Datalogisk Institut
KØBENHAVNS UNIVERSITET

DIKU

# A Unified Heuristic for a Large Class of Vehicle Routing Problems with Backhauls

**Stefan Røpke and David Pisinger**

**Technical Report no. 2004/14**
**ISSN: 0107-8283**

**DIKU**
University of Copenhagen • Universitetsparken 1
DK-2100 Copenhagen • Denmark

# A Unified Heuristic for a Large Class of Vehicle Routing Problems with Backhauls

Stefan Ropke and David Pisinger [*]

28th July 2004

**Abstract**

The Vehicle Routing Problem with Backhauls is a generalization of the ordinary capacitated vehicle routing problem where goods are delivered from the depot to the linehaul customers, and additional goods are brought back to the depot from the backhaul customers. Numerous ways of modeling the backhaul constraints have been proposed in the literature, each imposing different restrictions on the handling of backhaul customers. A survey of these models is presented, and a unified model is developed that is capable of handling most variants of the problem from the literature. The unified model can be seen as a Rich Pickup and Delivery Problem with Time Windows, which can be solved through an improved version of the large neighborhood search heuristic proposed by Ropke (2003). The results obtained in this way are comparable to or improve on similar results found by state of the art heuristics for the various variants of the problem. The heuristic has been tested on 338 problems from the literature and it has improved the best known solution for 227 of these. An additional benefit of the unified modeling and solution method is that it allows the dispatcher to mix various variants of the Vehicle Routing Problem with Backhauls for the individual customers or vehicles.

**Keywords: metaheuristics, vehicle routing problems, large neighborhood search**

## 1 Introduction

In the classical *Capacitated Vehicle Routing Problem* (CVRP) we have to deliver goods from a depot to a set of customers, using a set of identical vehicles. Each customer demands a certain quantity of goods and the vehicles have a limited capacity. Our task is to construct routes starting and ending at the depot that minimize the total travel distance and that obey the capacity of the vehicles.

The problems that need to be solved in real life situations are usually much more complicated. One complication that arises in practice is that goods not only need to be brought from the depot to the customers, but also must be picked up at a number of customers and brought back to the depot. A simple way of handling such problems is to solve two independent CVRPs. One for the delivery (*linehaul*) customers and one for the pickup (*backhaul*) customers, such that some vehicles would be designated to linehaul customers and others to backhaul customers. This approach is not likely to create high quality solutions though — it seems more profitable to serve both pickup and delivery customers using the same vehicles. The *Vehicle Routing Problem with Backhauls* (VRPB) models problems with both pickup and delivery customers in the same route.

[*]DIKU - Department of Computer Science, University of Copenhagen, Universitetsparken 1, DK-2100 Copenhagen Ø, Denmark. E-mail: {sropke, pisinger}@diku.dk

Applications of VRPB can be found in the distribution of groceries. Groceries are delivered to supermarkets and grocery stores from a central distribution center and groceries are picked up at production sites and brought to the distribution center. Another application is the handling of returnable bottles, where full bottles are brought to customers and empty bottles are brought back to breweries to be recycled. Such applications are likely to become more common in the future due to the increased awareness of environmental issues. It is important to develop fast and robust algorithms for real-life transportation problems, which are able to handle various side constraints that appear in practice.

The general trend in the transportation sector is that transportation companies are merging to larger units which can provide a large number of delivery services. In order to get the most possible benefit from the vehicle fleet, it can be attractive to service conceptually different transportation tasks by the same fleet, thus models are needed that can handle all additional constraints associated with a transportation task. Cordeau et al. [6] for example provide a unified approach for several Vehicle Routing Problems with Time Windows. The present paper considerably extends the expressibility of the model, by also allowing pickup and delivery requests, precedence constraints, etc. This allows us to formulate the six most common variants of vehicle routing problems with backhauls within the framework, and to find high quality heuristic solutions that are comparable to or improve on similar results for specialized algorithms.

The underlying problem of all of the problems we consider is the *Pickup and Delivery Problem with Time Windows* (PDPTW), which we will describe in Section 2. A survey of the six most common variants of vehicle routing problems with backhauls — and additional, less frequently used models — is given in Section 3. The subsequent sections present the heuristic algorithm proposed in this paper, which is outlined in Figure 1. Some of the problem types we wish to solve are illustrated at the top of the figure. To solve an instance of one of these problem types, we transform it to an instance of the *Rich Pickup and Delivery Problem with Time Windows*, as illustrated by the arrows from the top row to the next row. Transformations are discussed in Section 4. The PDPTW instance is solved by a heuristic which will be presented in Section 5; this produces a PDPTW solution that finally is interpreted as a solution to the original problem. This solution framework has been tested on 338 benchmarks problems proposed in the literature. The results of this computational test are reported in Section 6. The paper is finally concluded in Section 7.

## 2 The Pickup and delivery problem with time windows (PDPTW)

Before starting to discuss the various variants of the VRPB we introduce the *Rich Pickup and Delivery Problem with Time Windows* (Rich PDPTW). All considered variants of the VRPB can be seen as extensions of the PDPTW. IP models of the PDPTW can be found in Desaulniers et al. [8] and Sigurd et al. [34], for our purpose we will only give a verbal description of our problem which differs slightly from the problems in the afore-mentioned papers.

In the Rich PDPTW we have *n requests* and *m vehicles*. A *request* $i \in \{1,...,n\}$ consists of picking up a quantity $l_i$ of goods at one location and delivering it to another location. With each request is associated a *pickup time window*, a *delivery time window*, and two *service times* $s_i^p$ and $s_i^d$ indicating how long the pickup and delivery operations take to perform. A vehicle is allowed to arrive at a location before the start of the time window, in which case it will have to wait before starting the corresponding operation. A vehicle may never arrive at a location after the end of the time window. Each request furthermore has an associated *pickup precedence number*, and a *delivery precedence number*. Each vehicle must visit the locations in nondecreasing order of precedence number (see e.g. Sigurd et al. [34] for various applications of precedence constraints).

Each request $i$ can only be served by a vehicle $k \in F_i$, where $F_i$ is the set of feasible vehicles corresponding to request $i$. Each vehicle $k \in \{1,\dots,m\}$ has an associated *capacity* $C_k$, a *start time* $b_k$ and *end time* $e_k$, and an

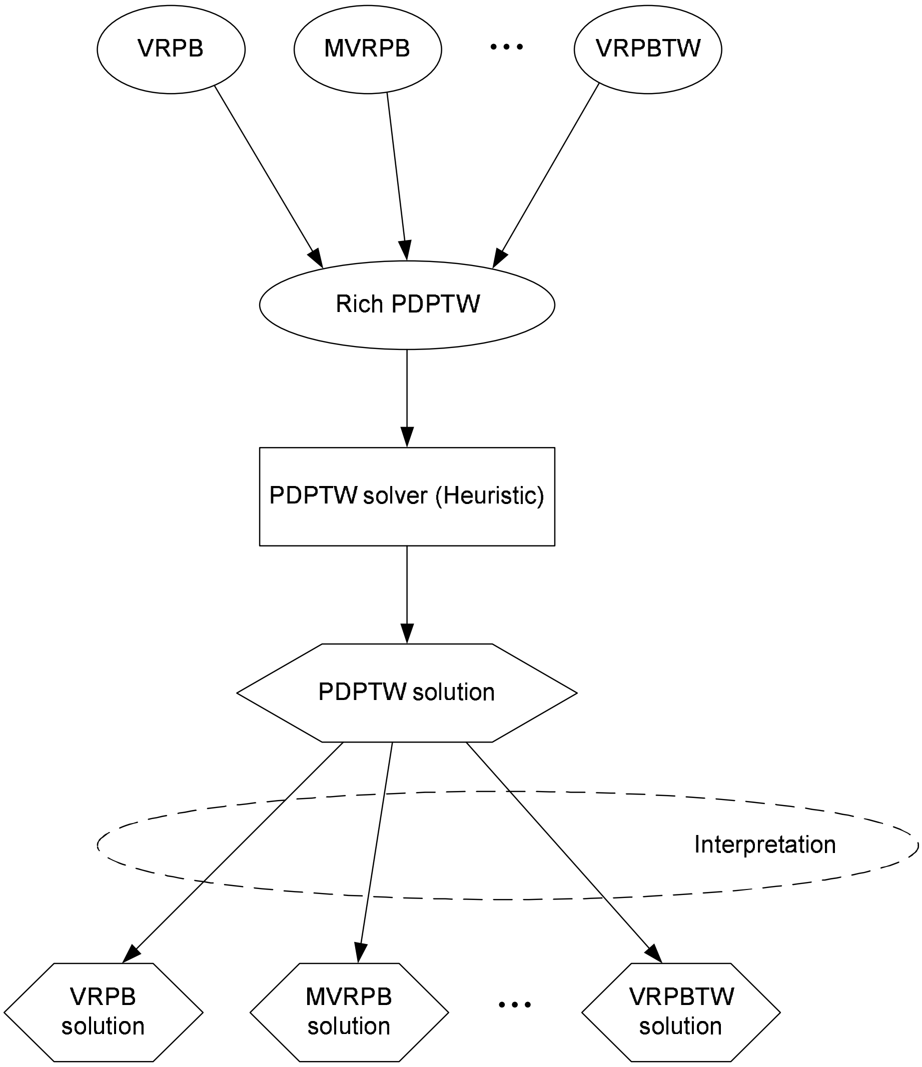


Figure 1: Solution framework: As described in Section 3 the algorithm accepts as input variants of the Vehicle Routing Problem with Backhauls, including: (VRPB), (MVRPB), (MDMVRPB), (VRPBTW), (MVRPBTW) and (VRPSDP). All of the problems are transformed to a Rich Pickup and Delivery Problem with Time Windows, which is solved heuristically through a Large Neighborhood Search algorithm. The last step of the algorithm transforms the obtained solution back to the original problem. The framework is not limited to backhaul models, but can be used to solve other types of vehicle routing problems, such as the vehicle routing problem with time windows or the capacitated vehicle routing problem.

associated *start terminal* $B_k$ and *end terminal* $E_k$ where it starts and ends its duty respectively. The vehicle must leave its start terminal at time $b_k$ even though this might introduce waiting time at the first customer visited. The vehicle must return to the end terminal at time $e_k$ or before.

The problem can be defined on a directed graph where the locations are represented by a set of *nodes* $V = \{1, ..., 2n+2m\}$, and for each *edge* $(i, j)$ we have an associated *distance* $d_{ij}$ and *travel time* $t_{ij}$, where we assume that travel times satisfy the triangle inequality while the only assumption on the distances is that they must be non-negative. The locations will often be referred to as *visits*.

The task is to construct a set of valid routes for a limited number of vehicles such that an associated *objective function* is minimized. The objective function is a weighted sum of 1) the sum of the distance traveled by the vehicles. 2) the number of requests not assigned to a vehicle. The two terms are weighted by the coefficients $\alpha$ and $\beta$. Notice that this objective function does not necessarily assign all requests to a vehicle. Requests not assigned to a vehicle are placed in a virtual *request bank*, which in a real world situation must be handled by a human dispatcher. Hence, normally a high value is assigned to the coefficient $\beta$ to stimulate that as many requests as possible are to be serviced. In the experiments performed in this paper, $\beta$ was chosen sufficiently high to avoid situations were some requests where left in the request bank upon termination.

# 3 Overview of vehicle routing problems with backhauls

This section gives an overview of the vehicle routing problems with backhauls proposed in the literature. We restrict ourselves to multi-vehicle problems. Single-vehicle problems have been studied by for example Gendreau et al. [14], Ghaziri and Osman [15] and Süral and Bookbinder [36].

## 3.1 The Vehicle Routing Problem with Backhauls (VRPB)

In the *vehicle routing problem with backhauls* (VRPB) we wish to minimize the total traveled distance and we are allowed to serve linehaul and backhaul customers on the same routes subject to the following limitations.

(A) If a route contains both linehaul and backhaul customers then the backhaul customers must be served after the linehaul customers.
(B) A route is not allowed to consist entirely of backhaul customers.
(C) The capacity of the vehicle should be obeyed, that is, neither the sum of the demands of the linehaul customers nor the sum of the demands of the backhaul customers served by a vehicle may exceed the vehicle capacity.
(D) The number of vehicles to use is given in advance. This means that even if it is possible to find better solutions using fewer or more vehicles, we must report the best solution we can find that uses the specified number of vehicles.
(E) All customers are serviced from a single depot.
(F) All vehicles have the same capacity.

Constraint (A) might seem artificial but it is justified by the fact that many vehicles are rear-loaded. This makes it problematic to try to load the vehicle with goods heading for the depot before we have delivered all goods to the customers as the pickup goods might block access to the delivery goods. The constraint is also justified by the fact that the linehaul customers frequently prefer early deliveries while backhaul customers prefer late pickups.

A recent survey of the VRPB was presented by Toth and Vigo [42]. Exact methods for the VRPB are proposed by Mingozzi et al. [26] and Toth and Vigo [41]. Heuristics have been developed by Anily [3], Casco et al. [5], Crispim and Brandao [7], Goetschalckx and Jacobs-Blecha [16], [22] and Toth and Vigo [40].

## 3.2 The Mixed Vehicle Routing Problem with Backhauls (MVRPB)

The *Mixed Vehicle Routing Problem with Backhauls* (MVRPB) is derived from the VRPB by relaxing limitations (A), (B) and (D). That is, we can mix linehaul and backhaul customers freely within a route and we are free to use as many vehicles as we want. We still have to obey the capacity limit of the vehicles. The capacity check is slightly more complicated in the MVRPB problem as the vehicle load fluctuates during the route. Furthermore, some MVRPB also have a duration limit that implies that routes should be completed within a certain time frame; for such problems the travel time between customers and the service time at the customers is given.

The name *Vehicle Routing Problem with Pickups and Deliveries* (VRPPD) is sometimes used instead of MVRPB. Heuristics for this problem are presented by Halse [19], Nagy and Salhi [27], [32] and Wade and Salhi [43], [44].

## 3.3 The Multiple Depot Mixed Vehicle Routing Problem with Backhauls (MDMVRPB)

The *Multiple Depot Mixed Vehicle Routing Problem with Backhauls* (MDMVRPB) is a generalization of the MVRPB. In the MDVRPB limitation (E) is relaxed such that we instead of just considering a single depot are faced with problems where several depots are present. At each depot a limited fleet of vehicles is available, and a vehicle should start and end its duty at the same depot. Heuristics for the problem are proposed by Nagy and Salhi [27], [32]. They denoted the problem the *Multi Depot Vehicle Routing Problem with Pickup and Deliveries*.

## 3.4 The Vehicle Routing Problem with Backhauls and Time Windows (VRPBTW)

The *Vehicle Routing Problem with Backhauls and Time Windows* (VRPBTW) extends VRPB by assigning a time window to each customer, by having travel times associated with each pair of locations, and by having service times associated with the customers. Visits at a customer should start within the time window. If the vehicle arrives too early at a customer it has to wait until the start of the time window. If the vehicle arrives too late the route is invalid. Limitations (B) and (D) from the VRPB are relaxed in the VRPBTW. The objective of VRPBTW is either to minimize the total traveled distance or to minimize the number of vehicles as the first priority and then minimize the total traveled distance as the second priority.

An exact algorithm for the VRPBTW based on column generation was proposed by Gelinas et al. [13], and heuristics were proposed by Duhamel et al. [12], Hasama et al. [20], Reimann et al. [30], Thangiah et al. [38] and Zhong and Cole [48].

## 3.5 The Mixed Vehicle Routing Problem with Backhauls and Time Windows (MVRPBTW)

The *Mixed Vehicle Routing Problem with Backhauls and Time Windows* (MVRPBTW) is derived from VRPBTW by relaxing limitation (A) saying that backhaul customers should be visited after linehaul customers. The objective that has been considered in the literature is to minimize the number of vehicles as the first priority and the distance traveled as the second priority. Two heuristics have been proposed in Kontoravdis and Bard [23] and Zhong and Cole [48].

## 3.6 The Vehicle Routing Problem with Simultaneous Deliveries and Pickups (VRPSDP)

In the *Vehicle Routing Problem with Simultaneous Deliveries and Pickups* (VRPSDP) a subset of the customers simultaneously demand goods from—and supply goods to—the depot, and thus both a delivery and a pickup should occur at these customers. The pickup and delivery should be performed simultaneously such that each customer is visited only once by a vehicle. Unloading is obviously done before loading at these customers. The

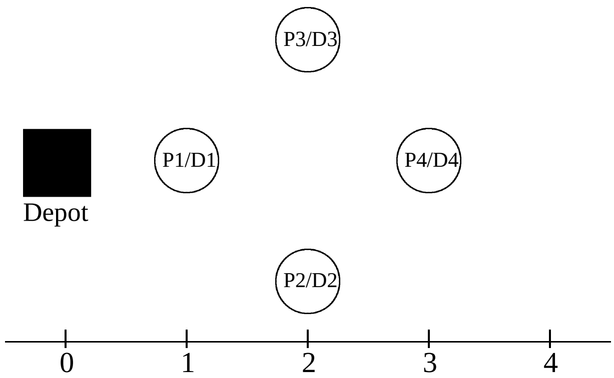


Figure 2: An example showing that simultaneous pickup and delivery at customers may increase the overall route lengths. The four customers have pickup/delivery requests of 2/2, 1/2, 1/2, 2/0 respectively. The vehicle has a capacity $C$ of 6 units, and normal Euclidean distances are used. In a MVRPB setting, the shortest route is D1, P2/D2, P4/D4, P3/D3, P1 of total length 7.66. If simultaneous pickup and deliveries are demanded, the shortest route becomes P3/P3, P2/D2, P4/D4, P1/D1 of total length 8.65.

simultaneous pickup and delivery operation decreases the customers' expenses or inconvenience associated with handling vehicles, but may result in longer routes as illustrated in Figure 2.

This problem was first introduced by Min [25] in the context of transportation material between public libraries and a library administration center (acting as a depot). Halse [19] presented exact and heuristic methods for the problem and Dethloff [9], [10] considered heuristic algorithms. Nagy and Salhi [32] used their MVRPB heuristic to solve the problem, but apparently the "simultaneous" constraint is not handled by the heuristic. This is discussed in further detail by Dethloff [10]. Two variants of the problem have been proposed recently. Nagy and Salhi [32] introduce a multi depot version of the problem, while Angelelli and Mansini [2] solve a version with time windows to optimality using column generation. The heuristic proposed in the present paper is not tested on the two last problem types although the underlying PDPTW model without modifications could handle these problem classes also.

### 3.7 Other backhauling problems

Wade and Salhi [45] introduce a problem that generalizes VRPB and MVRPB. In this problem one is not allowed to mix linehaul and backhaul customers on a route freely. A vehicle can only start to serve backhaul customers after a certain percentage of the linehaul load has been delivered. If this percentage is set to 0% then we get the MVRPB and if the percentage is set to 100% then we get the VRPB. Percentages in between 0% and 100% result in a blend between VRPB and MVRPB.

Halskau et al. [17] propose a backhauling problem with so called *lasso* tours. In their problem most customers require both a pickup and a delivery. At the first few customers visited on a route a delivery is performed to free up some room in the vehicle, at the customers in the middle of the route, the delivery and pickup operation is performed simultaneously. The tour is ended by visiting the first couple of customers again, this time in the reverse order to perform the omitted pickups. This creates a tour that looks like a lasso, as the first customers that are visited twice form the spoke of the lasso, while the customers that are visited once form the loop of the lasso.

These two problem variants cannot be solved by the heuristic presented in this paper in its present form. It would only require minor modifications to the heuristic and the underlying model to be able to solve these problems though.

# 4 Problem transformations

This section describes how each of the problems discussed in Section 3.1–3.6 can be transformed to a Rich PDPTW. The basic transformation is to represent a linehaul customer by a request with a pickup at the depot and a delivery at the linehaul customer. Backhaul customers are represented by a request with a pickup at the backhaul customer and a delivery at the depot. This transformation might seem sufficient to represent the MVRPB but it has the flaw that it allows a vehicle to go back to the depot for re-stocking or offloading and afterwards continue its duty. This is not allowed in a standard MVRPB. The problem is easily solved by assigning precedences to the different tasks: pickups at the depot get precedence 1, deliveries at linehaul customers and pickups at backhaul customers get precedence 2 and deliveries at the depot get precedence 3.

The backhaul after linehaul constraint (A) found in VRPB is also easily modeled using precedences. Instead of giving linehaul and backhaul customers identical precedences, we assign precedence 2 to the linehaul deliveries, precedence 3 to the backhaul pickups and precedence 4 to the deliveries at the depot.

In the VRPB we have to use a specified number of vehicles as stated by constraint (D). Our model only allows us to set an upper bound on the number of vehicles, so we need to model a vehicle equality constraint. This is done by modifying the distance matrix by setting the distance from the start terminal to the end terminal of each vehicle to $M$, where $M$ is a sufficiently large number. This forces the heuristic towards solutions with at least one request on each route in order to avoid the penalty $M$.

The VRPB constraint (B) saying that no route can consist of backhauls only, is handled in a similar way. Here we add the penalty $M$ to the cost of each edge from a start terminal to one of the backhaul pickup locations. This drives the heuristic towards solutions where such edges are not used, which means that at least one linehaul customer is served before a backhaul customer.

The simultaneous delivery and pickup constraint in VRPSDP is also modeled using penalties. As before, the delivery to a customer is modeled by a request from the depot to the customer and a pickup at a customer is modeled as a request going from the customer to the depot. In order to ensure that the delivery and pickup occur "simultaneously" we modify the distance matrix. The distance from a delivery visit to the simultaneous pickup visit is set to zero, while the distances from the pickup to all other visits are increased by the penalty term $M$. This forces the heuristic to visit the simultaneous pickup after a delivery. The situation is illustrated on Figure 3.

