Department of Econometrics
Erasmus School of Economics
Erasmus University Rotterdam

Reproduction of the iterative three-component heuristic for the team orienteering problem with time windows

Brian Gelderblom
381197
Supervisor: T.R. Visser MSc
Co-reader: dr. R. Spliet

Bachelor Thesis
Econometrics & Operations Research
4 July 2016

Abstract

This thesis reproduces the Iterative 3-Component Heuristic (I3CH) of Hu and Lim (2014). The I3CH tries to solve the Team Orienteering Problem with Time Windows (TOPTW). The aim of the TOPTW is to maximize the total profit score obtained by visiting a set of locations, while each location can only be visited in its time window, with a limited number of vehicles.

The I3CH consists three components to solve the TOPTW. A local search and simulated annealing procedure are used to search routes in the solution space. The route recombination procedure uses mathematical programming to select a combination of routes found in the local search and simulated annealing procedure such that the profit score is maximized.

Our reproduction of the I3CH performs 0.11% better than the I3CH of Hu and Lim (2014), but our heuristic is slower than theirs. The profit scores of 87 out of 304 instances are improved. We found 16 solutions with higher profit scores than the best known solutions available in 2012 and 10 solutions with higher profit scores than the latest best known solutions (Gunawan, Lau, and Lu, 2015).

Moreover, this thesis extends the TOPTW by taking into account that certain locations cannot be visited by a route. The heuristic is changed in multiple ways, which leads to lower computation times.

Contents

1 Introduction

This thesis studies the Team Orienteering Problem with Time Windows (TOPTW). In this problem, each location has a certain profit score, a service time and a time window in which the location should be visited. The objective of the TOPTW is to maximize the total profit scores obtained by visiting the locations. The number of routes is limited and each location can be visited at most once. Each route starts and ends at a depot and locations should be served in their time window.

The TOPTW arises in many practical logistic problems, for instance in some companies that often repair services to their customers. Technicians typically visit many customers per day and try to repair those products. Often, technicians are not able to visit all waiting customers per day, because they can only work for a fixed amount of time per day. Customers can only be visited in a specific time interval.

Hu and Lim (2014) pose an Iterative three-Component Heuristic (I3CH) to solve TOPTW. This heuristic uses local search, simulated annealing and route recombination iteratively. A local search and simulated annealing procedure is used to search routes in the solution space. The route recombination procedure uses mathematical programming to select a combination of routes found in the local search and simulated annealing procedure such that the profit score is maximized.

The aim of the thesis is to first implement the I3CH described in the article "An iterative three-component heuristic for the team orienteering problem with time windows" by Hu and Lim published in 2014. Second, an extension of the TOPTW and its effect on the performance of the I3CH will be investigated. In this extension, we take into account that technicians are specialized and are not qualified to serve all customers.

A description of the problem definition is given in chapter 2. In chapter 3, a literature review is given. The methodology is described in chapter 4 and a description of the data in chapter 5. In chapter 6, the results of the I3CH are shown. The implementation and the results of an extension are given in chapter 7. The conclusion can be found in chapter 8.

2 Problem definition

In this chapter, a description of the TOPTW is given. Second, an integer programming problem formulated as a Mixed Integer Program (MIP) of the TOPTW is described. Third, an introduction of an extension for the TOPTW, TOPTWST, is given, where locations can only be visited by a subset of the routes.

2.1 TOPTW

TOPTW is defined as follows. Let a complete graph $G = (V, A)$ be given. The set of vertices $V = \{0, 1, 2, \ldots, n\}$ represents $n + 1$ different locations and the set $A = \{(i, j) : i \neq j \in V\}$ represents the connecting arcs between the locations. Let the travel time t_{ij} between location i and j be equal to the Euclidian distance d_{ij} from i to j. Let location 0 represents the depot. Every route must start and end in the depot. At most m routes may be used.

Every route must begin and end at the depot within its time window $[O_0, C_0]$. Each customer $i \in \{1, \ldots, n\}$ has a profit score p_i, a service time T_i and a time window $[O_i, C_i]$. The technician is only able to start service in the time window of the customer. In case of an early arrival at customer i, the technician has to wait until time O_i. Every customer can only be visited at most once.

