Dre07] M. Drexl. *On Some Generalized Routing Problems*. PhD thesis, RWTH Aachen University, 2007.

[FGI05] B. Funke, T. Grünert, and S. Irnich. Local search for vehicle routing and scheduling problems: Review and conceptual integration. *Journal of Heuristics*, 11(4):267–306, 2005.

[FJ81] M.L. Fisher and R. Jaikumar. Generalized assignment heuristic for vehicle routing. *NETWORKS.*, 11(2):109–124, 1981.

[G⁺89] F. Glover et al. Tabu search-part I. *ORSA journal on Computing*, 1(3):190–206, 1989.

[Ger96] J.C. Gerdessen. Vehicle routing problem with trailers. *European Journal of Operational Research*, 93(1):135–147, 1996.

[GHL94] M. Gendreau, A. Hertz, and G. Laporte. A tabu search heuristic for the vehicle routing problem. *Management Science*, 40(10):1276–1290, 1994.

[GI97] R. Gentleman and R. Ihaka. The r project for statistical computing. `http://www.r-project.org`, 1997.

[Glo90] F. Glover. Tabu search-part II. *ORSA Journal on computing*, 2(1):4–32, 1990.

[GM74] B.E. Gillett and L.R. Miller. A heuristic algorithm for the vehicle-dispatch problem. *Operations Research*, 22(2):340–349, 1974.

[GM09] K. Galschiøt Markus. Online stochastic vehicle routing, May 2009. Master's Thesis.

[HS05] H.H. Hoos and T. Stützle. *Stochastic Local Search: Foundations and Applications*. Morgan Kaufmann, 2005.

[IFG06] S. Irnich, B. Funke, and T. Grünert. Sequential search and its application to vehicle-routing problems. *Computers and Operations Research*, 33(8):2405–2429, 2006.

[Irn08a] S. Irnich. A unified modeling and solution framework for vehicle routing and local search-based metaheuristics. *INFORMS Journal on Computing*, 20(2):270, 2008.

[Irn08b] S. Irnich. Resource extension functions: Properties, inversion, and generalization to segments. *OR Spectrum*, 30(1):113–148, 2008.

[JPSP] M. Jepsen, B. Petersen, S. Spoorendonk, and D. Pisinger. Subset-row inequalities applied to the vehicle routing problem with time windows. *Cut and Column Generation*, page 25.

[KGV83] S. Kirkpatrick, CD Gelatt, and MP Vecchi. Optimization by Simulated Annealing. *Science*, 220(4598):671–680, 1983.

[LYC] S.W. Lin, V.F. Yu, and S.Y. Chou. A note on the truck and trailer routing problem. *Expert Systems With Applications*, 37(1):899–903.

[Nik09] J. Nikolajsen. Algorithms for optimizing work schedules for service personnel in the home care sector, February 2009. Master's Thesis.

[Reg98] C. Rego. A subpath ejection method for the vehicle routing problem. *Management Science*, pages 1447–1459, 1998.

[Reg01] C. Rego. Node-ejection chains for the vehicle routing problem: Sequential and parallel algorithms. *Parallel Computing*, 27(3):201–222, 2001.

[S.07] Spoorendonk S. Optimal solutions for the solomon instances. `http://www.diku.dk/hjemmesider/ansatte/spooren/solomon/vrptw.html`, September 2007.

[Sav92] M.W.P. Savelsbergh. The vehicle routing problem with time windows: minimizing route duration. *INFORMS Journal on Computing*, 4(2):146, 1992.

[Sch04] S. Scheuerer. *Neue Tabusuche-Heuristiken für die logistische Tourenplanung bei restringierendem Anhängereinsatz, mehreren Depots und Planungsperioden*. PhD thesis, Universität Regensburg, Universitätsstraße, 93042 Regensburg, 2004.

[Sem95] F. Semet. A two-phase algorithm for the partial accessibility constrained vehicle routing problem. *Annals of Operations Research*, 61(1):45–65, 1995.

[Sol84] M.M. Solomon. Vrptw benchmark problems. `http://web.cba.neu.edu/~msolomon/problems.htm`, 1984.

[Sol87] M.M. Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, pages 254–265, 1987.

[Sol05] M.M. Solomon. Vrptw best known solutions identified by heuristics. `http://web.cba.neu.edu/~msolomon/heuristi.htm`, March 2005.

[ST93] F. Semet and E. Taillard. Solving real-life vehicle routing problems efficiently using tabu search. *Annals of Operations Research*, 41(4):469–488, 1993.

[SVDHU05] H. Sontrop, P. Van Der Horn, and M. Uetz. *Fast ejection chain algorithms for vehicle routing with time windows*. Springer, 2005.

[TV87] P. Toth and D. Vigo. *The vehicle routing problem*. Society for Industrial Mathematics, 1987.

[TV02] P. Toth and D. Vigo. An overview of vehicle routing problems. *The vehicle routing problem*, pages 1–26, 2002.

[TV03] P. Toth and D. Vigo. The granular tabu search and its application to the vehicle-routing problem. *INFORMS Journal on Computing*, 15(4):333, 2003.

[WS04] I.M. Whittley and G.D. Smith. The attribute based hill climber. *Journal of Mathematical Modelling and Algorithms*, 3(2):167–178, 2004.

[YLC08] VF Yu, SW Lin, and SY Chou. Solving the truck and trailer routing problem based on a simulated annealing heuristic. *Computers & Operations Research*, pages 0305–0548, 2008.