The multiple depots in the MDMVRPB are harder to model even though the underlying PDPTW model already supports multiple depots. The problem is that we until now have modeled a linehaul customer by a pickup at the depot and a delivery at the customer, and vice-versa for the backhaul customers. In the multi depot problems we cannot assign a request to a given depot in advance as we do not know where the pickup of a linehaul request or the delivery of a backhaul request should occur. To model this kind of constraint we do the following. For each vehicle in the problem (remember that in the MDMVRPB a fixed number of vehicles is available in each depot) we create a dummy request with pickup and delivery locations at the depot of the vehicle. There is no demand associated with the dummy requests. A dummy request should only be served by the vehicle it is designed for, which is ensured by letting its feasible set of vehicles $F_i$ contain that one vehicle only. We still represent each linehaul customer by one request. All pickups of these linehaul requests take place at a virtual depot. All distances to and from the virtual depot are set to zero. Backhaul customers are represented in the same way — by a pickup at the backhaul customer and a delivery at the virtual depot. The idea is that linehaul requests should travel via the dummy pickup location and backhaul requests should travel via the dummy delivery location. This is ensured using precedences: Linehaul pickups get precedence 1, pickups of the dummy requests get precedence 2, linehaul deliveries and backhaul pickups get precedence 3, deliveries of the dummy requests get precedence 4 and linehaul deliveries get precedence 5. This forces the dummy request to "surround" the linehaul deliveries and backhaul pickups such that the distance to and from the right depot is used. Figure 4 shows an example of a MDMVRPB

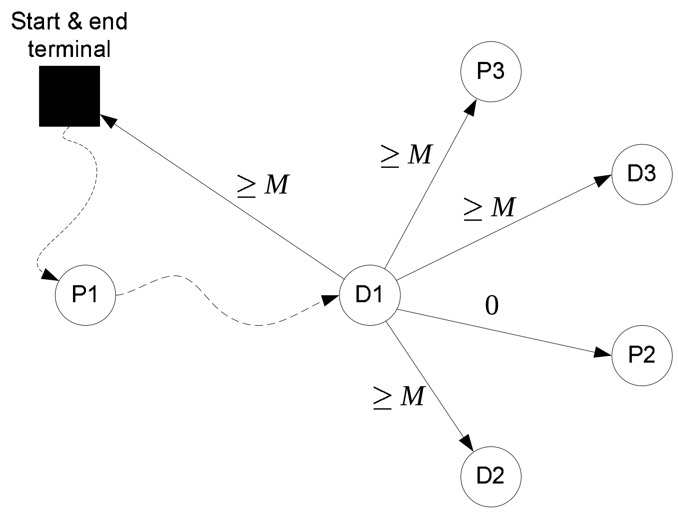


Figure 3: Modeling of simultaneous delivery and pickup. Request 1 is a delivery to a customer, request 2 represents the simultaneous pickup at the same customer and request 3 is another unrelated request. The names "P*x*" denotes "the pickup of request *x*" and "D*x*" denotes "the delivery of request *x*". Edge weights are the distances $d_{ij}$. In order to ensure that D1 is followed by P2 we increase all other distances from D1 with $M$, while the distance from D1 to P2 is set to zero. In this way, the algorithm will first visit the pickup site of request 1 (the depot) and then travel to the delivery site of request 1 (the customer site). We might perform other visits along the dashed edges. After performing the delivery of request 1, only one edge has cost less than $M$, hence we go to P2 which is the simultaneous pickup.

route with two linehaul customers and one backhaul customer.

A remark should be made about penalty based modeling: If a feasible solution exists that does not violate any of the constraints, the optimal solution will not contain any of the penalty terms. However, since we use heuristics for solving the model, we may end up with a solution which still contains some penalties. This can easily be detected by inspecting the objective value and the heuristic can either be repeated (hoping that a second run will find a better solution) or some manual adjustment of the data may be needed, e.g. by increasing the number of vehicles or by removing some customers which cannot be handled. It should, however, be pointed out that the heuristic has never produced any infeasible solutions during the computational experiments performed in Section 6.

We made heavy use of precedences in the transformations described above. The precedences can also be used to speed up the heuristic when faced with the problem types described in this paper. Consider for example the

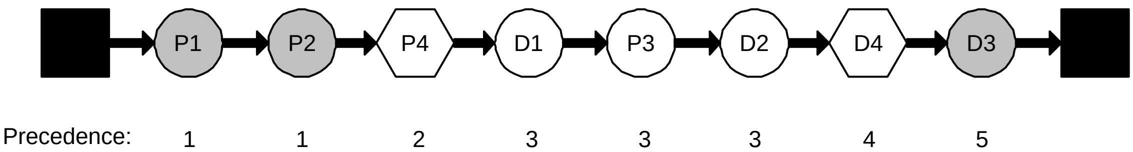


Figure 4: An example of a MDMVRPB route with two linehaul customers and one backhaul customer. The linehaul customers are represented by request 1 and 2 and the backhaul customer is represented by request 3. Request 4 is the dummy request. The start and end terminals are represented by squares, the visits of the normal requests are represented by circles and the visits of the dummy request are represented by hexagons. Pickups and deliveries at the depot are shown in grey and the precedence of the visits is displayed underneath the route. One can observe that the actual MDMVRPB route can be inspected by looking at the white visits; here the hexagons should be viewed as depot visits and the normal deliveries and pickups correspond to the linehaul and backhaul customers respectively.

MVRPB where several pickup and deliveries occur at the depot and all permutations of the pickups at the depot within a route are feasible and equally good as long as the deliveries stay fixed (and similarly for the backhaul deliveries). We can use precedences to create an ordering on the pickups and deliveries at the depot such that only one permutation is valid. We enumerate the request from 1 to $n$. If request $i$ involves a pickup at the depot, then this pickup gets precedence $i$, if request $i$ involves a delivery at the depot then this delivery gets precedence $i+n+2$. Pickups and deliveries that corresponds to visits at the customers gets precedence $n+1$. The same idea can be used for the five other problems as well.

## 5 Solution methods

Recent work on local search methods indicate that larger neighborhoods may be needed to solve some difficult optimization problems as shown by e.g. Ahuja et al. [1]. Due to the size of the neighborhoods, various heuristics are generally used to search the neighborhood in order to keep the time complexity at a reasonable level. This means, that the performance of a local search algorithm is limited by the quality of the heuristic that searches the neighborhood. To work around this bottleneck, Ropke [31] proposed to use several heuristics to search the neighborhood, where the frequency of using each heuristic is based on some empirical evidence from the search. An extended version of this heuristic is used to solve our PDPTW model.

The heuristic is based on *Large Neighborhood Search* (LNS) as proposed by Shaw [35] and it has similarities with the *Ruin and Recreate* (R&R) framework proposed by Schrimpf et al. [33]. Our heuristic repeatedly runs through the following steps:

**LNS iteration**

1 Choose a removal heuristic $R$ and an insertion heuristic $I$.
2 Remove a number $q$ of requests from the routes using heuristic $R$.
3 Insert the free requests into the existing routes using heuristic $I$.
4 Evaluate the objective function of the new solution.
5 If the objective function is improved, accept the new solution. Otherwise accept the new solution with a probability that depends on the increase of the objective function.

The heuristic differs from the ordinary LNS and R&R methods by incorporating several large-neighborhood heuristics, which are applied with a variable frequency controlled by a learning layer. Each insertion or removal heuristic in the LNS heuristic may have various properties. Some heuristics are used to *intensify* the search while other heuristics mainly play the role of *diversifying* the search. In this way, the learning layer not only distributes CPU-time among the various heuristics involved, but also controls the intensification or diversification of the search based on empirical information. This can be seen as an extension of the tabu search methods described by Hertz et al. [21]. One may also see the LNS algorithm as a variant of *Variable Neighborhood Search* (VNS) described by Hansen and Mladenovic [18], the main difference being that VNS operates on one type of neighborhood with variable depth, while LNS operates with structurally different neighborhoods.

In the PDPTW heuristic the removal heuristic $R$ removes up to 40% of the requests in each iteration. This enables the heuristic to make significant changes to the current solution in a single iteration. We use six different removal heuristics in our LNS heuristic; each removal heuristic has its own strategy for choosing the requests to remove. The heuristics are:

- *Random removal*: The requests are chosen at random.
- *Shaw removal*: Remove related requests, i.e. requests that are geographically close to each other (Shaw [35]).

- *Worst request removal*: Remove the request whose removal decreases the cost function the most.
- *Cluster removal*: Attempt to partition the nodes into subsets so that the nodes in each subset are somehow "close to each other". For a more detailed description of this removal heuristics see Section 5.3.
- *History based removal*: This heuristic makes use of historical information when removing requests. Two variants of this heuristic have been considered as will be described in Sections 5.4 and 5.5.

The first three removal heuristics have been used previously [31] while the three last are new.

In order to insert the requests we use the five insertion heuristics proposed by Ropke [31]. The heuristics can be divided into two classes:

- *Basic insertion heuristics*: which are similar to the insertion heuristic of Solomon [37]. In each iteration a request is inserted into the solution such that the cost function is increased the least possible.
- *Regret insertion heuristics*: which are similar to heuristics proposed by Potvin and Rousseau [29] and Tillman and Cain [39]. In each iteration of the standard version of the heuristic a request is inserted so as to maximize the gap in the cost function between inserting the request into its best route and its second best route.

The insertion heuristics are described in more details in [31].

In each step of the PDPTW heuristic one removal and one insertion heuristic are used. Computational experiments have shown that in order to reach high-quality solutions all removal and insertion heuristics are necessary, but their contribution to the solution process may vary during the search.

The *monitoring and learning* layer observes how often a given removal or insertion heuristic contributes to a new, accepted solution, and increases the probability of choosing the given heuristic according to its success. This is done using *roulette wheel selection* where each heuristic has a probability corresponding to its success-rate. In order to ensure that statistical information is collected for all heuristics throughout the search, each heuristic is used not less than a given lower limit.

The LNS algorithm is basically a local search algorithm, and hence it can be combined with most state-of-art local search paradigms. Using the *simulated annealing* paradigm, we evaluate the cost function after each LNS step. If the cost has decreased or is unchanged, the new solution is always accepted. If the cost has increased, the solution is randomly accepted with a probability exponentially decreasing with the increase of the cost.

### 5.1 Measuring the distance between two requests

In the removal heuristics we need a measure for the distance $d(r_1,r_2)$ between two requests $r_1$ and $r_2$. Ropke [31] used the following expression: $d(r_1,r_2) = d_{a_1,a_2} + d_{b_1,b_2}$ where $a_1$ and $a_2$ are the pickups of the requests and $b_1$ and $b_2$ are the deliveries. This works fine for the pure PDPTW problems but the definition is problematic for backhaul problems. Consider for example two requests corresponding to a linehaul and a backhaul customer located far from the depot. Using the old distance function, the distance between these two requests would be large even though the linehaul and backhaul customer are located close to each another. Instead we use $d(r_1,r_2) = \frac{1}{4}(d_{a_1,a_2} + d_{a_1,b_2} + d_{b_1,a_2} + d_{b_1,b_2})$. If a pickup or a delivery is located at the depot then the distances involving this visit are removed from the formula and the denominator is decremented accordingly.

### 5.2 Simplified Shaw removal

Shaw [35] defines a removal method that removes related requests. Ropke [31] defines the relatedness between two requests in terms of the distance between the two requests, their capacity demands, temporal information and information about which vehicles can serve the requests. In this paper we take a simpler approach as we define the relatedness between two requests solely by the distance $d(r_1, r_2)$ between the requests.

### 5.3 Cluster removal

Given a set of points in the plane we can ask to partition the set into $k \geq 2$ disjoint subsets such that the points within each subset are close together with respect to the distance $d(r_1, r_2)$. We say that we partition the points into $k$ clusters.

A heuristic for finding such a partition can be constructed by modifying Kruskal's algorithm [24] for the minimum spanning tree problem. Instead of running Kruskal's algorithm to the end, it can be stopped when $k$ connected components are left. These connected components are our approximation of the desired clusters.

The clustering algorithm is used in a removal heuristic as follows. First a route is selected at random. Then the requests on this route are partitioned into two clusters. One of these clusters is chosen at random and the requests from the chosen cluster are removed. If we need to remove more requests then we pick one of the removed requests and find a request that is close to the chosen request. The new request should come from a route that has not been touched by removals in the current iteration. The route of the new request is partitioned into two clusters and so the process continues until the desired number of requests has been removed. The motivation for the heuristic is to remove large chunks of related requests from a few routes instead of removing a few requests from each route. Figure 5 illustrates when the cluster removal heuristic can be useful.

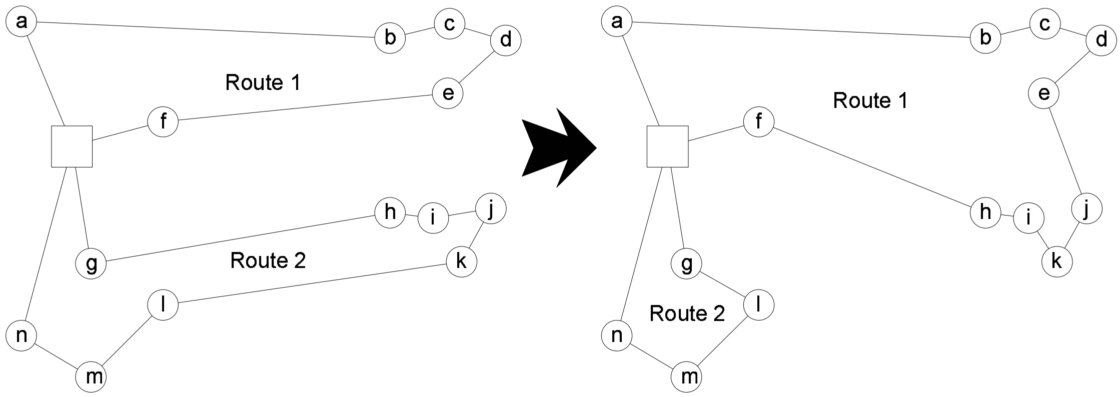


Figure 5: Cluster Removal example: The circles mark the delivery locations, all pickups take place at the depot (marked by the square). In the figure to the left we have a suboptimal solution and we would like to move to the solution shown in the right part of the figure where requests $h$-$k$ are placed on the same route as requests $a$-$f$. To reach this solution we need to remove requests $h$, $i$, $j$ and $k$ at once. If just one of the requests $h$, $i$, $j$ or $k$ is left on route 2 then the insertion heuristics most likely are going to insert the rest of the requests back into route 2. The removal heuristics presented so far may not be able to remove all of the requests at once, but the cluster removal heuristic does just that. The result of applying the clustering algorithm on route 2 would be the two clusters $g, l, m, n$ and $h, i, j, k$ and the last cluster would be removed with probability 0.5.

### 5.4 Neighbor graph removal

None of the removal heuristics proposed so far have made any use of historical information when removing requests. The decision about which requests to remove has been made solely by using the information available in the current state.

The *neighbor graph removal heuristic* uses both historical information and the current state to select the requests to remove. The historical information is stored in a complete, directed, weighted graph called the *neighbor graph*. The graph contains a node for each visit in the problem. The weight of all edges is initially set to plus infinity. The weight of an edge $(a,b)$ stores the cost of the best solution encountered so far in which the visit corresponding to $a$ is performed just before the visit corresponding to $b$. Each time a new solution is discovered during the search, the edge weights in the graph are updated if necessary.

The graph is used to remove requests that seem to be placed in an unsuitable place. When the removal heuristic is invoked it calculates a score for each request in the current solution. The score is calculated by summing the edge weights in the neighbor graph corresponding to the neighbor configuration in the current solution. The requests with high scores seem to be misplaced and are removed. Every time a request has been removed the scores of the surrounding requests are recalculated. Some randomness is introduced in the removal process in order to avoid removing the same requests over and over again. Specifically the randomness ensures that we sometimes do not remove the requests with the highest score but instead remove some with slightly lower scores.

### 5.5 Request graph removal

In the request graph removal heuristic we store historical information in a graph called the *request graph*. This graph is complete and undirected and each node in the graph corresponds to a request in the PDPTW problem. The weight of an edge $(a,b)$ denotes the number of times the two requests corresponding to $a$ and $b$ have been served by the same vehicle in the $t$ best unique solutions observed so far in the search. The weights of all edges are initially set to zero, and in all experiments the parameter $t$ was set to 100.

This graph could be used in a similar fashion as the graph described in Section 5.4. That is, we could examine all planned requests $r$ and calculate the score

$$score(r) = \sum_{i \in R(r), i \neq r} w_{ri}$$

where $R(r)$ is the set of requests in the route containing $r$ and $w_{ri}$ is the weight of the edge between $r$ and $i$ in the requests graph. A request with a low score is situated in an unsuitable route according to the request graph and should be removed. Our initial experiments indicated that this was an unpromising approach, probably because it strongly counteracts the diversification mechanisms in the LNS heuristic.

Instead, the graph is used to define the relatedness between two requests, such that two requests are considered to be related if the weight of the corresponding edge in the request graph is high. This relatedness measure is used as in the removal heuristic proposed by Shaw [35], mentioned in Section 5.2.

## 6 Computational experiments

### 6.1 Parameter tuning

Even though the proposed heuristic is controlled by quite a few parameters, we have tried to keep the parameter tuning to a minimum in this paper. This is achieved by using the same parameters that were found in the parameter tuning performed by Ropke [31], where applicable. The only parameters that have been tuned are the two parameters that control the simulated annealing: *the cooling rate* $c$ and the *start temperature control parameter* $w$. After each LNS iteration the temperature $T$ is updated using the recursion $T := cT$. The parameter $w$ controls the start temperature $T_0$. In order to set the start temperature $T_0$ we use an estimate of the objective value of a reasonable solution to the problem. This estimate is found by obtaining an initial solution using one of our insertion heuristics and calculating the *modified objective value* $z'$ of this solution. The modified objective value is obtained by setting the coefficient $\beta$ to zero, such that unplanned requests do not make the estimate of the objective value unreasonably high. Now the start temperature is set such that a solution that is $1+w$ times larger than $z'$ is accepted with probability 0.5 when the current solution has objective $z'$. We have tested the algorithm on 11 problems chosen from 5 of the 6 problem categories. The configuration $w = 0.05$ and $c = 0.9998$ proved to be the best among the 30 configurations tested. The same parameters were used for all problem types considered in the following sections.

## 6.2 Test strategy

The LNS heuristic is tested on 9 data sets proposed in the literature. The test serves two major purposes. The first purpose is to compare three configurations of the LNS heuristic against each other. The three configurations are:

- A configuration similar to the one used by Ropke [31]. This configuration benefits from the learning layer but is limited to the 3 "old" removal heuristics: The *simplified Shaw removal*, the *worst removal* and the *random request removal*. This configuration is denoted *standard* in the following.
- A configuration that uses all 6 removal heuristics but has disabled the learning layer. This implies that all removal and insertion heuristics are equally likely to be selected during the search. This configuration is denoted *6R - no learning* in the following (the "6R" indicates that 6 removal heuristics are in use).
- The last configuration is similar to the second, but in the third configuration the learning layer is activated again. The configuration is denoted *6R - normal learning*.

These three configurations allow us to see if the new removal heuristics improve the quality of the heuristic and enable us to judge the effectiveness of the learning layer.

The second major purpose of the test is to compare the solution quality obtained by the unified heuristic to the results obtained by more specialized heuristics proposed for the various problem types. We want to know whether a general heuristic can be competitive with specialized heuristics.

The stopping criterion employed is to stop when the heuristic has performed 25000 remove-insert iterations. Each configuration of the heuristic is applied 10 times to each problem instance. The reported computation times are, however, for a single run of the algorithm.

All problems considered in the following are geometric problems where distances and travel times are defined by the Euclidean distance, hence the triangle inequality is satisfied for both parameters. When it has been necessary to calculate distances from a set of coordinates we have used double precision calculations unless otherwise stated. For many of the problem classes we only present a summary of the experiments performed. We refer the reader to the appendix for the full tables for these problems. All experiments were performed on a Linux based PC, equipped with 256 MB RAM and a 1.5 GHz Pentium IV processor. The heuristic was implemented in C++.

## 6.3 The Vehicle Routing Problem with Backhauls (VRPB)

The first problem type we study is the symmetric VRPB. This problem along with the VRPBTW is probably the most studied of the backhaul problems. Two data sets are proposed in the literature, the first was proposed by Goetschalckx and Jacobs-Blecha [16] and contains 62 instances with between 20 and 150 customers. The second data set was proposed by Toth and Vigo [40] and contains 33 instances with between 21 and 100 customers. We denote the two data sets the *Goetschalckx* and the *Toth-Vigo* data sets respectively.

Comparing results on the Goetschalckx data set are a little problematic as at least 3 different rounding conventions have been used for calculating the distances between the customers in the data sets. We report our results obtained using 2 of the 3 rounding conventions and refer to the appendix for a discussion about the third rounding convention and the results obtained using it.

Currently the two best heuristics for the VRPB are probably the heuristic proposed by Toth and Vigo [40] and the heuristic by Osman and Wassan [28]. The heuristic by Toth and Vigo finds good solutions in a short time while the heuristic proposed by Osman and Wassan spends more time but on the overall finds better solutions. We compare our heuristic with the results found by Osman and Wassan as the running time of our algorithm is comparable to that of Osman and Wassan's heuristic. In order to calculate the distance between two customers, Osman and Wassan used floating point arithmetic, hence we do the same (using double precision) in the tests reported in Table 1.

The tests show that the configurations using all 6 removal heuristics are better than the one using only three removal heuristics. This test also shows that the configuration that does not include the learning layer overall is slightly better than the configuration including the learning layer, which is a bit surprising. All configurations of the LNS heuristics do better than Osman and Wassan's heuristic when looking at how many best known solutions the heuristics have found. It should be noted that the best solution found by Osman and Wassan's heuristic was found in 8 experiments, while we used 10 experiments for each LNS configuration. If one looks at the sum of the best solution costs identified by the heuristics, it is observed that the LNS heuristics overall only marginally improve the solutions found by Osman and Wassan's heuristic; for all LNS heuristics the improvement is within 0.1%. All together the LNS heuristics improved the solution of 26 of the 62 problem instances. Finally we see that the average solution costs found by the LNS heuristics are quite good as they on average are less than 0.5% from the best known solution costs.

Generally it is hard to compare the running time of our heuristic to that of the heuristics proposed in the literature, as the computational experiments have been performed on different computers. According to the Linpack benchmarks reports [11], our computer has a TPP rating (*Toward Peak Performance*) of 1311 MFlops while Osman and Wassan's Computer has a TPP rating of 25 MFlops, implying that our computer is around 53 times faster. The average time for solving one problem was between 69 and 73 seconds for the LNS heuristics. Osman and Wassan tested two versions of their heuristic, the fastest version using around 2800 seconds to solve one problem and the slower version using 4000 seconds. This corresponds to 52 and 75 seconds on our computer, which is very comparable to the time used by our algorithm. Hence our general heuristic is on par with Osman and Wassan's specialized heuristic both with respect to solution quality and solution times.

The second way to calculate the distances is to round them to one decimal, and store them as an integers using a fixed point representation. The final result is rounded to an integer. This type of rounding is used in the exact methods developed by Toth and Vigo [41] and Mingozzi et al. [26]. 34 of the 62 instances have been solved to optimality and a good solution is provided for 13 more problems without proving optimality. Table 2 summarizes the results obtained by applying the heuristic to these 47 problems (problem *A1*-*K4*) using the same rounding conventions as the exact methods. These results also show that the configurations that use the new removal heuristics are better than the one that only uses the 3 old removal heuristics. This time the configurations with and without the learning layer are virtually equally good. All configurations find 28 optimal solutions out of the 34 optimal solutions reported by Toth and Vigo [41] and Mingozzi et al. [26]. Eight new best solutions were found in the tests.

The *Toth-Vigo* data set have been approached by the exact methods of Toth and Vigo [41] and Mingozzi et al. [26] and by the heuristics of Crispim and Brandao [7], Osman and Wassan [28] and Toth and Vigo [40]. Table 3 reports the results found by the LNS heuristic compared with the best known results from the literature. We see that the configuration with learning enabled provides the best solutions on the average; furthermore it is the only one which identifies all known optimal solutions. The configuration without learning overall finds slightly better solutions compared to the learning version when summing the best solution from the ten experiments. The LNS heuristics improve the best known solutions to 5 of the problems.

A class of asymmetric problem instances was proposed by Toth and Vigo [41], but we have not included this data set in our test even though our PDPTW model would be able to handle the asymmetric problems.

### 6.4 The Mixed Vehicle Routing Problem with Backhauls (MVRPB)

Two data sets have been proposed for the MVRPB. The first set is based on a relaxed version of the *Goetschalckx* problems, and it has been studied by Halse [19] and Wade and Salhi [43], [44]. The other data set, which was proposed by Nagy and Salhi [27], is constructed by transforming 14 well-known CVRP instances into MVRPB instances. Three MVRPB instances are constructed from each CVRP instance, having 10%, 25% and 50% of the customers transformed to backhaul customers. Heuristics are applied to the last data set by Dethloff [9] and Nagy and Salhi [27], [32]. We decided to test our heuristic on MVRPB by using the last data set.

The chosen data set contains 42 problems with 50 to 199 customers. Table 4 compares the solutions obtained by the LNS heuristics to the solutions obtained by Nagy and Salhi. Unfortunately it is not possible to include the results obtained by Dethloff [9] in the table as Dethloff only tested his algorithm on a subset of the problems. The heuristic named NS1 in the table is a construction algorithm and the heuristic named NS2 is a construction heuristic followed by an improvement algorithm. Both are much faster than the LNS heuristics. The comparison shows that great improvements can be achieved by using a more advanced heuristics such as the LNS heuristic proposed here, as we get results that are more than 10% better than those obtained by the simpler heuristics. We succeeded in improving the best known solution for 41 out of the 42 problems. On the last problem we matched the solution reported by Nagy and Salhi. Notice that the average solution cost decreases when more customers are turned into backhaul customers in the solutions provided by the LNS heuristic. This is expected as a greater percentage of backhaul customers leads to greater flexibility in the planning as long as the percentage of backhaul customers is not greater than 50%. It is worth noting that Nagy and Salhi's results do not show this behavior.

Table 5 compares the three LNS configurations. The results show that the configurations with six removal heuristics overall are better than the one with three removal heuristics when one compares the gaps. The results also show that the configuration with the learning layer enabled is better than the one without the learning layer. One can also notice that the computation time increases as more customers are turned into backhaul customers. This behavior can most likely be explained by the fact that routes in general contain many customers when the percentage of backhauls customers is around 50%. Long routes imply that more time is spent in the insertion heuristics.

### 6.5 The Multiple Depot Mixed Vehicle Routing Problem with Backhauls (MDMVRPB)

Only one data set has been proposed for the MDMVRPB. This data set was proposed by Nagy and Salhi [32] and is constructed from Gillett and Johnson's 11 multi depot vehicle routing problems. Each of the 11 problems are turned into three MDMVRPB problems by creating problems with 10%, 25% and 50% backhaul customers; thus the MDMVRPB data set contains 33 problems with between 50 and 249 customers. The only heuristics that have been applied to the problems so far are those by Nagy and Salhi which also were used for the MVRPB discussed in Section 6.4.

In Table 6 we compare the results obtained by the LNS heuristic with those obtained by the best heuristics of Nagy and Salhi [27], [32]. It has been necessary to reconstruct the problems from Gillett and Johnson's original problems following the description in [32], as the original problems no longer were available from the authors. We believe that the problems have been constructed properly. The reconstructed problems have been made available on the web [46] for future comparisons. Again, we observe that the LNS heuristic offers huge improvements over

| | Best known | | Standard | | | | 6R - no learning | | | | 6R - normal learning | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | n | cost | avg. sol. | best sol. | avg. gap (%) | avg. time (s) | avg. sol. | best sol. | avg. gap (%) | avg. time (s) | avg. sol. | best sol. | avg. gap (%) | avg. time (s) |
| A1 | 25 | **229885.65** | 229885.65 | **229885.65** | 0.00 | 7 | 229885.65 | **229885.65** | 0.00 | 7 | 229885.65 | **229885.65** | 0.00 | 7 |
| A2 | 25 | **180119.21** | 180119.21 | **180119.21** | 0.00 | 8 | 180119.21 | **180119.21** | 0.00 | 8 | 180119.21 | **180119.21** | 0.00 | 8 |
| A3 | 25 | **163405.38** | 163405.38 | **163405.38** | 0.00 | 9 | 163405.38 | **163405.38** | 0.00 | 10 | 163405.38 | **163405.38** | 0.00 | 9 |
| A4 | 25 | **155796.41** | 155796.41 | **155796.41** | 0.00 | 10 | 155796.41 | **155796.41** | 0.00 | 10 | 155796.41 | **155796.41** | 0.00 | 11 |
| B1 | 30 | **239080.15** | 239080.16 | 239080.16 | 0.00 | 9 | 239080.16 | 239080.16 | 0.00 | 9 | 239080.16 | 239080.16 | 0.00 | 9 |
| B2 | 30 | **198047.77** | 198047.77 | **198047.77** | 0.00 | 10 | 198047.77 | **198047.77** | 0.00 | 10 | 198047.77 | **198047.77** | 0.00 | 10 |
| B3 | 30 | **169372.29** | 169372.29 | **169372.29** | 0.00 | 13 | 169372.29 | **169372.29** | 0.00 | 14 | 169372.29 | **169372.29** | 0.00 | 14 |
| C1 | 40 | **250556.77** | 250846.82 | **250556.77** | 0.12 | 14 | 250560.15 | **250556.77** | 0.00 | 14 | 250556.77 | **250556.77** | 0.00 | 13 |
| C2 | 40 | **215020.23** | 215020.23 | **215020.23** | 0.00 | 16 | 215020.23 | **215020.23** | 0.00 | 16 | 215020.23 | **215020.23** | 0.00 | 16 |
| C3 | 40 | **199345.96** | 199345.96 | **199345.96** | 0.00 | 18 | 199345.96 | **199345.96** | 0.00 | 20 | 199345.96 | **199345.96** | 0.00 | 18 |
| C4 | 40 | **195366.63** | 195366.63 | **195366.63** | 0.00 | 19 | 195366.63 | **195366.63** | 0.00 | 19 | 195366.63 | **195366.63** | 0.00 | 19 |
| D1 | 38 | **322530.13** | 322530.13 | **322530.13** | 0.00 | 12 | 322530.13 | **322530.13** | 0.00 | 12 | 322530.13 | **322530.13** | 0.00 | 12 |
| D2 | 38 | **316708.86** | 316708.86 | **316708.86** | 0.00 | 11 | 316708.86 | **316708.86** | 0.00 | 12 | 316708.86 | **316708.86** | 0.00 | 12 |
| D3 | 38 | **239478.63** | 239478.63 | **239478.63** | 0.00 | 13 | 239478.63 | **239478.63** | 0.00 | 13 | 239478.63 | **239478.63** | 0.00 | 13 |
| D4 | 38 | **205831.94** | 205831.94 | **205831.94** | 0.00 | 16 | 205831.94 | **205831.94** | 0.00 | 16 | 205831.94 | **205831.94** | 0.00 | 15 |
| E1 | 45 | **238879.58** | 238879.58 | **238879.58** | 0.00 | 18 | 238879.58 | **238879.58** | 0.00 | 18 | 238879.58 | **238879.58** | 0.00 | 18 |
| E2 | 45 | **212263.11** | 212463.34 | **212263.11** | 0.09 | 23 | 212263.11 | **212263.11** | 0.00 | 23 | 212458.75 | **212263.11** | 0.09 | 22 |
| E3 | 45 | **206659.17** | 206710.33 | **206659.17** | 0.02 | 26 | 206697.72 | **206659.17** | 0.02 | 27 | 206761.96 | **206659.17** | 0.05 | 26 |
| F1 | 60 | **264299.6** | 268346.03 | 267060.43 | 1.53 | 31 | 268430.58 | 267060.43 | 1.56 | 30 | 268306.24 | 267060.43 | 1.52 | 29 |
| F2 | 60 | 265653.47 | 265214.16 | **265214.16** | 0.00 | 29 | 265214.16 | **265214.16** | 0.00 | 29 | 265214.16 | **265214.16** | 0.00 | 28 |
| F3 | 60 | **241120.77** | 241969.77 | 241969.77 | 0.35 | 37 | 241969.77 | 241969.77 | 0.35 | 36 | 241969.77 | 241969.77 | 0.35 | 35 |
| F4 | 60 | **233861.85** | 235175.20 | 235175.20 | 0.56 | 43 | 235528.13 | 235175.20 | 0.71 | 44 | 235449.66 | 235175.20 | 0.68 | 42 |
| G1 | 57 | **306305.4** | 306388.11 | **306305.40** | 0.03 | 23 | 306322.98 | **306305.40** | 0.01 | 23 | 306354.90 | **306305.40** | 0.02 | 22 |
| G2 | 57 | **245440.99** | 245529.35 | **245440.99** | 0.04 | 29 | 245440.99 | **245440.99** | 0.00 | 28 | 245440.99 | **245440.99** | 0.00 | 27 |
| G3 | 57 | **229507.48** | 229507.48 | **229507.48** | 0.00 | 33 | 230737.17 | **229507.48** | 0.54 | 32 | 230583.46 | **229507.48** | 0.47 | 30 |
| G4 | 57 | 235251.47 | 232913.81 | **232521.25** | 0.17 | 32 | 233006.36 | **232521.25** | 0.21 | 32 | 233263.98 | **232521.25** | 0.32 | 31 |
| G5 | 57 | **221730.35** | 221826.32 | **221730.35** | 0.04 | 35 | 222435.96 | **221730.35** | 0.32 | 36 | 222442.67 | **221730.35** | 0.32 | 35 |
| G6 | 57 | **213457.45** | 213541.70 | **213457.45** | 0.04 | 40 | 214090.55 | **213457.45** | 0.30 | 42 | 213457.45 | **213457.45** | 0.00 | 39 |
| H1 | 68 | **268933.06** | 269342.45 | **268933.06** | 0.15 | 41 | 269467.78 | **268933.06** | 0.20 | 42 | 269317.64 | **268933.06** | 0.14 | 39 |
| H2 | 68 | **253365.5** | 253423.34 | **253365.50** | 0.02 | 49 | 253462.09 | **253365.50** | 0.04 | 49 | 254194.18 | **253365.50** | 0.33 | 47 |
| H3 | 68 | **247449.04** | 247532.87 | **247449.04** | 0.03 | 56 | 247508.59 | **247449.04** | 0.02 | 55 | 247449.04 | **247449.04** | 0.00 | 53 |
| H4 | 68 | **250220.77** | 250317.37 | **250220.77** | 0.04 | 52 | 250269.07 | **250220.77** | 0.02 | 53 | 250269.07 | **250220.77** | 0.02 | 52 |
| H5 | 68 | **246121.31** | 246532.25 | **246121.31** | 0.17 | 58 | 246767.73 | **246121.31** | 0.26 | 58 | 246217.90 | **246121.31** | 0.04 | 55 |
| H6 | 68 | **249135.32** | 249294.67 | **249135.32** | 0.06 | 55 | 249231.92 | **249135.32** | 0.04 | 57 | 249206.96 | **249135.32** | 0.03 | 55 |
| I1 | 90 | 351606.91 | 350958.02 | 350258.81 | 0.20 | 55 | 350852.85 | **350245.28** | 0.17 | 54 | 350897.94 | 350247.61 | 0.19 | 52 |
| I2 | 90 | 309955.04 | 312489.95 | **309943.84** | 0.82 | 66 | 311016.93 | **309943.84** | 0.35 | 65 | 310434.77 | **309943.84** | 0.16 | 63 |
| I3 | 90 | **294507.38** | 295236.14 | **294507.38** | 0.25 | 86 | 294858.13 | **294507.38** | 0.12 | 83 | 294821.76 | **294507.38** | 0.11 | 81 |
| I4 | 90 | 295999.65 | 296820.65 | **295988.45** | 0.28 | 79 | 296159.12 | **295988.45** | 0.06 | 77 | 296401.46 | **295988.45** | 0.14 | 76 |
| I5 | 90 | 302524.33 | 302707.04 | **301236.01** | 0.49 | 76 | 301909.59 | **301236.01** | 0.22 | 75 | 301980.98 | **301236.01** | 0.25 | 74 |
| J1 | 95 | 335593.42 | 336680.78 | **335006.68** | 0.50 | 60 | 336522.31 | **335006.68** | 0.45 | 58 | 336789.92 | 335479.75 | 0.53 | 56 |
| J2 | 95 | 310800.53 | 312206.97 | **310417.21** | 0.58 | 71 | 312458.56 | **310417.21** | 0.66 | 67 | 311763.08 | **310417.21** | 0.43 | 65 |
| J3 | 95 | **279219.21** | 281807.92 | **279219.21** | 0.93 | 94 | 279423.74 | **279219.21** | 0.07 | 87 | 279729.03 | **279219.21** | 0.18 | 84 |
| J4 | 95 | 296773.38 | 298412.68 | 297232.88 | 0.63 | 77 | 297781.22 | **296533.16** | 0.42 | 74 | 297344.74 | 297086.58 | 0.27 | 72 |
| K1 | 113 | 395546.4 | 397774.56 | 394846.98 | 0.86 | 86 | 395993.78 | **394375.63** | 0.41 | 83 | 397076.46 | 395006.60 | 0.68 | 81 |
| K2 | 113 | 363214.24 | 365791.18 | 362656.70 | 1.01 | 100 | 362998.61 | **362130.00** | 0.24 | 97 | 363253.47 | **362130.00** | 0.31 | 96 |
| K3 | 113 | 366222.05 | 367806.64 | **365694.08** | 0.58 | 99 | 366218.02 | **365694.08** | 0.14 | 97 | 366388.14 | **365694.08** | 0.19 | 95 |
| K4 | 113 | 349038.84 | 351441.74 | **348949.39** | 0.71 | 113 | 349266.17 | **348949.39** | 0.09 | 111 | 349241.78 | **348949.39** | 0.08 | 108 |
| L1 | 150 | 426017.86 | 428037.41 | **426013.41** | 0.48 | 162 | 427658.80 | **426013.41** | 0.39 | 153 | 427641.03 | 426281.89 | 0.38 | 149 |
| L2 | 150 | 402245.17 | 402073.43 | 401466.27 | 0.21 | 192 | 401587.25 | **401228.80** | 0.09 | 181 | 401492.36 | 401247.70 | 0.07 | 176 |
| L3 | 150 | 403886.22 | 404784.84 | **402677.72** | 0.52 | 187 | 403029.19 | **402677.72** | 0.09 | 176 | 402860.67 | **402677.72** | 0.05 | 174 |
| L4 | 150 | 384844.01 | 387660.68 | **384636.33** | 0.79 | 220 | 385207.32 | **384636.33** | 0.15 | 207 | 385073.14 | **384636.33** | 0.11 | 205 |
| L5 | 150 | 388061.69 | 390091.24 | **387564.55** | 0.65 | 210 | 388677.62 | **387564.55** | 0.29 | 211 | 389778.12 | **387564.55** | 0.57 | 200 |
| M1 | 125 | 400860.79 | 402962.88 | 401006.99 | 1.02 | 108 | 401540.39 | **398913.70** | 0.66 | 104 | 401666.48 | **398913.70** | 0.69 | 102 |
| M2 | 125 | 398908.71 | 400924.09 | 399001.11 | 0.53 | 108 | 401724.68 | 399336.27 | 0.73 | 102 | 401347.29 | **398827.67** | 0.63 | 100 |
| M3 | 125 | 377352.81 | 379362.69 | 377411.62 | 0.85 | 122 | 378502.30 | 377212.23 | 0.62 | 115 | 378031.96 | **376159.13** | 0.50 | 114 |
| M4 | 125 | 348624.42 | 349984.33 | 348624.42 | 0.45 | 147 | 348663.06 | **348417.94** | 0.07 | 140 | 348905.97 | 348532.69 | 0.14 | 137 |
| N1 | 150 | **408926.4** | 414655.53 | 409210.18 | 1.40 | 162 | 414044.03 | 410789.32 | 1.25 | 156 | 414915.65 | 410419.05 | 1.46 | 155 |
| N2 | 150 | **409280.16** | 413434.54 | 410595.02 | 1.02 | 164 | 413124.59 | 409385.19 | 0.94 | 155 | 415985.72 | 411131.25 | 1.64 | 153 |
| N3 | 150 | 396167.85 | 402418.80 | 398841.27 | 2.05 | 181 | 399363.23 | **394337.86** | 1.27 | 177 | 400984.40 | 396827.00 | 1.69 | 170 |
| N4 | 150 | 397753.86 | 401362.13 | 397363.45 | 1.67 | 178 | 402131.56 | 398965.12 | 1.86 | 172 | 400553.31 | **394788.36** | 1.46 | 170 |
| N5 | 150 | 376431.84 | 380168.38 | 375895.96 | 1.79 | 222 | 377447.83 | **373476.30** | 1.06 | 214 | 378201.49 | 375201.45 | 1.27 | 210 |
| N6 | 150 | 377665.19 | 381099.86 | 377368.09 | 1.96 | 216 | 376612.61 | **373758.65** | 0.76 | 211 | 376966.15 | 373789.70 | 0.86 | 209 |
| Tot. | | 18058230 | 18124900 | 18055590 | | 4536 | 18093048 | 18042916 | | 4405 | 18098312 | 18044860 | | 4299 |
| Avg. | | | | | 0.43 | 73 | | | 0.29 | 71 | | | 0.31 | 69 |
| BTPB | | | | 20 | | | | 24 | | | | 23 | | |
| #B | | 36 | | 43 | | | | 53 | | | | 46 | | |

Table 1: *Goetschalckx* problems. The table compares the results obtained by the three configurations of the LNS heuristics with the best results obtained by Osman and Wassan's heuristic [28]. The two first columns show the problem name and the number of customers in the problem. The third column displays the best solution found by Osman and Wassan's heuristic. The rest of the columns are divided into three sections, one for each configuration. These should be interpreted as follows: *avg. sol.* - the average of the solution costs obtained in the 10 experiments, *best sol.* - the cost of the best solution found in the 10 experiments, *avg. gap (%)* - the gap between average and best known solution cost, *avg. time (s)* - the average time needed to perform one experiment (in seconds). The best solution for each problem instance is marked with bold. The row *Tot.* at the bottom of the table gives the sum of the given column and the row *Avg.* gives the average of the column. The row *BTPB* reports the number of problem instances where a particular configuration found solutions that were better than the previous best known solution, the row *#B* contains the number of times the heuristic found the best known solution to a problem.

| | Avg. gap (%) | #B | Avg. time (s) | Opt. | BTPB |
|---|---|---|---|---|---|
| Standard | 0.28 | 35 | 39 | 28 | 8 |
| 6R - no learning | 0.18 | 38 | 40 | 28 | 8 |
| 6R - normal learning | 0.17 | 36 | 40 | 28 | 8 |

Table 2: Summary of testing the 47 first *Goetschalckx* problems using distances rounded to one decimal. Each row in the table corresponds to one of the three LNS configurations. The columns *Avg. gap (%)* and *Avg. Time (s)* should be interpreted like the corresponding entries in the *Avg.* row in Table 1. The rest of the columns are: *#B* - the number of problems where the best known solution was reached, *Opt.* the number of optimal solutions found (out of 34 known optimal solutions), *BTPB* - the number of problems for which the heuristic improved the solutions found by the branch and bound methods. The improved solutions correspond to problems were the branch and bound algorithms did not reach optimality because they were stopped before optimality was proved.

the simpler heuristics. This time the solution costs are decreased by up to 24% and the best known solutions to all problems were improved. As before we note that the heuristics proposed by Nagy and Salhi are faster than the LNS heuristic.

Table 7 compares the three LNS configurations with each other. The most interesting observation is that the multi depot problems seem to be the hardest problems considered so far, as the average solutions are farther from the best known solutions than before, but the results must anyway be considered as very promising.

### 6.6 The Vehicle Routing Problem with Backhauls and Time Windows (VRPBTW)

The VRPBTW is another well-studied backhauling problem. The primary objective considered in the heuristics described in the literature is to minimize the number of vehicles used and the secondary objective is to minimize the traveled distance. These objectives are also used in our experiments. The vehicle minimization is done by solving the problem for a decreasing number of vehicles, as proposed by Ropke [31]. Gelinas et al. [13] proposed a data set containing 15 problems with 100 customers and Thangiah et al. [38] introduced a data set containing 24 large problems.

Our heuristics are tested on both data sets. The results obtained on Gelinas' data set are presented in Table 8. Five papers have reported results on this data set: Duhamel et al. [12], Hasama et al. [20], Reimann et al. [30], Thangiah et al. [38] and Zhong and Cole [48]. It should be noted that apparently there is no standard for how distances should be represented internally in the heuristic, which makes comparisons a bit problematic. We have chosen to represent the distances using doubles like Reimann et al. [30], as is standard in the literature about VRPTW heuristics. The tables reveal that we are able to improve 10 out of the 15 solutions and reduce the number of vehicles needed for 5 of the problems. Again the configurations using all removal heuristics turns out to be the best.

The only heuristic that has been applied to the large VRPBTW problems is the heuristic by Thangiah et al. [38]. Table 9 compares the results obtained by this algorithm to the results obtained by the LNS heuristic. We see that the LNS heuristic is able to decrease the necessary number of vehicles by a large amount and at the same time also decrease the traveled distance. The best known solutions to all 24 problems were improved by the LNS heuristic. Table 10 gives further information about the performance of the LNS heuristic, including the running time. The time increases with the problem size, but its growth is not alarming. Once again the configurations using 6 removal heuristics found the best solutions.

| | Best known | | | | Standard | | | | 6R - no learning | | | | 6R - normal learning | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | n | cost | opt | reference | avg. sol. | best sol. | avg. gap (%) | avg. time (s) | avg. sol. | best sol. | avg. gap (%) | avg. time (s) | avg. sol. | best sol. | avg. gap (%) | avg. time (s) |
| EIL22.50A | 21 | **371** | X | TV + EHP | 371 | **371** | 0.00 | 8 | 371 | **371** | 0.00 | 8 | 371 | **371** | 0.00 | 8 |
| EIL22.66A | 21 | **366** | X | TV + EHP | 366 | **366** | 0.00 | 7 | 366 | **366** | 0.00 | 8 | 366 | **366** | 0.00 | 7 |
| EIL22.80A | 21 | **375** | X | TV + EHP | 375 | **375** | 0.00 | 7 | 375 | **375** | 0.00 | 8 | 375 | **375** | 0.00 | 8 |
| EIL23.50A | 22 | **682** | X | TV + EHP | 709 | **682** | 3.94 | 13 | 682 | **682** | 0.00 | 12 | 682 | **682** | 0.00 | 12 |
| EIL23.66A | 22 | **649** | X | TV + EHP | 654 | **649** | 0.77 | 12 | 649 | **649** | 0.00 | 13 | 649 | **649** | 0.00 | 13 |
| EIL23.80A | 22 | **623** | X | TV + EHP | 625 | **623** | 0.26 | 11 | 623 | **623** | 0.00 | 12 | 623 | **623** | 0.00 | 12 |
| EIL30.50A | 29 | **501** | X | TV + EHP | 501 | **501** | 0.00 | 17 | 501 | **501** | 0.00 | 19 | 501 | **501** | 0.00 | 18 |
| EIL30.66A | 29 | **537** | X | TV + EHP | 537 | **537** | 0.00 | 13 | 537 | **537** | 0.00 | 14 | 537 | **537** | 0.00 | 14 |
| EIL30.80A | 29 | **514** | X | TV + EHP | 514 | **514** | 0.00 | 13 | 514 | **514** | 0.00 | 14 | 514 | **514** | 0.00 | 14 |
| EIL33.50A | 32 | **738** | X | TV + EHP | 738 | **738** | 0.00 | 17 | 738 | **738** | 0.00 | 20 | 738 | **738** | 0.00 | 20 |
| EIL33.66A | 32 | **750** | X | TV + EHP | 750 | **750** | 0.00 | 15 | 750 | **750** | 0.00 | 17 | 750 | **750** | 0.00 | 16 |
| EIL33.80A | 32 | **736** | X | TV + EHP | 737 | **736** | 0.18 | 15 | 736 | **736** | 0.05 | 15 | 736 | **736** | 0.05 | 15 |
| EIL51.50A | 50 | **559** | X | TV + EHP | 561 | **559** | 0.41 | 35 | 559 | **559** | 0.00 | 39 | 559 | **559** | 0.00 | 36 |
| EIL51.66A | 50 | **548** | X | TV + EHP | 553 | **548** | 0.91 | 30 | 550 | **548** | 0.35 | 31 | 549 | **548** | 0.11 | 30 |
| EIL51.80A | 50 | **565** | X | TV + EHP | 569 | **565** | 0.65 | 28 | 571 | **565** | 1.12 | 29 | 570 | **565** | 0.80 | 28 |
| EILA76.50A | 75 | **739** | X | TV + EHP | 740 | **739** | 0.16 | 49 | 739 | **739** | 0.00 | 50 | 739 | **739** | 0.00 | 48 |
| EILA76.66A | 75 | **768** | X | TV + EHP | 774 | **768** | 0.77 | 44 | 774 | 769 | 0.73 | 44 | 772 | **768** | 0.51 | 42 |
| EILA76.80A | 75 | **781** | | TV + EHP | 794 | 783 | 1.63 | 41 | 794 | 783 | 1.72 | 40 | 791 | 783 | 1.22 | 39 |
| EILB76.50A | 75 | **801** | X | TV + EHP | 804 | **801** | 0.31 | 42 | 802 | **801** | 0.12 | 42 | 803 | **801** | 0.25 | 40 |
| EILB76.66A | 75 | **873** | X | TV + EHP | 876 | **873** | 0.38 | 38 | 875 | **873** | 0.22 | 38 | 873 | **873** | 0.01 | 37 |
| EILB76.80A | 75 | **919** | X | TV + EHP | 927 | **919** | 0.90 | 36 | 924 | **919** | 0.58 | 38 | 922 | **919** | 0.37 | 37 |
| EILC76.50A | 75 | **713** | X | TV + EHP | 715 | **713** | 0.21 | 60 | 713 | **713** | 0.04 | 61 | 713 | **713** | 0.00 | 59 |
| EILC76.66A | 75 | **734** | X | EHP | 740 | 735 | 0.75 | 51 | 739 | **734** | 0.69 | 51 | 736 | **734** | 0.23 | 50 |
| EILC76.80A | 75 | **733** | | TV + EHP | 738 | 734 | 0.71 | 48 | 741 | 736 | 1.09 | 48 | 738 | 737 | 0.70 | 47 |
| EILD76.50A | 75 | **690** | X | TV + EHP | 702 | **690** | 1.77 | 71 | 696 | **690** | 0.81 | 75 | 691 | **690** | 0.20 | 71 |
| EILD76.66A | 75 | **715** | | TV + EHP | 717 | **715** | 0.22 | 59 | 716 | **715** | 0.20 | 60 | 715 | **715** | 0.00 | 57 |
| EILD76.80A | 75 | **694** | | EHP | 699 | **694** | 0.72 | 53 | 699 | 695 | 0.76 | 55 | 696 | **694** | 0.26 | 53 |
| EILA101.50A | 100 | 842 | | OSMAN | 845 | 837 | 1.72 | 138 | 840 | **831** | 1.05 | 137 | 836 | **831** | 0.55 | 129 |
| EILA101.66A | 100 | **846** | X | TV + EHP | 852 | **846** | 0.67 | 99 | 848 | **846** | 0.21 | 100 | 846 | **846** | 0.05 | 99 |
| EILA101.80A | 100 | 875 | | OSMAN | 872 | 862 | 1.77 | 91 | 869 | **857** | 1.41 | 87 | 866 | 861 | 1.03 | 86 |
| EILB101.50A | 100 | 933 | | EHP | 930 | **925** | 0.54 | 82 | 928 | **925** | 0.31 | 79 | 929 | **925** | 0.38 | 77 |
| EILB101.66A | 100 | 998 | | OSMAN | 1007 | 994 | 1.79 | 69 | 1010 | **989** | 2.13 | 66 | 1001 | 991 | 1.24 | 66 |
| EILB101.80A | 100 | 1021 | | OSMAN | 1022 | 1018 | 1.43 | 63 | 1021 | 1010 | 1.26 | 61 | 1015 | **1008** | 0.65 | 61 |
| Tot. | | 23189 | | | 23314 | 23160 | | 1373 | 23251 | 23140 | | 1394 | 23201 | 23142 | | 1349 |
| Avg. | | | | | | | 0.71 | 42 | | | 0.45 | 42 | | | 0.26 | 41 |
| BTPB | | | | | 5 | | | | 5 | | | | 5 | | | |
| #B | 28 | | | | 26 | | | | 28 | | | | 29 | | | |

Table 3: *Toth-Vigo* data set. The column *opt* indicates if optimality is proven for the particular instance and the column *reference* points to the algorithm that found the solution in the *best known* column. *TV* refers to the exact method by Toth and Vigo [41], *EHP* refers to the exact algorithm by Mingozzi et al. [26] and *OSMAN* refers to the heuristic by Osman and Wassan [28].

| | NS1 | NS2 | Standard | 6R - no learning | 6R - normal learning |
|---|---|---|---|---|---|
| 10% | 1011 | 995 | 956 (3.9%) | **955 (4.0%)** | 956 (3.9%) |
| 25% | 1034 | 998 | 923 (7.5%) | 923 (7.5%) | **922 (7.6%)** |
| 50% | 1045 | 991 | **881 (11.1%)** | **881 (11.1%)** | **881 (11.1%)** |

Table 4: Summary of the 42 Nagy-Salhi MVRPB problem instances. This table compares the solutions obtained by the LNS heuristic to those obtained by Nagy and Salhi [27], [32]. Each row reports the average solution over 14 MVRPB instances with a particular percentage of backhaul customers (10%, 25% or 50%). The columns NS1 and NS2 contain the best results reported by Nagy and Salhi in [32] and [27] respectively. The last three columns show the results obtained by the LNS heuristic. The numbers in parenthesis show how much better the LNS solutions are compared to the solutions reported by Nagy and Salhi.

| | Standard | | | | 6R - no learning | | | | 6R - normal learning | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Avg. gap (%) | #B | BTPB | Avg. time (s) | Avg. gap (%) | #B | BTPB | Avg. time (s) | Avg. gap (%) | #B | BTPB | Avg. time (s) |
| 10% | 0.51 | 10 | 13 | 129 | 0.43 | 11 | 13 | 133 | 0.37 | 11 | 13 | 133 |
| 25% | 0.49 | 11 | 14 | 135 | 0.38 | 9 | 14 | 142 | 0.30 | 11 | 14 | 143 |
| 50% | 0.71 | 7 | 13 | 164 | 0.45 | 10 | 14 | 178 | 0.41 | 12 | 13 | 178 |

Table 5: This table provides a comparison of the 3 LNS configurations when applied to the 42 Nagy-Salhi MVRPB instances. Each row summarizes 14 instances with the same percentage of backhaul customers. The meaning of the headings is as in Table 2.

| | NS1 | NS2 | Standard | 6R - no learning | 6R - normal learning |
|---|---|---|---|---|---|
| 10% | 2008 | 1996 | 1798 (9.9%) | **1795 (10.1%)** | 1799 (9.9%) |
| 25% | 2050 | 2007 | 1671 (16.7%) | 1663 (17.1%) | **1662 (17.2%)** |
| 50% | 2088 | 1993 | 1512 (24.1%) | 1510 (24.2%) | **1509 (24.3%)** |

Table 6: Summary of results obtained on the 33 Nagy-Salhi MDMVRPB instances. The columns NS1 and NS2 contain the best results reported by Nagy and Salhi in [32] and [27] respectively.

| | Standard | | | | 6R - no learning | | | | 6R - normal learning | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Avg. gap (%) | #B | BTPB | Avg. time (s) | Avg. gap (%) | #B | BTPB | Avg. time (s) | Avg. gap (%) | #B | BTPB | Avg. time (s) |
| 10% | 0.93 | 7 | 11 | 204 | 0.63 | 10 | 11 | 217 | 0.61 | 6 | 11 | 216 |
| 25% | 0.97 | 5 | 11 | 219 | 0.65 | 6 | 11 | 237 | 0.66 | 8 | 11 | 237 |
| 50% | 0.88 | 8 | 11 | 258 | 0.71 | 6 | 11 | 288 | 0.66 | 7 | 11 | 288 |

Table 7: Nagy-Salhi MDMVRPB instances. Comparison of the performance of the three LNS configurations.

| | Best known | | | | Standard | | | | 6R - no learning | | | | 6R - normal learning | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | % BH | m | cost | ref | avg. #veh. | best sol. | best #veh. | avg. time (s) | avg. #veh. | best sol. | best #veh. | avg. time (s) | avg. #veh. | best sol. | best #veh. | avg. time (s) |
| BHR101A | 10% | 22 | 1831.68 | RDH | 22.0 | **1818.86** | **22** | 98 | 22.0 | **1818.86** | **22** | 107 | 22.0 | **1818.86** | **22** | 109 |
| BHR101B | 30% | 23 | 1999.16 | RDH | 23.0 | 1959.86 | 23 | 94 | 23.0 | **1959.56** | **23** | 101 | 23.0 | **1959.56** | **23** | 103 |
| BHR101C | 50% | **24** | **1909.84** | HKK | 24.0 | 1939.10 | 24 | 93 | 24.0 | 1939.10 | 24 | 100 | 24.0 | 1939.10 | 24 | 101 |
| BHR102A | 10% | 19 | 1677.62 | RDH | 19.0 | **1653.19** | **19** | 110 | 19.0 | **1653.19** | **19** | 118 | 19.0 | **1653.19** | **19** | 121 |
| BHR102B | 30% | **21** | **1764.3** | TPS | 22.0 | 1750.70 | 22 | 103 | 22.0 | 1750.70 | 22 | 111 | 22.0 | 1750.70 | 22 | 114 |
| BHR102C | 50% | **21** | **1745.7** | TPS | 22.0 | 1775.76 | 22 | 103 | 22.0 | 1775.76 | 22 | 111 | 22.0 | 1775.76 | 22 | 113 |
| BHR103A | 10% | **15** | **1371.6** | TPS | 15.0 | 1387.57 | 15 | 117 | 15.0 | 1387.57 | 15 | 123 | 15.0 | 1387.57 | 15 | 128 |
| BHR103B | 30% | 16 | 1395.88 | RDH | 15.0 | **1390.33** | **15** | 108 | 15.0 | **1390.33** | **15** | 112 | 15.0 | **1390.33** | **15** | 115 |
| BHR103C | 50% | **16** | **1486.56** | ZC | 17.0 | 1457.31 | 17 | 106 | 17.0 | 1456.48 | 17 | 113 | 17.0 | 1456.48 | 17 | 115 |
| BHR104A | 10% | 11 | 1205.78 | RDH | 11.0 | 1084.22 | 11 | 127 | 11.0 | **1084.17** | **11** | 130 | 11.0 | **1084.17** | **11** | 132 |
| BHR104B | 30% | 12 | 1128.3 | RDH | 11.0 | 1163.24 | 11 | 119 | 11.0 | **1154.84** | **11** | 121 | 11.0 | **1154.84** | **11** | 122 |
| BHR104C | 50% | 12 | 1208.46 | RDH | 11.0 | 1191.41 | 11 | 117 | 11.0 | **1191.38** | **11** | 119 | 11.0 | **1191.38** | **11** | 119 |
| BHR105A | 10% | 16 | 1544.81 | RDH | 15.5 | 1564.88 | 15 | 104 | 15.3 | **1561.28** | **15** | 110 | 15.4 | **1561.28** | **15** | 109 |
| BHR105B | 30% | 16 | 1592.23 | RDH | 16.0 | **1583.30** | **16** | 97 | 16.0 | **1583.30** | **16** | 102 | 16.0 | **1583.30** | **16** | 102 |
| BHR105C | 50% | 17 | 1633.01 | RDH | 16.6 | 1711.36 | 16 | 96 | 16.6 | 1710.75 | 16 | 100 | 16.5 | **1710.19** | **16** | 100 |
| Tot. | | 261 | 23495 | | 260.2 | 23432 | 259 | 1593 | 260.0 | 23418 | 259 | 1679 | 259.9 | 23417 | 259 | 1703 |
| Avg. | | | | | | | | 106 | | | | 112 | | | | 114 |
| BTPB | | | | | 10 | | | | 10 | | | | 10 | | | |
| #B | 5 | | | | 4 | | | | 9 | | | | 10 | | | |

Table 8: The table shows the results obtained on the VRPBTW instances proposed by Gelinas et al. [13]. The first column shows the name of the problem, the next columns are: *%BH* - ratio of backhaul customers, *m* - number of vehicles in best known solution, *cost* - distance traveled in best known solution, *ref* - HKK = Hasama et al. [20], RDH = Reimann et al. [30], TPS = Thangiah et al. [38] and ZC = Zhong and Cole [48], the result found by Zhong and Cole was listed in their technical report [47]. The rest of the columns report the solutions found by the LNS heuristics: *avg. #veh.* - average number of vehicles *best #veh.* - lowest number of vehicles found. The other columns should be interpreted as in Table 1. The original data files do not specify the latest return time to the depot and the maximum capacity of the vehicle. In our experiments these parameters have been set to the values they have in the original Solomon problems from which the Gelinas problems were created.

| | TPS | | Standard | | 6R - no learning | | 6R - normal learning | |
|---|---|---|---|---|---|---|---|---|
| | #veh. | cost | #veh. | cost | #veh. | cost | #veh. | cost |
| 250 | 517 | 57509 | 449 | 54256 | **444** | **54711** | 445 | 54499 |
| 500 | 799 | 94144 | 677 | 83498 | 676 | 82946 | **675** | **82796** |

Table 9: Large VRPBTW instances. This table compares the 3 LNS configurations to the heuristics by Thangiah et al. (TPS). The data set contains 12 problems containing 250 customers and 12 containing 500 customers. The best solutions found by the heuristics have been accumulated and the table shows the total number of vehicles needed and the total traveled distance for all instances of a particular size. The vehicle capacity was set to 200 for all problems and no latest arrival time was specified for the depot.

| | Standard | | | | 6R - no learning | | | | 6R - normal learning | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Customers | Avg. #veh. | #B | BTPB | Avg. time (s) | Avg. #veh. | #B | BTPB | Avg. time (s) | Avg. #veh. | #B | BTPB | Avg. time (s) |
| 250 | 37.5 | 1 | 12 | 489 | 37.3 | 6 | 12 | 492 | 37.4 | 5 | 12 | 504 |
| 500 | 57.1 | 0 | 12 | 1562 | 56.8 | 4 | 12 | 1651 | 56.7 | 8 | 12 | 1570 |

Table 10: Comparison of the three LNS configurations when faced with the large VRPBTW instances proposed by Thangiah. The *Avg. #veh* column displays the average of the average number of vehicles needed to serve all customers.

| | LB | KB | | ZC | | Standard | | 6R - no learning | | 6R - normal learning | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | #veh. | #veh. | cost | #veh. | cost | #veh. | cost | #veh. | cost | #veh. | cost |
| MR2 | 4 | 4 | 1168.53 | 4 | 1016.66 | 4 | 904.55 | **4** | **902.73** | 4 | 903.00 |
| MC2 | 4 | 4 | 1094.94 | 4.625 | 903.56 | **4** | **731.38** | 4 | 732.38 | 4 | 732.13 |
| MRC2 | 4 | 4.5 | 1496.91 | 4.125 | 1330.31 | **4.125** | **1125.00** | 4.125 | 1129.25 | 4.125 | 1127.63 |

Table 11: Kontoravdis MVRPBTW problems. The table compares the results reported by Kontoravdis and Bard [23] (KB) and Zhong and Cole [48] (ZC) with the results obtained using the LNS heuristics. The primary objective in these problems is to minimize the number of vehicles needed to serve the customers. The data set is divided into three classes according to the geographical distribution of the customers in the problems: randomly distributed customers (MR2), clustered customers (MC2), and a mix between the two first categories (MRC2). The MRC2 and MC2 classes both contain 8 problems while the MR2 class contains 11 problems. Each row in the table summarizes the performance on each class. The column *LB #veh.* shows the lower bound on the number of vehicles as given by Kontoravdis and Bard.

| | Standard | | | | 6R - no learning | | | | 6R - normal learning | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Avg. gap (%) | #B | BTPB | Avg. time (s) | Avg. gap (%) | #B | BTPB | Avg. time (s) | Avg. gap (%) | #B | BTPB | Avg. time (s) |
| MR2 | 1.34 | 4 | 11 | 362 | 0.63 | 8 | 11 | 375 | 0.63 | 8 | 11 | 368 |
| MC2 | 0.62 | 6 | 8 | 162 | 0.60 | 5 | 8 | 165 | 0.65 | 5 | 8 | 163 |
| MRC2 | 2.83 | 5 | 8 | 183 | 1.99 | 1 | 8 | 183 | 1.76 | 4 | 8 | 180 |

Table 12: The table compares the three LNS configurations when applied to Kontoravdis' MVRPBTW problems. In all test runs the heuristics reached the same number of vehicles when applied to the same problem. This allows us to report the *avg. gap*, which doesn't make sense if the heuristics use a different number of vehicles to solve the same problem.

### 6.7 The Mixed Vehicle Routing Problem with Backhauls and Time Windows (MVRPBTW)

Two datasets have been proposed for the MVRPBTW. Hasama et al. [20] use Gelinas' data set by relaxing the linehaul-before-backhaul constraint while Kontoravdis and Bard [23] construct 27 new problems from Solomon's VRPTW problems. We test our heuristics using Kontoravdis and Bard's data set which also has been attempted by Zhong and Cole [48]. The LNS heuristic is compared to the previous heuristics in Table 11. Again the LNS heuristic is able to find solutions of better quality compared to the older heuristics. It is interesting to note that the LNS heuristic reaches the lower bound on the number of vehicles needed to solve the problems on all but one instance. The LNS heuristics improved all the previously best known solutions to the problem instances.

Table 12 provides the usual comparison of the three LNS configurations. It should be observed that the MRC2 problems turn out to be hard to solve, as indicated by the rather large gaps. This is not surprising as the MRC2 problems were constructed from Solomon's RC2 VRPTW problems, which are known to be hard to solve. One cannot expect that adding the extra complexity of backhaul customers should make the problems easier to solve.

### 6.8 The Vehicle Routing Problem with Simultaneous Deliveries and Pickups (VRPSDP)

Allthough the VRPSDP is not the problem in the backhauling family that has received the most attention, there exist nevertheless quite a few data sets for the problem. The first data set was proposed by Min [25] and contained only one problem, which originated from a real life application. Halse [19] proposed a set containing 16 problems constructed from CVRP problems and Dethloff [10] proposed 40 new problems containing 50 customers each. Nagy and Salhi [32] constructed two classes of VRPSDP problems and two classes of multi depot VRPSDP

| | Dethloff | NS1 | NS2 | Standard | 6R - no learning | 6R - normal learning |
|---|---|---|---|---|---|---|
| Dethloff | 824 | - | - | 747 ( 9.3%) | 746 (9.5%) | **745 (9.6%)** |
| NS-X | 1006 | 1096 | 991 | 927 (6.5%) | 925 (6.7%) | **919 (7.3%)** |

Table 13: Summary of the results obtained on the VRPSDP instances. The table should be interpreted like Table 4. The row denoted *Dethloff* summarizes the results obtained on Dethloff's 40 instances [10] and the single instance provided by Min [25]. Each of Dethloff's instances contains 50 customers. The row marked *NS-X* summarizes Nagy and Salhi's 14 VRPSDP instances of class *X* [32]. These problems contain between 50 and 200 customers. Results for these problems are reported by Dethloff [10] and Nagy and Salhi [32], [27]. The columns *Dethloff*, *NS1* and *NS2* summarize the best results reported in [10], [32] and [27] respectively.

| | Standard | | | | 6R - no learning | | | | 6R - normal learning | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Avg. gap (%) | #B | BTPB | Avg. time (s) | Avg. gap (%) | #B | BTPB | Avg. time (s) | Avg. gap (%) | #B | BTPB | Avg. time (s) |
| Dethloff | 1.07 | 24 | 40 | 128 | 0.96 | 23 | 40 | 129 | 0.58 | 36 | 40 | 155 |
| NS-X | 2.81 | 6 | 11 | 685 | 2.73 | 7 | 13 | 686 | 2.00 | 7 | 12 | 772 |

Table 14: The table compares the 3 LNS configurations when applied to VRPSDP instances.

problems. Finally Angelelli and Mansini [2] presented a class of VRPSDP problems with time windows.

As mentioned earlier we are not going to test our heuristic on the multi depot and time window variants of the VRPSDP. The problems we choose for our tests are Min's problem, Dethloff's problems and the first class of Nagy and Salhi's VRPSDP problems (the one denoted with an *X* in [32]). The results are summarized in Tables 13 and 14. Again it must be stressed that the heuristics by Dethloff and Nagy and Salhi are simple construction heuristics that are substantially faster than the LNS heuristics.

The LNS heuristics find the optimal solution to Min's problem (the optimal solution was found by Halse [19]) and are able to improve all of the best known solutions to Dethloff's problems which were found using Dethloff's construction heuristic. The *6R - normal learning* configuration is able to improve the best known solutions by more than 9%. Having said that, it should be noticed that the LNS heuristics are fairly slow when faced with this type of problems, because each order is represented by 2 requests and introduces significant overhead in the algorithm. This also suggests that this problem type would benefit greatly from a specialized version of the LNS heuristic where the overhead can be avoided. The LNS heuristic also experiences difficulties when faced with the larger problems from Nagy and Salhi's data set. Here the avg. gap increases to 2% for the best configuration, but the heuristic nevertheless improves 13 of the 14 best known solutions. The configuration with learning enabled and using all 6 removal heuristics clearly is the most robust configuration when faced with these hard problems.

### 6.9 Computational experiments conclusion

In Section 6.2 we raised a number of questions that the computational experiments should clarify. The first question was whether it is possible to design a unified heuristic for a large class of vehicle routing problems with backhauls that is able to provide solutions comparable to those obtained by specialized heuristics. We believe that the experiments conducted in this paper show that this indeed is possible. This is an interesting achievement, as it to a large extent allows practitioners to focus on a single heuristic and apply this to the problems they are faced with instead of "reinventing the wheel" each time a new problem type needs to be solved.

The second question asked to give an evaluation of the effect of the three new removal heuristics and the

| | #prob | Standard | | 6R - no learning | | 6R - normal learning | |
|---|---|---|---|---|---|---|---|
| | | Avg. gap (%) | #B | Avg. gap (%) | #B | Avg. gap (%) | #B |
| Goetschalckx 1 | 62 | 0.43 | 43 | 0.29 | 53 | 0.31 | 46 |
| Goetschalckx 2 | 47 | 0.28 | 35 | 0.17 | 38 | 0.17 | 36 |
| Toth-Vigo | 33 | 0.71 | 26 | 0.45 | 28 | 0.26 | 29 |
| MVRPB 50% | 14 | 0.71 | 7 | 0.45 | 10 | 0.41 | 12 |
| MVRPB 25% | 14 | 0.49 | 11 | 0.38 | 9 | 0.3 | 11 |
| MVRPB 10% | 14 | 0.51 | 10 | 0.43 | 11 | 0.37 | 11 |
| MDMVRPB 50% | 11 | 0.88 | 8 | 0.71 | 6 | 0.66 | 7 |
| MDMVRPB 25% | 11 | 0.97 | 5 | 0.65 | 6 | 0.66 | 8 |
| MDMVRPB 10% | 11 | 0.93 | 7 | 0.63 | 10 | 0.61 | 6 |
| VRPSDP 1 | 41 | 1.07 | 24 | 0.96 | 23 | 0.58 | 36 |
| VRPSDP 2 | 14 | 2.81 | 5 | 2.73 | 7 | 2.00 | 7 |
| MVRPBTW C | 8 | 0.63 | 6 | 0.6 | 5 | 0.65 | 5 |
| MVRPBTW R | 11 | 1.34 | 4 | 0.63 | 8 | 0.63 | 8 |
| MVRPBTW RC | 8 | 2.83 | 5 | 1.99 | 1 | 1.76 | 3 |
| VRPBTW 1 | 15 | | 4 | | 9 | | 10 |
| VRPBTW 2 | 24 | | 1 | | 10 | | 13 |
| Avg. | | 0.81 | | 0.62 | | 0.50 | |
| Sum | 338 | | 201 | | 234 | | 248 |

Table 15: Summary of experiments. This table shows a summary of the tests performed in this paper. Each row in the table corresponds to a problem class. Most of the titles in the first row should be fairly self explanatory: *Goetschalckx 1* - Goetschalckx VRPB without rounding distances, *Goetschalckx 2* - Goetschalckx VRPB where distances have been rounded to one decimal. *VRPSDP 1* - Dethloff VRPSDP , *VRPSDP 2* - Nagy-Salhi VRPSDP, *VRPBTW 1* - Gelinas VRPBTW *VRPBTW 2* - Thangiah VRPBTW. The column *#prob* displays the number of problems in each class. The *Avg.* row shows the averages of the *Avg. gap(%)* column. The numbers in the avg. row were calculated by summing the products of the numbers in the *#prob* column with the numbers in the *gap* column and dividing the sum by the total number of problems. This was done to take into account that some data sets contains more problems than others. The missing entries in the VRPBTW rows have been left out because the primary objective of these problems is to minimize the number of vehicles and not all test runs resulted in the same number of vehicles. Reporting the gap for these runs could make the heuristic that could not reach the minimum number of vehicles look too good.

consequence of disabling the learning layer. Table 15 provides an overview of the experiments performed. The *Avg.* row displays the overall gaps between average solutions and best known solutions. This gap is an indication of the robustness of the heuristic. The *Sum* row contains the number of problems for which the particular LNS configuration found the best known solution. The table clearly shows the impact of adding the three new removal heuristics, as we see a great improvement in the quality of the heuristic from configuration 1 to configuration 3. The table also shows that disabling the learning layer decreases the overall quality of the results as expected. Although comparable results can be obtained without the learning layer for specific problem types, the learning layer apparently helps the algorithm to adapt to all the various problem types.

# 7 Conclusion

This paper is the first to present a unified heuristic for a large class of vehicle routing problems with backhauls. For this purpose we have introduced a Rich VRPTW model which extends the ordinary VRP model with time windows, pickup and delivery pairs, as well as precedence constraints. The model is very expressive, and it allows us to model all of the most common VRPB models within the framework, as well as other routing problems from the literature. The unified model has the additional benefit that it allows us to combine pickup and delivery request with a more clean VRPB or VRPSPD, as well as scheduling mixed transportation problems for a general fleet of vehicles.

For several of the VRPB problem types presented in this paper, we report the first applications of a metaheuristic to the problem. The results are very promising as we found a new best solution to 67% of the problems tested. Even faster and better performing heuristics could be constructed by specializing the proposed heuristic to just one of the problem types. We have chosen not to do this to maintain the generality of the solution approach.

The present experiments indicate that the combination of several neighborhoods makes it easier for the local search heuristic to explore the solution space, and hence to find solutions of high quality. This conforms to similar observations for simpler neighborhoods.

The monitoring and learning layer to control the choice of neighborhoods can be seen as a layer which maintains a proper balance between intensification and diversification. Several other approaches have been working with this balance, see e.g. Reactive Tabu Search [4]. In the proposed framework we do not explicitly care about which heuristics intensify or diversify the search. The layer steadily maintains a proper balance of the heuristics so that new, improved solutions are found. The computational results show that the learning layer overall is able to increase the robustness of the heuristic but also indicate that further refinements may be possible as the configuration without the learning layer occasionally outperformed the configuration that included the learning layer.

An interesting topic for further research would be to apply the framework proposed in this paper to combinatorial optimization problems outside the vehicle routing domain.

## 8 Acknowledgements

The authors wish to thank Jan Dethloff, George Kontoravdis, Marc Reimann, Sam R. Thangiah and Daniele Vigo for kindly providing us with the data sets used in this paper and for answering questions regarding the data sets. Furthermore we wish to thank Jakob Birkedal Nielsen for proposing the Cluster Removal heuristic and Gabor Nagy for sending us his working paper.

## 9 Appendix

This section contains additional information about the experiments performed in section 6. Tables 16 to 31 list the individual solutions found to the many problem instances considered in this paper.

An important comment should be made about Table 17. The results in this table were obtained by rounding distances to the nearest integer when doing distance calculations. This gives results that look like the results reported in Table III in Osman and Wassan [28] and Table 1 in Toth and Vigo [40] but both author pairs state that results in these tables were found using a different rounding procedure. We have not been able to reproduce the results in the two mentioned tables from Toth and Vigo and Osman and Wassan papers using the rounding procedures described in the papers. Consequently, the objective values listed in the column *Best known* in table 17 should only be seen as a rough guideline of the obtainable solution quality, and the table should not be used to make a direct comparison between the LNS heuristic and the heuristics by Toth and Vigo and Osman and Wassan.

| | Best known | | | | Std. Removals | | | | 6R - no learning | | | | 6R - normal learning | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | n | cost | opt. | reference | avg. sol. | best sol. | avg. gap (%) | avg. time (s) | avg. sol. | best sol. | avg. gap (%) | avg. time (s) | avg. sol. | best sol. | avg. gap (%) | avg. time (s) |
| A1 | 25 | **229886** | X | TV + EHP | 229886 | **229886** | 0.00 | 7 | 229886 | **229886** | 0.00 | 7 | 229886 | **229886** | 0.00 | 7 |
| A2 | 25 | **180119** | X | TV + EHP | 180119 | **180119** | 0.00 | 7 | 180119 | **180119** | 0.00 | 8 | 180119 | **180119** | 0.00 | 8 |
| A3 | 25 | **163405** | X | TV + EHP | 163405 | **163405** | 0.00 | 9 | 163405 | **163405** | 0.00 | 10 | 163405 | **163405** | 0.00 | 9 |
| A4 | 25 | **155796** | X | TV + EHP | 155796 | **155796** | 0.00 | 10 | 155796 | **155796** | 0.00 | 11 | 155796 | **155796** | 0.00 | 10 |
| B1 | 30 | **239080** | X | TV + EHP | 239080 | **239080** | 0.00 | 8 | 239080 | **239080** | 0.00 | 9 | 239080 | **239080** | 0.00 | 9 |
| B2 | 30 | **198048** | X | TV + EHP | 198048 | **198048** | 0.00 | 10 | 198048 | **198048** | 0.00 | 10 | 198048 | **198048** | 0.00 | 10 |
| B3 | 30 | **169372** | X | TV + EHP | 169372 | **169372** | 0.00 | 12 | 169372 | **169372** | 0.00 | 14 | 169372 | **169372** | 0.00 | 14 |
| C1 | 40 | **249449** | X | TV + EHP | 250899 | 250557 | 0.58 | 13 | 251037 | 250557 | 0.64 | 14 | 250557 | 250557 | 0.44 | 13 |
| C2 | 40 | **215020** | X | TV + EHP | 215020 | **215020** | 0.00 | 15 | 215020 | **215020** | 0.00 | 16 | 215020 | **215020** | 0.00 | 16 |
| C3 | 40 | **199346** | X | TV + EHP | 199346 | **199346** | 0.00 | 17 | 199346 | **199346** | 0.00 | 18 | 199346 | **199346** | 0.00 | 18 |
| C4 | 40 | **195366** | X | TV + EHP | 195366 | **195366** | 0.00 | 18 | 195366 | **195366** | 0.00 | 19 | 195366 | **195366** | 0.00 | 19 |
| D1 | 38 | **322530** | X | TV + EHP | 322530 | **322530** | 0.00 | 11 | 322530 | **322530** | 0.00 | 12 | 322530 | **322530** | 0.00 | 12 |
| D2 | 38 | **316709** | X | TV + EHP | 316709 | **316709** | 0.00 | 11 | 316709 | **316709** | 0.00 | 13 | 316709 | **316709** | 0.00 | 12 |
| D3 | 38 | **239479** | X | EHP | 239479 | **239479** | 0.00 | 12 | 239479 | **239479** | 0.00 | 13 | 239479 | **239479** | 0.00 | 12 |
| D4 | 38 | **205832** | X | EHP | 205832 | **205832** | 0.00 | 14 | 205832 | **205832** | 0.00 | 15 | 205832 | **205832** | 0.00 | 15 |
| E1 | 45 | **238880** | X | TV + EHP | 238880 | **238880** | 0.00 | 16 | 238880 | **238880** | 0.00 | 18 | 238880 | **238880** | 0.00 | 18 |
| E2 | 45 | **212263** | X | TV + EHP | 212547 | **212263** | 0.13 | 21 | 212263 | **212263** | 0.00 | 23 | 212505 | **212263** | 0.11 | 24 |
| E3 | 45 | **206659** | X | TV + EHP | 206698 | **206659** | 0.02 | 24 | 206698 | **206659** | 0.02 | 27 | 206711 | **206659** | 0.03 | 26 |
| F1 | 60 | **263173** | X | TV + EHP | 268334 | 267060 | 1.96 | 28 | 268463 | 267060 | 2.01 | 29 | 268321 | 267060 | 1.96 | 29 |
| F2 | 60 | **265213** | X | TV + EHP | 265213 | **265213** | 0.00 | 27 | 265213 | **265213** | 0.00 | 28 | 265213 | **265213** | 0.00 | 28 |
| F3 | 60 | **241120** | X | TV + EHP | 241969 | 241969 | 0.35 | 33 | 241969 | 241969 | 0.35 | 35 | 241969 | 241969 | 0.35 | 35 |
| F4 | 60 | **233861** | X | TV + EHP | 236547 | 235175 | 1.15 | 40 | 235258 | 235175 | 0.60 | 42 | 235449 | 235175 | 0.68 | 42 |
| G1 | 57 | **306305** | X | EHP | 306450 | 306306 | 0.05 | 21 | 306306 | 306306 | 0.00 | 22 | 306306 | 306306 | 0.00 | 22 |
| G2 | 57 | **245441** | X | EHP | 245441 | **245441** | 0.00 | 27 | 245441 | **245441** | 0.00 | 27 | 245441 | **245441** | 0.00 | 27 |
| G3 | 57 | **229507** | X | TV | 229536 | **229507** | 0.01 | 30 | 230430 | **229507** | 0.40 | 30 | 230003 | **229507** | 0.22 | 30 |
| G4 | 57 | **232521** | - | EHP | 232784 | **232521** | 0.11 | 29 | 233767 | **232521** | 0.54 | 31 | 233649 | **232521** | 0.48 | 31 |
| G5 | 57 | **221730** | X | TV | 221805 | **221730** | 0.03 | 33 | 221771 | **221730** | 0.02 | 35 | 221730 | **221730** | 0.00 | 36 |
| G6 | 57 | **213457** | X | TV | 213562 | **213457** | 0.05 | 38 | 213457 | **213457** | 0.00 | 41 | 214084 | **213457** | 0.29 | 39 |
| H1 | 68 | **268933** | X | TV | 269701 | **268933** | 0.29 | 38 | 269276 | **268933** | 0.13 | 40 | 269371 | **268933** | 0.16 | 40 |
| H2 | 68 | **253365** | X | TV + EHP | 253414 | **253365** | 0.02 | 45 | 253437 | **253365** | 0.03 | 48 | 253365 | **253365** | 0.00 | 47 |
| H3 | 68 | **247449** | X | TV + EHP | 247684 | **247449** | 0.10 | 51 | 247474 | **247449** | 0.01 | 53 | 247475 | **247449** | 0.01 | 54 |
| H4 | 68 | **250221** | X | TV + EHP | 250244 | **250221** | 0.01 | 49 | 250221 | **250221** | 0.00 | 52 | 250295 | **250221** | 0.03 | 51 |
| H5 | 68 | **246121** | X | TV + EHP | 247300 | **246121** | 0.48 | 56 | 246170 | **246121** | 0.02 | 57 | 246140 | **246121** | 0.01 | 55 |
| H6 | 68 | **249135** | X | TV + EHP | 249397 | **249135** | 0.11 | 53 | 249246 | **249135** | 0.04 | 54 | 249246 | **249135** | 0.04 | 55 |
| I1 | 90 | 353021 | - | EHP | 351106 | 350437 | 0.25 | 52 | 350951 | **350246** | 0.20 | 52 | 351069 | 350801 | 0.24 | 52 |
| I2 | 90 | **309943** | X | EHP | 311714 | 309944 | 0.57 | 63 | 310738 | 309944 | 0.26 | 63 | 310846 | 309944 | 0.29 | 63 |
| I3 | 90 | 294833 | - | EHP | 296221 | **294507** | 0.58 | 80 | 294728 | **294507** | 0.07 | 84 | 294950 | **294507** | 0.15 | 81 |
| I4 | 90 | **295988** | - | EHP | 296889 | **295988** | 0.30 | 75 | 296172 | **295988** | 0.06 | 75 | 296374 | **295988** | 0.13 | 76 |
| I5 | 90 | **301226** | - | EHP | 302666 | 301236 | 0.48 | 71 | 301619 | 301236 | 0.13 | 74 | 302066 | 301236 | 0.28 | 73 |
| J1 | 95 | **335006** | - | EHP | 336598 | 335007 | 0.48 | 57 | 336475 | 335480 | 0.44 | 58 | 336347 | 335007 | 0.40 | 57 |
| J2 | 95 | 315644 | - | EHP | 311853 | **310417** | 0.46 | 65 | 311440 | **310417** | 0.33 | 66 | 310964 | **310417** | 0.18 | 67 |
| J3 | 95 | 282447 | - | EHP | 282335 | 280401 | 1.12 | 83 | 279801 | **279219** | 0.21 | 86 | 279468 | **279219** | 0.09 | 84 |
| J4 | 95 | 300548 | - | EHP | 298004 | 296773 | 0.50 | 72 | 297529 | **296533** | 0.34 | 74 | 297249 | **296533** | 0.24 | 72 |
| K1 | 113 | 394637 | - | EHP | 398657 | **394376** | 1.09 | 82 | 397183 | **394376** | 0.71 | 83 | 395965 | 394517 | 0.40 | 82 |
| K2 | 113 | 362360 | - | EHP | 364447 | **362130** | 0.64 | 96 | 363103 | **362130** | 0.27 | 98 | 363258 | **362130** | 0.31 | 95 |
| K3 | 113 | **365693** | - | EHP | 367725 | 365694 | 0.56 | 95 | 366549 | 365694 | 0.23 | 97 | 366698 | 365694 | 0.27 | 96 |
| K4 | 113 | 358308 | - | EHP | 352064 | **348950** | 0.89 | 108 | 349775 | **348950** | 0.24 | 109 | 349483 | **348950** | 0.15 | 107 |
| Tot. | | 12174445 | | | 12188672 | 12157811 | | 1832 | 12172829 | 12156670 | | 1902 | 12171434 | 12156893 | | 1881 |
| Avg. | | | | | | | 0.28 | 39 | | | 0.18 | 40 | | | 0.17 | 40 |
| < PB | | | | | | 8 | | | | 8 | | | | 8 | | |
| #B | | 39 | | | | 35 | | | | 38 | | | | 36 | | |

Table 16: *Goetschalckx* data set. The results have been produced by using distances rounded to one decimal and rounding the final result to an integer. This rounding scheme allows us to compare the LNS heuristics to the exact methods by Toth and Vigo [41] and Mingozzi et al. [26], we only report results on the instances that either Toth and Vigo or Mingozzi et al. attempted to solve. The table should be read like Table 3, notice that the row *<PB* should be interpretted like the *BTPB* row in Table 3.

| | Best known | | | Std. Removals | | | | 6R - no learning | | | | 6R - normal learning | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | n | cost | reference | avg. sol. | best sol. | avg. gap (%) | avg. time (s) | avg. sol. | best sol. | avg. gap (%) | avg. time (s) | avg. sol. | best sol. | avg. gap (%) | avg. time (s) |
| A1 | 25 | **229884** | TV | 229884 | **229884** | 0.00 | 7 | 229884 | **229884** | 0.00 | 8 | 229884 | **229884** | 0.00 | 8 |
| A2 | 25 | **180117** | TV | 180117 | **180117** | 0.00 | 8 | 180117 | **180117** | 0.00 | 9 | 180117 | **180117** | 0.00 | 8 |
| A3 | 25 | **163403** | TV | 163403 | **163403** | 0.00 | 9 | 163403 | **163403** | 0.00 | 10 | 163403 | **163403** | 0.00 | 10 |
| A4 | 25 | **155795** | TV | 155795 | **155795** | 0.00 | 10 | 155795 | **155795** | 0.00 | 11 | 155795 | **155795** | 0.00 | 11 |
| B1 | 30 | **239077** | TV | 239077 | **239077** | 0.00 | 9 | 239077 | **239077** | 0.00 | 10 | 239077 | **239077** | 0.00 | 9 |
| B2 | 30 | **198045** | TV | 198045 | **198045** | 0.00 | 10 | 198045 | **198045** | 0.00 | 11 | 198045 | **198045** | 0.00 | 11 |
| B3 | 30 | **169368** | TV | 169368 | **169368** | 0.00 | 13 | 169368 | **169368** | 0.00 | 15 | 169368 | **169368** | 0.00 | 15 |
| C1 | 40 | **250557** | TV | 250557 | **250557** | 0.00 | 13 | 250557 | **250557** | 0.00 | 14 | 250557 | **250557** | 0.00 | 14 |
| C2 | 40 | **215019** | TV | 215019 | **215019** | 0.00 | 15 | 215019 | **215019** | 0.00 | 17 | 215019 | **215019** | 0.00 | 17 |
| C3 | 40 | **199344** | TV | 199344 | **199344** | 0.00 | 18 | 199344 | **199344** | 0.00 | 20 | 199344 | **199344** | 0.00 | 21 |
| C4 | 40 | **195365** | TV | 195365 | **195365** | 0.00 | 18 | 195365 | **195365** | 0.00 | 20 | 195365 | **195365** | 0.00 | 20 |
| D1 | 38 | **322533** | TV | 322533 | **322533** | 0.00 | 11 | 322533 | **322533** | 0.00 | 13 | 322533 | **322533** | 0.00 | 13 |
| D2 | 38 | **316711** | TV | 316711 | **316711** | 0.00 | 11 | 316711 | **316711** | 0.00 | 13 | 316711 | **316711** | 0.00 | 12 |
| D3 | 38 | **239482** | TV | 239482 | **239482** | 0.00 | 12 | 239482 | **239482** | 0.00 | 14 | 239482 | **239482** | 0.00 | 13 |
| D4 | 38 | **205834** | TV | 205834 | **205834** | 0.00 | 15 | 205834 | **205834** | 0.00 | 16 | 205834 | **205834** | 0.00 | 16 |
| E1 | 45 | **238880** | TV | 238880 | **238880** | 0.00 | 17 | 238880 | **238880** | 0.00 | 19 | 238880 | **238880** | 0.00 | 19 |
| E2 | 45 | **212262** | TV | 212262 | **212262** | 0.00 | 23 | 212262 | **212262** | 0.00 | 24 | 212262 | **212262** | 0.00 | 24 |
| E3 | 45 | **206658** | TV | 206734 | **206658** | 0.04 | 25 | 206709 | **206658** | 0.02 | 28 | 206722 | **206658** | 0.03 | 28 |
| F1 | 60 | **263175** | TV | 268435 | 267061 | 2.00 | 30 | 267941 | 267061 | 1.81 | 31 | 268242 | 267061 | 1.93 | 31 |
| F2 | 60 | **265214** | TV | 265230 | **265214** | 0.01 | 29 | 265214 | **265214** | 0.00 | 30 | 265214 | **265214** | 0.00 | 30 |
| F3 | 60 | **241121** | OW | 242014 | 241970 | 0.37 | 36 | 241970 | 241970 | 0.35 | 38 | 241970 | 241970 | 0.35 | 37 |
| F4 | 60 | **233861** | TV | 235912 | 235178 | 0.88 | 42 | 235261 | 235178 | 0.60 | 45 | 235204 | 235178 | 0.57 | 44 |
| G1 | 57 | **306304** | OW | 306455 | **306304** | 0.05 | 22 | 306336 | **306304** | 0.01 | 23 | 306304 | **306304** | 0.00 | 24 |
| G2 | 57 | **245441** | TV | 245533 | **245441** | 0.04 | 28 | 245441 | **245441** | 0.00 | 29 | 245441 | **245441** | 0.00 | 29 |
| G3 | 57 | **229506** | OW | 229963 | **229506** | 0.20 | 32 | 230421 | **229506** | 0.40 | 33 | 230414 | **229506** | 0.40 | 32 |
| G4 | 57 | 232646 | TV | 233142 | **232519** | 0.27 | 31 | 233951 | **232519** | 0.62 | 33 | 233705 | **232519** | 0.51 | 33 |
| G5 | 57 | **221731** | OW | 221823 | **221731** | 0.04 | 36 | 221858 | **221731** | 0.06 | 38 | 221800 | **221731** | 0.03 | 39 |
| G6 | 57 | **213457** | TV | 213605 | **213457** | 0.07 | 41 | 213457 | **213457** | 0.00 | 43 | 213516 | **213457** | 0.03 | 43 |
| H1 | 68 | **268933** | OW | 269630 | **268933** | 0.26 | 40 | 269460 | **268933** | 0.20 | 43 | 269226 | **268933** | 0.11 | 42 |
| H2 | 68 | **253366** | TV | 253513 | **253366** | 0.06 | 48 | 253463 | **253366** | 0.04 | 50 | 253414 | **253366** | 0.02 | 50 |
| H3 | 68 | **247449** | TV | 247803 | **247449** | 0.14 | 54 | 247594 | **247449** | 0.06 | 57 | 247472 | **247449** | 0.01 | 57 |
| H4 | 68 | **250221** | TV | 250449 | **250221** | 0.09 | 51 | 250269 | **250221** | 0.02 | 56 | 250269 | **250221** | 0.02 | 55 |
| H5 | 68 | **246121** | TV | 246367 | **246121** | 0.10 | 57 | 246265 | **246121** | 0.06 | 61 | 246339 | **246121** | 0.09 | 60 |
| H6 | 68 | **249136** | TV | 249280 | **249136** | 0.06 | 54 | 249284 | **249136** | 0.06 | 60 | 249187 | **249136** | 0.02 | 59 |
| I1 | 90 | 351609 | OW | 351136 | 350437 | 0.25 | 54 | 350902 | **350248** | 0.19 | 55 | 350992 | **350248** | 0.21 | 55 |
| I2 | 90 | 309957 | OW | 312017 | **309946** | 0.67 | 66 | 311039 | **309946** | 0.35 | 66 | 310739 | **309946** | 0.26 | 66 |
| I3 | 90 | **294509** | OW | 295043 | **294509** | 0.18 | 86 | 294788 | **294509** | 0.09 | 88 | 294773 | **294509** | 0.09 | 88 |
| I4 | 90 | **295988** | TV | 296414 | **295988** | 0.14 | 79 | 296370 | **295988** | 0.13 | 80 | 296254 | **295988** | 0.09 | 84 |
| I5 | 90 | 302525 | OW | 302482 | **301238** | 0.41 | 75 | 301916 | **301238** | 0.23 | 78 | 302225 | **301238** | 0.33 | 80 |
| J1 | 95 | 335590 | OW | 336867 | **335004** | 0.56 | 60 | 336418 | 335478 | 0.42 | 61 | 336243 | **335004** | 0.37 | 60 |
| J2 | 95 | 310798 | OW | 312248 | **310417** | 0.59 | 70 | 311378 | **310417** | 0.31 | 71 | 311662 | **310417** | 0.40 | 70 |
| J3 | 95 | **279220** | OW | 281860 | 279307 | 0.95 | 90 | 279830 | **279220** | 0.22 | 91 | 279889 | **279220** | 0.24 | 92 |
| J4 | 95 | 296774 | OW | 297926 | 296861 | 0.47 | 77 | 297487 | **296533** | 0.32 | 79 | 297436 | **296533** | 0.30 | 78 |
| K1 | 113 | 395544 | OW | 397824 | 394511 | 0.88 | 87 | 396806 | 394458 | 0.62 | 88 | 395328 | **394369** | 0.24 | 87 |
| K2 | 113 | 363213 | OW | 365244 | 362358 | 0.86 | 102 | 363938 | **362128** | 0.50 | 103 | 363350 | **362128** | 0.34 | 102 |
| K3 | 113 | 366222 | OW | 368228 | **365693** | 0.69 | 100 | 366593 | **365693** | 0.25 | 102 | 366420 | **365693** | 0.20 | 101 |
| K4 | 113 | 349037 | OW | 351283 | **348947** | 0.67 | 115 | 349713 | **348947** | 0.22 | 116 | 349191 | **348947** | 0.07 | 114 |
| L1 | 150 | 426021 | OW | 427532 | **426014** | 0.36 | 162 | 427786 | 426283 | 0.42 | 162 | 428031 | 426178 | 0.47 | 160 |
| L2 | 150 | 402246 | OW | 402643 | **401231** | 0.35 | 196 | 401917 | 401426 | 0.17 | 192 | 401720 | **401231** | 0.12 | 188 |
| L3 | 150 | 403886 | OW | 404400 | **402681** | 0.43 | 189 | 402829 | **402681** | 0.04 | 189 | 402681 | **402681** | 0.00 | 187 |
| L4 | 150 | 384843 | OW | 388152 | **384635** | 0.91 | 221 | 384962 | **384635** | 0.08 | 219 | 385656 | **384635** | 0.27 | 218 |
| L5 | 150 | 388060 | OW | 392003 | **387563** | 1.15 | 216 | 388986 | **387563** | 0.37 | 215 | 388398 | **387563** | 0.22 | 214 |
| M1 | 125 | 400858 | OW | 403542 | **400085** | 0.86 | 109 | 402393 | 400660 | 0.58 | 109 | 402158 | 401076 | 0.52 | 108 |
| M2 | 125 | 398902 | OW | 400668 | 398712 | 0.81 | 109 | 400842 | 399263 | 0.85 | 109 | 400262 | **397448** | 0.71 | 106 |
| M3 | 125 | 377352 | OW | 379724 | 377139 | 0.70 | 123 | 378814 | 377399 | 0.46 | 121 | 378548 | **377093** | 0.39 | 121 |
| M4 | 125 | 348624 | OW | 349623 | 348604 | 0.31 | 149 | 349083 | **348530** | 0.16 | 149 | 349331 | **348530** | 0.23 | 146 |
| N1 | 150 | **408921** | OW | 416448 | 409897 | 1.84 | 166 | 414350 | 410046 | 1.33 | 165 | 413239 | 409506 | 1.06 | 163 |
| N2 | 150 | **409275** | OW | 415947 | 410232 | 1.63 | 167 | 415411 | 410232 | 1.50 | 163 | 415049 | 410616 | 1.41 | 163 |
| N3 | 150 | **396162** | OW | 401857 | 396870 | 1.44 | 185 | 401359 | 396825 | 1.31 | 182 | 400977 | 397546 | 1.22 | 180 |
| N4 | 150 | 397748 | OW | 402257 | 398293 | 1.89 | 180 | 399558 | **394785** | 1.21 | 180 | 401012 | 398667 | 1.58 | 181 |
| N5 | 150 | 376426 | OW | 380571 | 377081 | 1.90 | 229 | 375915 | **373471** | 0.65 | 227 | 378486 | 374553 | 1.34 | 222 |
| N6 | 150 | 377660 | OW | 379159 | 375646 | 1.45 | 221 | 378089 | **373752** | 1.16 | 226 | 376755 | 375348 | 0.80 | 225 |
| Tot. | | 18053986 | | 18130660 | 18051840 | | 4555 | 18096042 | 18044295 | | 4629 | 18092915 | 18048852 | | 4598 |
| Avg. | | | | | | 0.45 | 73 | | | 0.30 | 75 | | | 0.28 | 74 |
| < PB | | | | | 20 | | | | 20 | | | | 20 | | |
| #B | | 39 | | | 45 | | | | 49 | | | | 51 | | |

Table 17: *Goetschalckx* data set. The results have been produced by using distances rounded to integers. The results in the *best known* columns were found by the heuristics proposed by Toth and Vigo (TV) [40] and Osman and Wassan (OW) [28]. Notice that the TV and OW heuristics might have used a different rounding procedure, and consequently this table cannot be used to compare the LNS heuristics to the two earlier heuristics (see the text in the appendix). The table is provided for future reference only.

| | Best known | | Std. Removals | | | | 6R - no learning | | | | 6R - normal learning | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | cost | Reference | avg. sol. | best sol. | avg. gap (%) | avg. time (s) | avg. sol. | best sol. | avg. gap (%) | avg. time (s) | avg. sol. | best sol. | avg. gap (%) | avg. time (s) |
| CMT01T | 541 | NS | 520 | **520** | 0.00 | 32 | 520 | **520** | 0.00 | 34 | 520 | **520** | 0.00 | 34 |
| CMT02T | 839 | NS | 790 | **783** | 0.95 | 52 | 792 | 784 | 1.13 | 56 | 788 | **783** | 0.63 | 57 |
| CMT03T | 903 | NS | 805 | 801 | 0.83 | 104 | 804 | 801 | 0.72 | 110 | 803 | **798** | 0.65 | 109 |
| CMT04T | 1111 | NS | 1004 | **998** | 0.63 | 203 | 1004 | **998** | 0.61 | 213 | 1005 | 1000 | 0.73 | 212 |
| CMT05T | 1423 | NS | 1239 | 1231 | 0.97 | 323 | 1239 | 1232 | 0.95 | 334 | 1234 | **1227** | 0.57 | 333 |
| CMT06T | 571 | NS | 555 | **555** | 0.00 | 29 | 555 | **555** | 0.00 | 31 | 555 | **555** | 0.00 | 31 |
| CMT07T | - | - | 909 | **903** | 0.69 | 48 | 907 | **903** | 0.38 | 52 | 904 | **903** | 0.16 | 52 |
| CMT08T | 911 | NS | 869 | **866** | 0.43 | 91 | 868 | **866** | 0.33 | 94 | 866 | **866** | 0.10 | 95 |
| CMT09T | **1164** | NS | 1172 | 1166 | 0.75 | 173 | 1170 | **1164** | 0.56 | 179 | 1172 | **1164** | 0.67 | 178 |
| CMT10T | 1418 | NS | 1410 | 1398 | 1.09 | 285 | 1408 | **1395** | 0.93 | 291 | 1410 | 1402 | 1.05 | 291 |
| CMT11T | 1075 | NS | 1002 | **999** | 0.29 | 158 | 1001 | **999** | 0.24 | 163 | 1003 | 1000 | 0.39 | 164 |
| CMT12T | 827 | NS | 789 | **788** | 0.14 | 92 | 788 | **788** | 0.00 | 96 | 788 | **788** | 0.00 | 96 |
| CMT13T | 1600 | NS | 1550 | **1544** | 0.35 | 124 | 1548 | **1544** | 0.23 | 126 | 1547 | **1544** | 0.21 | 127 |
| CMT14T | 866 | NS | 827 | **827** | 0.06 | 84 | 827 | **827** | 0.00 | 86 | 827 | **827** | 0.00 | 86 |
| Tot. | 13249 | | 13442 | 13378 | | 1801 | 13430 | 13375 | | 1864 | 13422 | 13376 | | 1864 |
| Avg. | | | | | 0.51 | 129 | | | 0.43 | 133 | | | 0.37 | 133 |
| < PB | | | 13 | | | | 13 | | | | 13 | | | |
| #B | 1 | | 10 | | | | 11 | | | | 11 | | | |

Table 18: Nagy and Salhi MVRPB problems with 10% backhaul customers. The entries in the *Best known* columns are the best result reported by Nagy and Salhi (NS) [32] and Dethloff (D) [9]. It should be noted that Dethloff's heuristic only have been applied to half of the problems. No solution were given for problem 7 (this explains the dash in the table).

| | Best known | | Std. Removals | | | | 6R - no learning | | | | 6R - normal learning | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | cost | Reference | avg. sol. | best sol. | avg. gap (%) | avg. time (s) | avg. sol. | best sol. | avg. gap (%) | avg. time (s) | avg. sol. | best sol. | avg. gap (%) | avg. time (s) |
| CMT01Q | 557 | NS | 490 | **490** | 0.02 | 35 | 490 | **490** | 0.00 | 40 | 490 | **490** | 0.00 | 41 |
| CMT02Q | 860 | NS | 737 | **732** | 0.62 | 57 | 736 | 733 | 0.54 | 64 | 737 | 733 | 0.64 | 65 |
| CMT03Q | 918 | NS | 752 | **747** | 0.68 | 119 | 751 | **747** | 0.58 | 126 | 749 | **747** | 0.23 | 128 |
| CMT04Q | 1164 | NS | 922 | **916** | 0.60 | 228 | 921 | 918 | 0.58 | 244 | 922 | 918 | 0.59 | 244 |
| CMT05Q | 1477 | NS | 1133 | 1124 | 1.35 | 358 | 1127 | **1118** | 0.83 | 382 | 1124 | 1119 | 0.52 | 381 |
| CMT06Q | 594 | NS | 555 | **555** | 0.00 | 28 | 555 | **555** | 0.00 | 30 | 555 | **555** | 0.00 | 30 |
| CMT07Q | - | - | 905 | **901** | 0.44 | 48 | 903 | **901** | 0.26 | 52 | 902 | **901** | 0.17 | 53 |
| CMT08Q | 918 | NS | 868 | **866** | 0.25 | 90 | 867 | **866** | 0.23 | 93 | 866 | **866** | 0.10 | 93 |
| CMT09Q | 1178 | NS | 1170 | **1162** | 0.69 | 167 | 1170 | 1164 | 0.69 | 170 | 1169 | **1162** | 0.62 | 171 |
| CMT10Q | 1477 | NS | 1404 | 1394 | 1.06 | 280 | 1405 | 1398 | 1.11 | 285 | 1402 | **1389** | 0.91 | 288 |
| CMT11Q | 1075 | NS | 941 | **939** | 0.22 | 183 | 941 | **939** | 0.23 | 195 | 941 | **939** | 0.12 | 196 |
| CMT12Q | 843 | NS | 731 | **729** | 0.28 | 100 | 731 | **729** | 0.21 | 107 | 730 | **729** | 0.17 | 108 |
| CMT13Q | 1613 | NS | 1554 | 1545 | 0.67 | 117 | 1546 | 1544 | 0.14 | 120 | 1546 | **1543** | 0.14 | 120 |
| CMT14Q | 873 | NS | 822 | **822** | 0.00 | 84 | 822 | **822** | 0.00 | 85 | 822 | **822** | 0.00 | 85 |
| Tot. | 13547 | | 12983 | 12922 | | 1896 | 12966 | 12924 | | 1993 | 12954 | 12914 | | 2003 |
| Avg. | | | | | 0.49 | 135 | | | 0.38 | 142 | | | 0.30 | 143 |
| < PB | | | 14 | | | | 14 | | | | 14 | | | |
| #B | 0 | | 11 | | | | 9 | | | | 11 | | | |

Table 19: Nagy Salhi MVRPB problems with 25% backhaul customers. See Table 18 for a decription.

| | Best known | | Std. Removals | | | | 6R - no learning | | | | 6R - normal learning | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | cost | Reference | avg. sol. | best sol. | avg. gap (%) | avg. time (s) | avg. sol. | best sol. | avg. gap (%) | avg. time (s) | avg. sol. | best sol. | avg. gap (%) | avg. time (s) |
| CMT01H | 536 | D | 468 | 466 | 0.63 | 44 | 465 | **465** | 0.08 | 50 | 466 | **465** | 0.19 | 51 |
| CMT02H | 801 | D | 666 | **663** | 0.47 | 69 | 664 | **663** | 0.21 | 76 | 664 | **663** | 0.21 | 78 |
| CMT03H | 850 | D | 705 | **701** | 0.62 | 165 | 702 | **701** | 0.14 | 183 | 702 | **701** | 0.11 | 186 |
| CMT04H | 1099 | D | 842 | 835 | 1.57 | 306 | 840 | **829** | 1.27 | 346 | 840 | **829** | 1.24 | 345 |
| CMT05H | 1329 | D | 996 | 986 | 1.36 | 461 | 994 | 986 | 1.12 | 514 | 991 | **983** | 0.78 | 514 |
| CMT06H | 595 | NS | 555 | **555** | 0.00 | 29 | 555 | **555** | 0.00 | 31 | 555 | **555** | 0.00 | 31 |
| CMT07H | - | - | 904 | 901 | 0.40 | 49 | 902 | 901 | 0.23 | 52 | 903 | **900** | 0.29 | 54 |
| CMT08H | 915 | NS | 866 | **866** | 0.08 | 92 | 867 | **866** | 0.14 | 94 | 868 | **866** | 0.26 | 95 |
| CMT09H | 1164 | NS | 1169 | 1164 | 0.72 | 172 | 1171 | **1161** | 0.86 | 176 | 1169 | 1166 | 0.69 | 177 |
| CMT10H | 1509 | NS | 1406 | **1389** | 1.24 | 290 | 1406 | 1396 | 1.23 | 295 | 1401 | 1393 | 0.92 | 296 |
| CMT11H | 961 | D | 829 | **818** | 1.37 | 271 | 820 | **818** | 0.26 | 315 | 818 | **818** | 0.04 | 303 |
| CMT12H | 765 | D | 636 | 630 | 1.00 | 135 | 633 | **629** | 0.65 | 146 | 635 | **629** | 0.86 | 150 |
| CMT13H | 1546 | NS | 1552 | 1544 | 0.54 | 120 | 1545 | 1544 | 0.13 | 123 | 1546 | **1543** | 0.18 | 125 |
| CMT14H | 866 | NS | 822 | **822** | 0.00 | 87 | 822 | **822** | 0.00 | 89 | 822 | **822** | 0.00 | 89 |
| Tot. | 12936 | | 12416 | 12338 | | 2291 | 12387 | 12335 | | 2490 | 12379 | 12333 | | 2493 |
| Avg. | | | | | 0.71 | 164 | | | 0.45 | 178 | | | 0.41 | 178 |
| < PB | | | 13 | | | | 14 | | | | 13 | | | |
| #B | 0 | | 7 | | | | 10 | | | | 12 | | | |

Table 20: Nagy Salhi MVRPB problems with 50% backhaul customers. See Table 18 for a decription.

| | Best known | | Std. Removals | | | | 6R - no learning | | | | 6R - normal learning | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | cost | Reference | avg. sol. | best sol. | avg. gap (%) | avg. time (s) | avg. sol. | best sol. | avg. gap (%) | avg. time (s) | avg. sol. | best sol. | avg. gap (%) | avg. time (s) |
| GJ01T | 614 | NS | 570 | **569** | 0.12 | 30 | 569 | **569** | 0.00 | 34 | 569 | **569** | 0.00 | 35 |
| GJ02T | 497 | NS | 464 | **464** | 0.04 | 34 | 464 | **464** | 0.04 | 37 | 464 | **464** | 0.00 | 38 |
| GJ03T | 662 | NS | 627 | **624** | 0.34 | 60 | 626 | **624** | 0.29 | 65 | 626 | 625 | 0.19 | 64 |
| GJ04T | 1055 | NS | 976 | 972 | 1.43 | 80 | 969 | **962** | 0.75 | 85 | 971 | **962** | 0.92 | 86 |
| GJ05T | 794 | NS | 739 | 735 | 0.85 | 114 | 738 | **733** | 0.62 | 118 | 738 | **733** | 0.61 | 119 |
| GJ06T | 914 | NS | 859 | **851** | 0.90 | 85 | 853 | **851** | 0.21 | 90 | 852 | **851** | 0.16 | 91 |
| GJ07T | 992 | NS | 864 | **854** | 1.23 | 82 | 862 | 855 | 1.04 | 88 | 859 | **854** | 0.59 | 87 |
| GJ08T | 4674 | NS | 4183 | **4134** | 1.17 | 417 | 4170 | **4134** | 0.86 | 431 | 4179 | 4152 | 1.08 | 435 |
| GJ09T | 4087 | NS | 3727 | 3684 | 1.39 | 452 | 3718 | **3677** | 1.12 | 492 | 3716 | 3678 | 1.06 | 485 |
| GJ10T | 4002 | NS | 3540 | 3502 | 1.58 | 444 | 3524 | **3485** | 1.11 | 472 | 3516 | 3492 | 0.88 | 467 |
| GJ11T | 3794 | NS | 3428 | **3390** | 1.12 | 445 | 3421 | **3390** | 0.92 | 469 | 3432 | 3409 | 1.23 | 464 |
| Tot. | 22085 | | 19977 | 19780 | | 2243 | 19915 | 19745 | | 2382 | 19921 | 19789 | | 2371 |
| Avg. | 2008 | | | | 0.93 | 204 | | | 0.63 | 217 | | | 0.61 | 216 |
| < PB | | | 11 | | | | 11 | | | | 11 | | | |
| #B | 0 | | 7 | | | | 10 | | | | 6 | | | |

Table 21: Nagy Salhi MDMVRPB problems with 10% backhaul customers.

| | Best known | | Std. Removals | | | | 6R - no learning | | | | 6R - normal learning | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | cost | Reference | avg. sol. | best sol. | avg. gap (%) | avg. time (s) | avg. sol. | best sol. | avg. gap (%) | avg. time (s) | avg. sol. | best sol. | avg. gap (%) | avg. time (s) |
| GJ01Q | 666 | NS | 529 | **528** | 0.04 | 32 | 528 | **528** | 0.00 | 36 | 528 | **528** | 0.00 | 38 |
| GJ02Q | 550 | NS | 451 | **450** | 0.34 | 38 | 451 | **450** | 0.27 | 43 | 451 | **450** | 0.39 | 44 |
| GJ03Q | 670 | NS | 607 | **605** | 0.26 | 64 | 608 | **605** | 0.40 | 71 | 607 | **605** | 0.36 | 72 |
| GJ04Q | 1168 | NS | 879 | 876 | 0.47 | 87 | 876 | **875** | 0.13 | 94 | 880 | 876 | 0.55 | 95 |
| GJ05Q | 828 | NS | 705 | **700** | 0.72 | 124 | 705 | 702 | 0.65 | 133 | 706 | 703 | 0.83 | 134 |
| GJ06Q | 978 | NS | 805 | **794** | 1.39 | 92 | 800 | **794** | 0.78 | 100 | 799 | **794** | 0.60 | 100 |
| GJ07Q | 940 | NS | 808 | 803 | 0.69 | 89 | 807 | 803 | 0.51 | 94 | 806 | **802** | 0.45 | 95 |
| GJ08Q | 4877 | NS | 3826 | 3799 | 1.72 | 449 | 3810 | 3774 | 1.29 | 479 | 3792 | **3762** | 0.80 | 478 |
| GJ09Q | 4087 | NS | 3433 | 3391 | 2.31 | 482 | 3393 | **3355** | 1.13 | 535 | 3394 | 3362 | 1.15 | 535 |
| GJ10Q | 3931 | NS | 3294 | 3259 | 1.61 | 477 | 3267 | 3245 | 0.79 | 513 | 3276 | **3242** | 1.04 | 510 |
| GJ11Q | 3840 | NS | 3191 | 3171 | 1.15 | 472 | 3192 | 3165 | 1.17 | 511 | 3189 | **3155** | 1.10 | 505 |
| Tot. | 22535 | | 18528 | 18375 | | 2407 | 18437 | 18296 | | 2609 | 18428 | 18279 | | 2608 |
| Avg. | 2049 | | | | 0.97 | 219 | | | 0.65 | 237 | | | 0.66 | 237 |
| < PB | | | 11 | | | | 11 | | | | 11 | | | |
| #B | 0 | | 5 | | | | 6 | | | | 8 | | | |

Table 22: Nagy Salhi MDMVRPB problems with 25% backhaul customers.

| | Best known | | Std. Removals | | | | 6R - no learning | | | | 6R - normal learning | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | cost | Reference | avg. sol. | best sol. | avg. gap (%) | avg. time (s) | avg. sol. | best sol. | avg. gap (%) | avg. time (s) | avg. sol. | best sol. | avg. gap (%) | avg. time (s) |
| GJ01H | 619 | NS | 499 | **499** | 0.06 | 36 | 499 | **499** | 0.04 | 40 | 499 | **499** | 0.00 | 42 |
| GJ02H | 562 | NS | 440 | **440** | 0.00 | 44 | 440 | **440** | 0.00 | 51 | 440 | **440** | 0.00 | 53 |
| GJ03H | 662 | NS | 584 | **581** | 0.60 | 73 | 583 | **581** | 0.40 | 81 | 583 | **581** | 0.35 | 82 |
| GJ04H | 1055 | NS | 795 | **789** | 0.73 | 102 | 797 | 790 | 0.91 | 112 | 796 | 790 | 0.84 | 114 |
| GJ05H | 853 | NS | 681 | **678** | 0.50 | 154 | 680 | **678** | 0.28 | 168 | 680 | **678** | 0.27 | 171 |
| GJ06H | 1034 | NS | 753 | 748 | 1.06 | 106 | 751 | 747 | 0.80 | 116 | 751 | **745** | 0.91 | 118 |
| GJ07H | 932 | NS | 739 | **733** | 0.88 | 107 | 734 | **733** | 0.23 | 117 | 735 | **733** | 0.29 | 113 |
| GJ08H | 5188 | NS | 3391 | 3370 | 1.92 | 530 | 3373 | **3327** | 1.38 | 581 | 3371 | 3342 | 1.31 | 577 |
| GJ09H | 4087 | NS | 3043 | **3005** | 1.27 | 582 | 3028 | 3006 | 0.78 | 646 | 3027 | 3008 | 0.75 | 650 |
| GJ10H | 4041 | NS | 2961 | 2931 | 1.16 | 547 | 2963 | 2930 | 1.21 | 644 | 2962 | **2927** | 1.19 | 637 |
| GJ11H | 3933 | NS | 2898 | **2855** | 1.49 | 557 | 2905 | 2880 | 1.74 | 609 | 2893 | 2859 | 1.33 | 606 |
| Tot. | 22966 | | 16785 | 16630 | | 2841 | 16753 | 16611 | | 3166 | 16738 | 16601 | | 3163 |
| Avg. | 2088 | | | | 0.88 | 258 | | | 0.71 | 288 | | | 0.66 | 288 |
| < PB | | | 11 | | | | 11 | | | | 11 | | | |
| #B | 0 | | 8 | | | | 6 | | | | 7 | | | |

Table 23: Nagy Salhi MDMVRPB problems with 50% backhaul customers.

| | Best known | | | | | Std. Removals | | | | 6R - no learning | | | | 6R - normal learning | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | % BH | n | m | cost | ref | avg. #veh. | best sol. | best #veh. | avg. time (s) | avg. #veh. | best sol. | best #veh. | avg. time (s) | avg. #veh. | best sol. | best #veh. | avg. time (s) |
| BHR1DO.10 | 10% | 250 | 49 | 5085 | TPS | 46.0 | 4848.2 | 46 | 556 | 46.0 | 4844.8 | 46 | 571 | 46.0 | **4843.9** | **46** | 586 |
| BHR1DO.30 | 20% | 250 | 48 | 5243 | TPS | 45.0 | 5074.2 | 45 | 502 | 45.0 | **5062.7** | **45** | 512 | 45.0 | 5066.9 | 45 | 528 |
| BHR1DO.50 | 50% | 250 | 52 | 5403.1 | TPS | 49.0 | 5122.6 | 49 | 508 | 49.0 | **5107.1** | **49** | 531 | 49.0 | 5113.7 | 49 | 541 |
| BHR1UP.10 | 10% | 250 | 39 | 4278.6 | TPS | 32.0 | 3942.3 | 32 | 514 | 31.8 | **4056.9** | **31** | 503 | 32.0 | 3943.1 | 32 | 530 |
| BHR1UP.30 | 30% | 250 | 41 | 4715.2 | TPS | 35.0 | 4448.9 | 35 | 475 | 35.0 | 4427.8 | 35 | 474 | 34.8 | **4549.7** | **34** | 488 |
| BHR1UP.50 | 50% | 250 | 43 | 4921.4 | TPS | 36.0 | 4443.7 | 36 | 465 | 35.6 | **4618.4** | **35** | 473 | 36.0 | 4442.2 | 36 | 476 |
| BHRC1DO.10 | 10% | 250 | 39 | 4613.4 | TPS | 33.0 | 4116.8 | 33 | 506 | 32.8 | 4310.4 | 32 | 500 | 32.6 | **4211.6** | **32** | 519 |
| BHRC1DO.30 | 20% | 250 | 41 | 4852.2 | TPS | 34.4 | **4506.3** | **34** | 466 | 34.2 | 4534.4 | 34 | 466 | 34.2 | 4526.2 | 34 | 478 |
| BHRC1DO.50 | 50% | 250 | 41 | 4329.4 | TPS | 35.0 | 4500.2 | 35 | 456 | 34.4 | **4513.9** | **34** | 458 | 34.6 | 4589.6 | 34 | 463 |
| BHRC1UP.10 | 10% | 250 | 40 | 4445.8 | TPS | 33.4 | 4160.9 | 33 | 497 | 33.0 | 4137.0 | 33 | 488 | 33.4 | **4105.3** | **33** | 506 |
| BHRC1UP.30 | 30% | 250 | 43 | 4722.4 | TPS | 36.0 | 4485.2 | 36 | 466 | 35.0 | **4538.0** | **35** | 459 | 35.4 | 4555.8 | 35 | 469 |
| BHRC1UP.50 | 50% | 250 | 41 | 4899.4 | TPS | 35.6 | 4605.8 | 35 | 455 | 35.6 | 4558.5 | 35 | 464 | 35.2 | **4550.2** | **35** | 464 |
| Tot. | | | 517 | 57509 | | 450.4 | 54255.1 | 449 | 5868 | 447.4 | 54710.0 | 444 | 5899 | 448.2 | 54498.3 | 445 | 6050 |
| Avg. | | | | | | | | | 489 | | | | 492 | | | | 504 |
| < PB | | | | | | 12 | | | | 12 | | | | 12 | | | |
| #B | | | 0 | | | 1 | | | | 6 | | | | 5 | | | |

Table 24: Thangiah et al. 250 customer VRPBTW instances. The previously best known results have been found in [38] (TPS).

| | Best known | | | | | Std. Removals | | | | 6R - no learning | | | | 6R - normal learning | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | % BH | n | m | cost | ref | avg. #veh. | best sol. | best #veh. | avg. time (s) | avg. #veh. | best sol. | best #veh. | avg. time (s) | avg. #veh. | best sol. | best #veh. | avg. time (s) |
| BHR1DO.10 | 10% | 500 | 67 | 7620.4 | TPS | 58.4 | 6899.9 | 58 | 1726 | 58.0 | **6860.2** | **58** | 1691 | 58.0 | 6868.0 | 58 | 1763 |
| BHR1DO.30 | 20% | 500 | 69 | 9020.2 | TPS | 59.4 | 7320.8 | 59 | 1555 | 58.8 | **7337.2** | **58** | 1557 | 59.0 | 7262.3 | 59 | 1595 |
| BHR1DO.50 | 50% | 500 | 76 | 8376.5 | TPS | 61.8 | 7342.7 | 61 | 1554 | 61.0 | 7342.4 | 61 | 1575 | 60.8 | **7294.7** | **60** | 1584 |
| BHR1UP.10 | 10% | 500 | 64 | 7267.2 | TPS | 55.0 | 6776.6 | 54 | 1660 | 55.0 | **6702.7** | **54** | 1378 | 54.6 | 6784.7 | 54 | 1692 |
| BHR1UP.30 | 30% | 500 | 73 | 7926.6 | TPS | 57.8 | 7243.0 | 57 | 1533 | 57.8 | 7055.0 | 57 | 1679 | 57.6 | **6991.0** | **57** | 1566 |
| BHR1UP.50 | 50% | 500 | 68 | 8043.7 | TPS | 59.4 | 7119.1 | 59 | 1500 | 59.0 | 7126.2 | 59 | 1741 | 58.6 | **7217.3** | **58** | 1548 |
| BHRC1DO.10 | 10% | 500 | 61 | 7099.4 | TPS | 52.2 | 6362.6 | 52 | 1652 | 52.2 | 6346.8 | 52 | 1814 | 52.2 | **6313.3** | **52** | 1658 |
| BHRC1DO.30 | 20% | 500 | 63 | 7707.1 | TPS | 54.8 | 6959.3 | 54 | 1511 | 54.8 | 6889.0 | 54 | 1703 | 54.4 | **6813.6** | **54** | 1530 |
| BHRC1DO.50 | 50% | 500 | 65 | 7771.6 | TPS | 55.0 | 6983.7 | 54 | 1503 | 54.8 | 6914.5 | 54 | 1727 | 54.4 | **6896.5** | **54** | 1520 |
| BHRC1UP.10 | 10% | 500 | 63 | 7209.4 | TPS | 55.8 | 6493.5 | 55 | 1584 | 55.2 | 6483.4 | 55 | 1622 | 55.2 | **6464.1** | **55** | 1591 |
| BHRC1UP.30 | 30% | 500 | 63 | 7967.1 | TPS | 58.0 | 7030.2 | 57 | 1476 | 58.0 | 6918.8 | 58 | 1628 | 58.0 | **7028.3** | **57** | 1500 |
| BHRC1UP.50 | 50% | 500 | 67 | 8135.1 | TPS | 57.4 | 6965.8 | 57 | 1486 | 56.6 | **6969.6** | **56** | 1701 | 57.2 | 6862.3 | 57 | 1296 |
| Tot. | | | 799 | 94144 | | 685.0 | 83497.3 | 677 | 18742 | 681.2 | 82945.7 | 676 | 19815 | 680.0 | 82796.0 | 675 | 18843 |
| Avg. | | | | | | | | | 1562 | | | | 1651 | | | | 1570 |
| < PB | | | | | | 12 | | | | 12 | | | | 12 | | | |
| #B | | | 0 | | | 0 | | | | 4 | | | | 8 | | | |

Table 25: Thangiah et al. 500 customer VRPBTW instances. The previously best known results are the solutions found by Thangiah et al. (TPS) [38].

| | Best known | | | Std. Removals | | | | | | 6R - no learning | | | | | | 6R - normal learning | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | veh. | cost | Reference | avg. sol. | avg. #veh. | best sol. | best #veh. | avg. gap (%) | avg. time (s) | avg. sol. | avg. #veh. | best sol. | best #veh. | avg. gap (%) | avg. time (s) | avg. sol. | avg. #veh. | best sol. | best #veh. | avg. gap (%) | avg. time (s) |
| MC201 | 5 | 763.88 | ZC | 774.59 | 4.0 | **766.82** | **4** | 1.01 | 140 | 769.96 | 4.0 | **766.82** | **4** | 0.41 | 141 | 766.82 | 4.0 | **766.82** | **4** | 0.00 | 144 |
| MC202 | 4 | 1186.24 | ZC | 736.76 | 4.0 | **732.93** | **4** | 0.52 | 165 | 734.85 | 4.0 | **732.93** | **4** | 0.26 | 171 | 737.51 | 4.0 | **732.93** | **4** | 0.63 | 165 |
| MC203 | 4 | 1096.31 | ZC | 710.14 | 4.0 | 705.86 | 4 | 0.80 | 171 | 708.68 | 4.0 | 707.75 | 4 | 0.60 | 177 | 708.48 | 4.0 | **704.49** | **4** | 0.57 | 174 |
| MC204 | 4 | 885.73 | ZC | 677.75 | 4.0 | **676.18** | **4** | 0.23 | 188 | 679.06 | 4.0 | **676.18** | **4** | 0.43 | 193 | 678.54 | 4.0 | **676.18** | **4** | 0.35 | 189 |
| MC205 | 5 | 781.7 | ZC | 754.81 | 4.0 | **748.34** | **4** | 0.86 | 152 | 755.72 | 4.0 | 751.96 | 4 | 0.99 | 153 | 758.83 | 4.0 | 751.96 | 4 | 1.40 | 152 |
| MC206 | 5 | 860.74 | ZC | 750.16 | 4.0 | 748.17 | 4 | 0.41 | 157 | 748.33 | 4.0 | **747.08** | **4** | 0.17 | 158 | 748.09 | 4.0 | **747.08** | **4** | 0.14 | 157 |
| MC207 | 5 | 792.96 | ZC | 745.38 | 4.0 | **737.39** | **4** | 1.08 | 161 | 745.43 | 4.0 | **737.39** | **4** | 1.09 | 164 | 745.57 | 4.0 | 738.70 | 4 | 1.11 | 162 |
| MC208 | 5 | 859.92 | ZC | 736.19 | 4.0 | **735.17** | **4** | 0.14 | 161 | 741.69 | 4.0 | 738.70 | 4 | 0.89 | 164 | 742.76 | 4.0 | 738.70 | 4 | 1.03 | 162 |
| Tot. | 37 | 7227 | | 5885.79 | 32.00 | 5850.87 | 32 | | 1295 | 5883.72 | 32.00 | 5858.82 | 32 | | 1320 | 5886.63 | 32.00 | 5856.87 | 32 | | 1305 |
| Avg. | 5 | | | | | | | 0.63 | 162 | | | | | 0.60 | 165 | | | | | 0.65 | 163 |
| < PB | | | | 8 | | | | | | 8 | | | | | | 8 | | | | | |
| #B | | 0 | | 6 | | | | | | 5 | | | | | | 5 | | | | | |

Table 26: Kontoravdis and Bard's MVRPBTW instances. C-type problems. The previously best known results are the solutions found by Zhong and Cole (ZC) [48]. Kontoravdis and Bard [23] do not give detailed information about their solutions.

| | Best known | | | Std. Removals | | | | | | 6R - no learning | | | | | | 6R - normal learning | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | veh. | cost | Reference | avg. sol. | avg. #veh. | best sol. | best #veh. | avg. gap (%) | avg. time (s) | avg. sol. | avg. #veh. | best sol. | best #veh. | avg. gap (%) | avg. time (s) | avg. sol. | avg. #veh. | best sol. | best #veh. | avg. gap (%) | avg. time (s) |
| MR201 | 4 | 1388.73 | ZC | 1272.20 | 4.0 | 1260.48 | 4 | 1.27 | 157 | 1263.64 | 4.0 | **1256.31** | **4** | 0.58 | 165 | 1261.90 | 4.0 | **1256.31** | **4** | 0.45 | 160 |
| MR202 | 4 | 1198.99 | ZC | 1101.54 | 4.0 | 1092.01 | 4 | 1.39 | 359 | 1092.08 | 4.0 | **1086.46** | **4** | 0.52 | 371 | 1092.36 | 4.0 | **1086.46** | **4** | 0.54 | 362 |
| MR203 | 4 | 988.82 | ZC | 913.57 | 4.0 | **894.54** | **4** | 2.13 | 387 | 900.08 | 4.0 | 896.14 | 4 | 0.62 | 383 | 899.59 | 4.0 | 896.14 | 4 | 0.56 | 374 |
| MR204 | 4 | 858.32 | ZC | 739.43 | 4.0 | 737.51 | 4 | 0.36 | 419 | 737.87 | 4.0 | 737.51 | 4 | 0.15 | 436 | 738.63 | 4.0 | **736.75** | **4** | 0.25 | 432 |
| MR205 | 4 | 1172.53 | ZC | 994.25 | 4.0 | **974.26** | **4** | 2.05 | 351 | 994.86 | 4.0 | **974.26** | **4** | 2.11 | 367 | 989.36 | 4.0 | **974.26** | **4** | 1.55 | 353 |
| MR206 | 4 | 979.5 | ZC | 910.42 | 4.0 | 897.03 | 4 | 1.83 | 386 | 896.47 | 4.0 | 894.05 | 4 | 0.27 | 397 | 894.25 | 4.0 | **894.04** | **4** | 0.02 | 388 |
| MR207 | 4 | 912.69 | ZC | 811.92 | 4.0 | **800.79** | **4** | 1.39 | 421 | 800.79 | 4.0 | **800.79** | **4** | 0.00 | 426 | 800.79 | 4.0 | **800.79** | **4** | 0.00 | 422 |
| MR208 | 4 | 764.52 | ZC | 722.34 | 4.0 | 719.12 | 4 | 0.85 | 412 | 719.05 | 4.0 | **716.28** | **4** | 0.39 | 435 | 718.91 | 4.0 | **716.28** | **4** | 0.37 | 431 |
| MR209 | 4 | 978.82 | ZC | 894.18 | 4.0 | **879.63** | **4** | 1.65 | 354 | 886.97 | 4.0 | **879.63** | **4** | 0.83 | 371 | 893.49 | 4.0 | 881.60 | 4 | 1.58 | 361 |
| MR210 | 4 | 1061.36 | ZC | 936.45 | 4.0 | 930.92 | 4 | 1.29 | 361 | 929.49 | 4.0 | **924.56** | **4** | 0.53 | 377 | 928.71 | 4.0 | **924.56** | **4** | 0.45 | 369 |
| MR211 | 4 | 878.81 | ZC | 767.51 | 4.0 | 763.54 | 4 | 0.58 | 376 | 770.42 | 4.0 | **763.09** | **4** | 0.96 | 400 | 772.21 | 4.0 | 765.03 | 4 | 1.19 | 394 |
| Tot. | 44 | 11183 | | 10063.80 | 44.00 | 9949.83 | 44 | | 3983 | 9991.73 | 44.00 | 9929.07 | 44 | | 4128 | 9990.20 | 44.00 | 9932.21 | 44 | | 4046 |
| Avg. | 4 | | | | | | | 1.34 | 362 | | | | | 0.63 | 375 | | | | | 0.63 | 368 |
| < PB | | | | | | 11 | | | | | | 11 | | | | | | 11 | | | |
| #B | | 0 | | | | 4 | | | | | | 8 | | | | | | 8 | | | |

Table 27: Kontoravdis and Bard's MVRPBTW instances. R-type problems.

| | Best known | | | Std. Removals | | | | | | 6R - no learning | | | | | | 6R - normal learning | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | veh. | cost | Reference | avg. sol. | avg. #veh. | best sol. | best #veh. | avg. gap (%) | avg. time (s) | avg. sol. | avg. #veh. | best sol. | best #veh. | avg. gap (%) | avg. time (s) | avg. sol. | avg. #veh. | best sol. | best #veh. | avg. gap (%) | avg. time (s) |
| MRC201 | 5 | 1498.9 | ZC | 1370.16 | 5.0 | **1346.30** | **5** | 1.77 | 234 | 1357.26 | 5.0 | 1355.42 | 5 | 0.81 | 238 | 1356.84 | 5.0 | 1355.63 | 5 | 0.78 | 236 |
| MRC202 | 4 | 1539.41 | ZC | 1263.92 | 4.0 | **1230.24** | **4** | 2.74 | 171 | 1261.06 | 4.0 | 1241.77 | 4 | 2.51 | 174 | 1250.88 | 4.0 | **1230.24** | **4** | 1.68 | 171 |
| MRC203 | 4 | 1303.48 | ZC | 1020.29 | 4.0 | 997.06 | 4 | 2.48 | 177 | 1004.33 | 4.0 | **995.63** | **4** | 0.87 | 175 | 999.79 | 4.0 | **995.63** | **4** | 0.42 | 176 |
| MRC204 | 4 | 932.48 | ZC | 843.99 | 4.0 | **833.60** | **4** | 1.25 | 187 | 844.08 | 4.0 | 835.13 | 4 | 1.26 | 188 | 846.01 | 4.0 | 836.89 | 4 | 1.49 | 187 |
| MRC205 | 4 | 1632.04 | ZC | 1461.20 | 4.0 | 1417.14 | 4 | 3.30 | 168 | 1449.27 | 4.0 | 1419.07 | 4 | 2.46 | 166 | 1452.19 | 4.0 | **1414.52** | **4** | 2.66 | 160 |
| MRC206 | 4 | 1433.43 | ZC | 1286.51 | 4.0 | **1231.52** | **4** | 4.47 | 168 | 1277.35 | 4.0 | 1249.48 | 4 | 3.72 | 170 | 1291.85 | 4.0 | 1254.51 | 4 | 4.90 | 166 |
| MRC207 | 4 | 1217.2 | ZC | 1119.23 | 4.0 | 1096.06 | 4 | 3.31 | 175 | 1109.06 | 4.0 | 1084.81 | 4 | 2.37 | 174 | 1101.21 | 4.0 | **1083.33** | **4** | 1.65 | 169 |
| MRC208 | 4 | 1085.57 | ZC | 875.86 | 4.0 | **847.46** | **4** | 3.35 | 180 | 863.90 | 4.0 | 852.25 | 4 | 1.94 | 182 | 851.50 | 4.0 | 849.30 | 4 | 0.48 | 175 |
| Tot. | 33 | 10643 | | 9241.15 | 33.00 | 8999.36 | 33 | | 1461 | 9166.31 | 33.00 | 9033.57 | 33 | | 1467 | 9150.29 | 33.00 | 9020.05 | 33 | | 1441 |
| Avg. | 4 | | | | | | | 2.83 | 183 | | | | | 1.99 | 183 | | | | | 1.76 | 180 |
| < PB | | | | | | 8 | | | | | | 8 | | | | | | 8 | | | |
| #B | | 0 | | | | 5 | | | | | | 1 | | | | | | 4 | | | |

Table 28: Kontoravdis and Bard's MVRPBTW instances. RC-type problems.

| | Best known | | | Std. Removals | | | | 6R - no learning | | | | 6R - normal learning | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | n | cost | reference | avg. sol. | best sol. | avg. gap (%) | avg. time (s) | avg. sol. | best sol. | avg. gap (%) | avg. time (s) | avg. sol. | best sol. | avg. gap (%) | avg. time (s) |
| Min | 22 | **88** | H | 88.3 | **88.0** | 0.34 | 37 | 88.5 | **88.0** | 0.57 | 44 | 88.5 | **88.0** | 0.57 | 50 |
| SCA3-0 | 50 | 689 | D | 640.6 | 640.5 | 0.71 | 173 | 641.1 | 640.5 | 0.79 | 173 | 638.3 | **636.1** | 0.35 | 232 |
| SCA3-1 | 50 | 765.6 | D | 698.4 | **697.8** | 0.08 | 170 | 698.8 | **697.8** | 0.14 | 173 | 697.8 | **697.8** | 0.00 | 218 |
| SCA3-2 | 50 | 742.8 | D | 659.3 | **659.3** | 0.00 | 161 | 660.2 | **659.3** | 0.14 | 160 | 659.3 | **659.3** | 0.00 | 203 |
| SCA3-3 | 50 | 737.2 | D | 681.4 | **680.6** | 0.12 | 182 | 682.7 | 681.3 | 0.31 | 179 | 681.4 | **680.6** | 0.11 | 241 |
| SCA3-4 | 50 | 747.1 | D | 691.2 | **690.5** | 0.11 | 160 | 693.1 | **690.5** | 0.38 | 166 | 691.0 | **690.5** | 0.08 | 208 |
| SCA3-5 | 50 | 784.4 | D | 662.2 | **659.9** | 0.34 | 178 | 660.5 | **659.9** | 0.10 | 179 | 659.9 | **659.9** | 0.00 | 226 |
| SCA3-6 | 50 | 720.4 | D | 651.3 | **651.1** | 0.04 | 179 | 652.1 | **651.1** | 0.15 | 171 | 651.3 | **651.1** | 0.04 | 233 |
| SCA3-7 | 50 | 707.9 | D | 667.9 | **666.1** | 0.27 | 169 | 667.0 | **666.1** | 0.13 | 162 | 667.0 | **666.1** | 0.13 | 206 |
| SCA3-8 | 50 | 807.2 | D | 721.3 | **719.5** | 0.26 | 167 | 719.5 | **719.5** | 0.00 | 157 | 719.5 | **719.5** | 0.00 | 190 |
| SCA3-9 | 50 | 764.1 | D | 681.0 | **681.0** | 0.01 | 171 | 681.0 | **681.0** | 0.01 | 167 | 681.0 | **681.0** | 0.00 | 220 |
| SCA8-0 | 50 | 1132.9 | D | 991.1 | 982.2 | 1.63 | 82 | 993.2 | 987.9 | 1.85 | 94 | 986.3 | **975.1** | 1.15 | 98 |
| SCA8-1 | 50 | 1150.9 | D | 1083.1 | 1072.8 | 2.92 | 82 | 1082.6 | 1068.8 | 2.87 | 94 | 1066.5 | **1052.4** | 1.35 | 95 |
| SCA8-2 | 50 | 1100.8 | D | 1046.3 | **1039.6** | 0.64 | 83 | 1049.9 | 1044.5 | 0.99 | 87 | 1049.2 | 1044.5 | 0.92 | 94 |
| SCA8-3 | 50 | 1115.6 | D | 1016.5 | 1007.8 | 2.49 | 85 | 1012.5 | **991.8** | 2.08 | 91 | 1006.3 | 999.1 | 1.45 | 94 |
| SCA8-4 | 50 | 1235.4 | D | 1067.4 | **1065.5** | 0.18 | 84 | 1067.0 | **1065.5** | 0.14 | 87 | 1065.6 | **1065.5** | 0.01 | 93 |
| SCA8-5 | 50 | 1231.6 | D | 1052.8 | 1039.6 | 2.50 | 84 | 1047.9 | 1040.4 | 2.02 | 89 | 1039.9 | **1027.1** | 1.24 | 96 |
| SCA8-6 | 50 | 1062.5 | D | 996.2 | 986.0 | 2.44 | 82 | 987.5 | **972.5** | 1.54 | 93 | 983.5 | 977.0 | 1.14 | 94 |
| SCA8-7 | 50 | 1217.4 | D | 1067.1 | 1062.2 | 0.57 | 82 | 1068.3 | 1063.2 | 0.69 | 88 | 1065.8 | **1061.0** | 0.45 | 92 |
| SCA8-8 | 50 | 1231.6 | D | 1086.4 | **1071.2** | 1.42 | 85 | 1084.3 | 1077.7 | 1.22 | 93 | 1078.8 | **1071.2** | 0.71 | 98 |
| SCA8-9 | 50 | 1185.6 | D | 1077.0 | 1067.3 | 1.55 | 82 | 1068.8 | **1060.5** | 0.79 | 86 | 1064.7 | **1060.5** | 0.40 | 92 |
| CON3-0 | 50 | 672.4 | D | 623.4 | 617.6 | 1.11 | 173 | 621.5 | **616.5** | 0.81 | 171 | 619.0 | **616.5** | 0.40 | 215 |
| CON3-1 | 50 | 570.6 | D | 558.1 | **554.5** | 0.65 | 190 | 555.5 | **554.5** | 0.18 | 190 | 554.5 | **554.5** | 0.00 | 245 |
| CON3-2 | 50 | 534.8 | D | 522.3 | **521.4** | 0.18 | 176 | 523.0 | **521.4** | 0.32 | 177 | 521.6 | **521.4** | 0.05 | 232 |
| CON3-3 | 50 | 656.9 | D | 591.2 | **591.2** | 0.00 | 185 | 591.2 | **591.2** | 0.00 | 177 | 591.2 | **591.2** | 0.00 | 231 |
| CON3-4 | 50 | 640.2 | D | 591.7 | **588.8** | 0.49 | 187 | 590.5 | **588.8** | 0.29 | 173 | 590.0 | **588.8** | 0.21 | 221 |
| CON3-5 | 50 | 604.7 | D | 566.3 | **563.7** | 0.47 | 181 | 567.3 | **563.7** | 0.64 | 179 | 564.4 | **563.7** | 0.12 | 209 |
| CON3-6 | 50 | 521.3 | D | 501.6 | **499.1** | 0.51 | 195 | 503.0 | 501.8 | 0.78 | 180 | 501.9 | 500.8 | 0.57 | 225 |
| CON3-7 | 50 | 602.8 | D | 579.7 | 577.5 | 0.56 | 178 | 584.1 | 578.4 | 1.32 | 181 | 579.5 | **576.5** | 0.53 | 227 |
| CON3-8 | 50 | 556.2 | D | 523.5 | **523.1** | 0.08 | 186 | 523.7 | **523.1** | 0.12 | 174 | 523.5 | **523.1** | 0.08 | 237 |
| CON3-9 | 50 | 612.8 | D | 585.9 | **578.2** | 1.32 | 175 | 587.4 | **578.2** | 1.58 | 163 | 588.2 | 586.4 | 1.71 | 207 |
| CON8-0 | 50 | 967.3 | D | 867.8 | **857.2** | 1.24 | 86 | 867.7 | 858.0 | 1.22 | 87 | 860.9 | **857.2** | 0.43 | 94 |
| CON8-1 | 50 | 828.7 | D | 761.0 | **740.9** | 2.72 | 85 | 754.2 | 741.7 | 1.81 | 92 | 750.5 | **740.9** | 1.30 | 94 |
| CON8-2 | 50 | 770.2 | D | 731.3 | 719.3 | 2.14 | 85 | 728.8 | 718.3 | 1.79 | 93 | 721.4 | **716.0** | 0.75 | 94 |
| CON8-3 | 50 | 906.7 | D | 827.9 | 822.9 | 2.07 | 88 | 816.7 | **811.1** | 0.69 | 91 | 813.7 | **811.1** | 0.33 | 98 |
| CON8-4 | 50 | 876.8 | D | 779.1 | **772.3** | 0.89 | 88 | 779.3 | **772.3** | 0.91 | 87 | 774.3 | **772.3** | 0.27 | 95 |
| CON8-5 | 50 | 866.9 | D | 773.9 | 763.1 | 2.41 | 85 | 772.2 | 763.1 | 2.19 | 92 | 766.5 | **755.7** | 1.44 | 94 |
| CON8-6 | 50 | 749.1 | D | 717.0 | 705.8 | 3.46 | 88 | 712.5 | 696.9 | 2.81 | 95 | 707.9 | **693.1** | 2.14 | 96 |
| CON8-7 | 50 | 929.8 | D | 844.8 | 831.5 | 3.69 | 86 | 843.1 | 818.0 | 3.48 | 94 | 833.1 | **814.8** | 2.24 | 94 |
| CON8-8 | 50 | 833.1 | D | 781.2 | 774.1 | 0.93 | 87 | 781.3 | 775.9 | 0.94 | 88 | 778.8 | **774.0** | 0.63 | 94 |
| CON8-9 | 50 | 877.3 | D | 813.3 | 812.0 | 0.49 | 86 | 814.5 | 812.0 | 0.64 | 91 | 813.0 | **809.3** | 0.46 | 92 |
| Tot. | | 33797 | | 30867.6 | 30642.7 | | 5267 | 30823.9 | 30592.8 | | 5308 | 30695.7 | 30530.3 | | 6368 |
| Avg. | | 824 | | | | 1.07 | 128 | | | 0.96 | 129 | | | 0.58 | 155 |
| < PB | | | | | 40 | | | | 40 | | | | 40 | | |
| #B | | 1 | | | 24 | | | | 23 | | | | 36 | | |

Table 29: The first problem in the table is Min's 21 customer problem which was solved to optimality by Halse (H) [19]. The rest of the problems were proposed by Dethloff (D) [10].

| | Best known | | | Std. Removals | | | | 6R - no learning | | | | 6R - normal learning | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | n | cost | reference | avg. sol. | best sol. | avg. gap (%) | avg. time (s) | avg. sol. | best sol. | avg. gap (%) | avg. time (s) | avg. sol. | best sol. | avg. gap (%) | avg. time (s) |
| SN1X | 50 | 501 | D | 475 | **467** | 1.75 | 171 | 472 | **467** | 1.22 | 190 | 473 | **467** | 1.27 | 221 |
| SN2X | 75 | 782 | D | 718 | **702** | 2.40 | 255 | 722 | 709 | 2.84 | 271 | 719 | 704 | 2.46 | 294 |
| SN3X | 100 | 847 | D | 739 | **727** | 1.56 | 768 | 746 | 731 | 2.54 | 695 | 741 | 731 | 1.93 | 863 |
| SN4X | 150 | 1050 | D | 919 | 894 | 4.75 | 1345 | 903 | **877** | 2.95 | 1459 | 893 | 879 | 1.79 | 1676 |
| SN5X | 199 | 1348 | D | 1132 | **1108** | 2.13 | 2057 | 1162 | 1138 | 4.83 | 2217 | 1130 | **1108** | 1.99 | 2340 |
| SN6X | 50 | 584 | D | 559 | **559** | 0.08 | 97 | 559 | **559** | 0.08 | 98 | 559 | **559** | 0.08 | 113 |
| SN7X | 75 | 961 | D | 918 | 905 | 1.82 | 154 | 917 | 903 | 1.72 | 158 | 910 | **901** | 0.95 | 167 |
| SN8X | 100 | 923 | NS | 872 | **866** | 0.69 | 384 | 872 | **866** | 0.74 | 367 | 868 | **866** | 0.29 | 413 |
| SN9X | 150 | 1215 | NS | 1239 | 1221 | 3.54 | 703 | 1235 | **1197** | 3.17 | 726 | 1228 | 1205 | 2.57 | 765 |
| SN10X | 199 | 1571 | D | 1526 | 1494 | 4.42 | 1136 | 1522 | 1490 | 4.13 | 1214 | 1501 | **1462** | 2.71 | 1275 |
| SN11X | 120 | 959 | D | 907 | 875 | 8.33 | 1725 | 921 | 875 | 10.05 | 1410 | 901 | **837** | 7.70 | 1821 |
| SN12X | 100 | 804 | D | 698 | 688 | 2.11 | 628 | 692 | **683** | 1.32 | 574 | 692 | 685 | 1.29 | 684 |
| SN13X | 120 | **1576** | D | 1637 | 1595 | 3.90 | 494 | 1601 | 1591 | 1.57 | 539 | 1593 | 1578 | 1.09 | 563 |
| SN14X | 100 | 871 | D | 904 | 876 | 4.69 | 354 | 896 | **863** | 3.75 | 367 | 897 | 885 | 3.87 | 387 |
| Tot. | | 13992 | | 13242 | 12976 | | 10270 | 13219 | 12947 | | 10286 | 13105 | 12866 | | 11585 |
| Avg. | | 999 | | | | 3.01 | 734 | | | 2.92 | 735 | | | 2.14 | 827 |
| < PB | | | | | 11 | | | | 13 | | | | 12 | | |
| #B | | 1 | | | 6 | | | | 7 | | | | 7 | | |

Table 30: Nagy and Salhi's VRPSDP instances. X-type problems.

| | Best known | | | Std. Removals | | | | 6R - no learning | | | | 6R - normal learning | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | n | cost | reference | avg. sol. | best sol. | avg. gap (%) | avg. time (s) | avg. sol. | best sol. | avg. gap (%) | avg. time (s) | avg. sol. | best sol. | avg. gap (%) | avg. time (s) |
| SN1Y | 50 | 501 | D | 470 | **467** | 0.69 | 194 | 471 | **467** | 0.88 | 192 | 469 | **467** | 0.53 | 235 |
| SN2Y | 75 | 782 | D | 692 | **685** | 1.04 | 315 | 704 | 691 | 2.88 | 268 | 694 | **685** | 1.34 | 331 |
| SN3Y | 100 | 847 | D | 743 | **734** | 1.20 | 632 | 751 | 742 | 2.28 | 625 | 747 | 738 | 1.71 | 708 |
| SN4Y | 150 | 1050 | D | 868 | **854** | 1.57 | 1866 | 876 | 856 | 2.56 | 1487 | 881 | 876 | 3.08 | 1788 |
| SN5Y | 199 | 1348 | D | 1158 | **1131** | 2.37 | 2030 | 1186 | 1132 | 4.86 | 2106 | 1169 | 1146 | 3.31 | 2177 |
| SN6Y | 50 | 584 | D | 560 | **559** | 0.21 | 93 | 562 | **559** | 0.63 | 96 | 560 | **559** | 0.20 | 101 |
| SN7Y | 75 | 961 | D | 987 | 969 | 3.73 | 158 | 1008 | 979 | 5.92 | 163 | 993 | **952** | 4.38 | 166 |
| SN8Y | 100 | 923 | NS | 896 | 880 | 2.63 | 361 | 916 | 894 | 4.85 | 362 | 895 | **873** | 2.43 | 398 |
| SN9Y | 150 | **1215** | NS | 1282 | 1267 | 5.55 | 732 | 1286 | 1256 | 5.83 | 720 | 1288 | 1271 | 6.03 | 757 |
| SN10Y | 199 | **1527** | NS | 1597 | 1567 | 4.57 | 1207 | 1596 | 1573 | 4.54 | 1195 | 1591 | 1552 | 4.16 | 1255 |
| SN11Y | 120 | 1070 | D | 972 | 938 | 5.71 | 1193 | 980 | 956 | 6.54 | 1154 | 951 | **920** | 3.42 | 1376 |
| SN12Y | 100 | 825 | D | 683 | **673** | 1.58 | 531 | 689 | 686 | 2.39 | 506 | 684 | 675 | 1.65 | 539 |
| SN13Y | 120 | **1576** | D | 1771 | 1726 | 12.38 | 531 | 1629 | 1612 | 3.34 | 538 | 1613 | 1602 | 2.33 | 547 |
| Tot. | | 13209 | | 12680 | 12451 | | 9844 | 12654 | 12403 | | 9412 | 12534 | 12315 | | 10378 |
| Avg. | | 944 | | | | 3.33 | 703 | | | 3.65 | 672 | | | 2.66 | 741 |
| < PB | | | | 9 | | | | 9 | | | | 10 | | | |
| #B | | 3 | | 7 | | | | 2 | | | | 6 | | | |

Table 31: Nagy and Salhi's VRPSDP instances. Y-type problems.

# References


[1] R.K. Ahuja, O. Ergun, J.B. Orlin, A.P Punnen *A survey of very large scale neighborhood search techniques*, Discrete Applied Mathematics **123** 75–102 (2002).

[2] E.Angelelli, R. Mansini, *The vehicle routing problem with time windows and simultaneous pick-up and delivery*, in *Quantitative Approaches to Distribution Logistics and Supply Chain Management*, (edited by A. Klose, M. G. Speranza, L. N. Van Wassenhove), Springer-Verlag, 249–267 (2002)

[3] S. Anily, *The vehicle-routing problem with delivery and back-haul options*, Naval Research Logistics **43** 415–434 (1996).

[4] R. Battiti, G. Tecchiolli, *The reactive tabu search*, ORSA Journal of Computing **6** 126–140 (1994).

[5] D.O. Casco, B.L. Golden, E.A. Wasil, *Vehicle routing with backhauls: models, algorithms and case studies*, in *Vehicle Routing: Methods and Studies* (Edited by B. Golden and A. Assad), North-Holland, Amsterdam 127–147 (1988).

[6] J.-F. Cordeau, G. Laporte, A. Mercier, *A unified tabu search heuristic for behicle routing problems with time windows*, Journal of the Operational Research Society, **52** 928–936 (2001).

[7] J. Crispim, J. Brandao, *Reactive tabu search and variable neighbourhood descent applied to the vehicle routing problem with backhauls*, MIC'2001 4th Metaheuristic International Conference, Porto, Portugal July 16–20 (2001).

[8] G. Desaulniers, J. Desrosiers, A. Ercmann, M.M. Solomon, F. Soumis, *VRP with pickup and delivery*, In P. Toth and D. Vigo (eds.): The Vehicle Routing Problem, SIAM Monographs on Discrete Mathematics and Applications **9**, SIAM, Philadelphia, 225-242, (2002).

[9] J. Dethloff, *Relation between vehicle routing problems: an insertion heuristic for the vehicle routing problem with simultaneous delivery and pick-up applied to the vehicle routing problem with backhauls*, Journal of the Operational Research Society **53** 115–118 (2002).

[10] J. Dethloff, *Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up*, OR Spektrum **23** 79-96 (2001).

[11] J.J. Dongarra, *Performance of various computers using standard linear equation software*, University of Tennessee Computer Science Technical Report, CS-89-85, (2004).

[12] C. Duhamel, J.-Y. Potvin, J.-M. Rousseau, *A tabu search heuristic for the vehicle routing problem with backhauls and time windows*, Transportation Science **31** 49–59 (1997).

[13] S. Gelinas, M. Desrochers, J. Desrosiers, M.M. Solomon., *A new branching strategy for time constrained routing problems with application to backhauling*, Annals of Operations Research **61** 91–109 (1995).

[14] M. Gendreau, G. Laporte, D. Vigo, *Heuristics for the traveling salesman problem with pickup and delivery*, Computers & Operations Research **26**, 699–714 (1999).

[15] H. Ghaziri, I.H. Osman, *A neural network algorithm for the traveling salesman problem with backhauls*, Computers & Industrial Engineering **44**, 267–281 (2003).

[16] M. Goetschalckx, C. Jacobs-Blecha, *The vehicle routing problem with backhauls*, European Journal of Operational Research **42** 39–51 (1989).

[17] Ø. Halskau; I. Gribkovskaia; K.N.B. Myklebost. *Models for pick-up and deliveries from depots with lasso solutions*. Proceedings of the 13th Annual Conference on Logistics Research - NOFOMA 2001, Collaboration in logistics : Connecting Islands using Information Technology. Reykjavik, Iceland, 2001-06-14 - 2001-06-15. Chalmers University of Technology, Göteborg, Sweden. 279–293 (2001).

[18] P. Hansen, N. Mladenovic, *An introduction to variable neighborhood search*, in: S. Voss et. al. (ed) Meta-Heuristics, Advances and Trends in Local Search Paradigms for Optimization, Kluwer, Boston, 433–458 (1999).

[19] K. Halse, *Modeling and solving complex vehicle routing problems*, PhD thesis, Institute of Mathematical Statistics and Operations Research (IMSOR), Technical University of Denmark (1992).

[20] T. Hasama, H. Kokubugata, H. Kawashima, *A heuristic approach based on the string model to solve vehicle routing problem with backhauls*, Proceedings of the 5th World Congress on Intelligent Transport Systems (ITS), Seoul, 1998.

[21] A. Hertz, E.D. Taillard, D. de Werra *A tutorial on tabu search*, in: E.H.L. Aarts and J. K. Lenstra, Local search in combinatorial optimization, Wiley, 121–136 (1997)

[22] C. Jacobs-Blecha, M. Goetschalckx, *The vehicle routing problem with backhauls: properties and solution algorithms*, Technical Report, 1992-1998, Georgia Tech Research Corporation.

[23] G. Kontoravdis, J.F. Bard, *A GRASP for the vehicle routing problem with time windows*, ORSA Journal on Computing **7** 10–23 (1995).

[24] J.B. Kruskal, *On the shortest spanning subtree of a graph and the traveling salesman problem*, Proceedings of the American Mathematical Society **7** 48–50 (1956).

[25] H. Min, *The multiple vehicle routing problem with simultaneous delivery and pickup*, Transportation Research Part A **23** 377–386 (1989).

[26] A. Mingozzi, S. Giorgi, R. Baldacci, *An exact method for the vehicle routing problem with backhauls,* Transportation Science **33**, 315–329 (1999).

[27] G. Nagy, S. Salhi, *Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries,* Working Paper no. 42, Canterbury Business School, 2003.

[28] I.H. Osman, N.A. Wassan, *A reactive tabu search meta-heuristic for the vehicle routing problem with backhauls,* Journal of Scheduling **5** 263–285 (2002).

[29] J.Y. Potvin, J.-M. Rousseau, *A parallel route building algorithm for the vehicle routing and scheduling problem with time windows,* European Journal of Operational Research **66** 331–340 (1993).

[30] M. Reimann, Doerner K., Hartl R.F., *Insertion based ants for vehicle routing problems with backhauls and time windows,* LNCS 2463 135–148 (2002).

[31] S. Ropke, *A local Search heuristic for the pickup and delivery problem with time windows,* Working paper, DIKU, University of Copenhagen (2003).

[32] S. Salhi, G. Nagy, *A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling,* Journal of the Operational Research Society (1999) 50, 1034-1042.

[33] G. Schrimpf, J. Schneider, H. Stamm-Wilbrandt, G. Dueck, *Record breaking optimization results – using the ruin & recreate principle*, J. Comp. Phys. **159**, 139-171 (2000).

[34] M. Sigurd, D. Pisinger, M. Sig, *The pickup and delivery problem with time windows and precendences*, Transportation Science **38** 197–209 (2004).

[35] P. Shaw, *Using constraint programming and local search methods to solve vehicle routing problems*, Proceedings CP-98 (Fourth International Conference on Principles and Practice of Constraint Programming) (1998).

[36] H. Süral, J.H. Bookbinder, *The single-vehicle routing problem with unrestricted backhauls*, Networks **41**, 127–136 (2003).

[37] M.M. Solomon, *Algorithms for the vehicle routing and scheduling problems with time window constraints* Operations Research, **35** 254–265, (1987).

[38] S.R. Thangiah, J.-Y. Potvin, Sun T., *Heuristic approaches to vehicle routing with backhauls and time windows,* Computers & Operations Research **23** 1043–1057 (1996).

[39] F.A. Tillman, T. M. Cain *An upper bound algorithm for the single and multiple terminal delivery problem* Management Science **18** 664-682 (1972).

[40] P. Toth, D. Vigo, *A heuristic algorithm for the symmetric and asymmetric vehicle routing problems with backhauls,* European Journal of Operational Research **113** 528–543 (1999).

[41] P. Toth, D. Vigo, *An exact algorithm for the vehicle routing problem with backhauls*, Transportation Science **31** 372-285 (1997).

[42] P. Toth, D. Vigo, *VRP with backhauls*, In P. Toth and D. Vigo (eds.): The Vehicle Routing Problem, SIAM Monographs on Discrete Mathematics and Applications **9**, SIAM, Philadelphia, 195-221, (2002).

[43] A. Wade, S. Salhi, *An ant system algorithm for the mixed vehicle routing problem with backhauls*, in M.G.C. Resende and J.P. de Sousa (eds.): Metaheuristics: Computer Decision-Making, Chapter 33, 699-719, Kluwer (2003).

[44] A. Wade, S. Salhi, *An ant system algorithm for the vehicle routing problem with backhauls*, MIC'2001 - 4th Metaheursistic International Conference.

[45] A.C. Wade, S. Salhi, *An investigation into a new class of vehicle routing problem with backhauls*, Omega **30** 497–487 (2002).

[46] Web page: www.diku.dk/~sropke

[47] Y. Zhong, M.H. Cole, *A simple approach to linehaul-backhaul problems: a guided local search approach for the vehicle routing problem*, Technical Report, Department of Industrial Engineering, University of Arkansas, USA, 2001.

[48] Y. Zhong, M.H. Cole, *A vehicle routing problem with backhauls and time windows: a guided local search solution*, Transportation Research Part E, Article in press (2004).