In a solution, not all customers may be visited due to time restrictions and the number of vehicles that are available. The aim of the TOPTW is to create at most m feasible routes such that the total profit scores are maximized. An example of a solution with $m = 2$ routes is shown in figure 1.

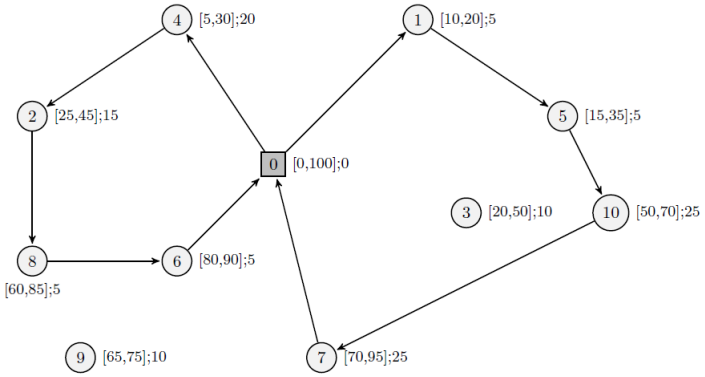


Figure 1: A possible solution

The time window of each location can be seen in the corresponding brackets. The time window of location 0, the depot, is $[0, 100]$. This means that the earliest time a route can start is 0 and the latest time a route can end is 100. All other locations can only be visited within their corresponding time window. The third number represents the profit score of the location.

According to figure 1, the first route starts in the depot and the locations 1, 5, 10 and 7 are visited, with a total profit score of 60. The second route visits the locations 4, 2, 8 & 6 and has a total profit score of 45. Both routes start and end in the depot. Location 3 and 9 are not visited.

2.2 Integer programming problem

Vansteenwegen, Souffriau, Vanden Berghe, and Van Oudheusden (2009) describe an integer programming problem formulated as an Integer Program (IP) of the TOPTW. We create the following decision

variables:

$$x_{ijr} = \begin{cases} 1 & \text{if location } i \text{ is visited before location } j \text{ on route } r, \\ 0 & \text{otherwise,} \end{cases}$$

$$y_{ir} = \begin{cases} 1 & \text{if location } i \text{ is visited on route } r, \\ 0 & \text{otherwise.} \end{cases}$$

Other variables are s_{ir}, which is equal to the start of the service at location i on route r, and M, which is a large constant. The IP can be formulated as follows:

$$\max \quad \sum_{r=1}^{m}\sum_{i=1}^{n} p_i y_{ir} \tag{1}$$

$$\text{s.t.} \quad \sum_{r=1}^{m}\sum_{j=1}^{n} x_{0jr} = \sum_{r=1}^{m}\sum_{i=1}^{n} x_{i0r} = m, \tag{2}$$

$$\sum_{i=1}^{n} x_{ikr} = \sum_{i=1}^{n} x_{kjr} = y_{kr} \qquad \forall k = 1,\ldots,n, \quad r = 1,\ldots m, \tag{3}$$

$$s_{ir} + T_i + d_{ij} - s_{jr} \leq M(1 - x_{ijr}) \qquad \forall i,j = 0,\ldots,n, \quad r = 1,\ldots,m, \tag{4}$$

$$\sum_{r=1}^{m} y_{kr} \leq 1 \qquad \forall k = 1,\ldots,n, \tag{5}$$

$$\sum_{i=0}^{n} \left(T_i y_{ir} + \sum_{j=1}^{n} d_{ij} x_{ijr} \right) \leq C_0 \qquad \forall r = 1,\ldots,m, \tag{6}$$

$$O_i \leq s_{ir} \qquad \forall i = 0,\ldots,n, \quad r = 1,\ldots,m, \tag{7}$$

$$s_{ir} \leq C_i \qquad \forall i = 0,\ldots,n, \quad r = 1,\ldots,m, \tag{8}$$

$$x_{ijr}, y_{ir} \in \{0,1\} \qquad \forall i,j = 0,\ldots,n, \quad r = 1,\ldots,m. \tag{9}$$

Equation (1) is the objective function, which maximizes the total profit. Constraints (2) make sure that every route start and end in the depot. According to constraints (3), all the locations on route r are connected. Constraints (4) determine the timeline of each route. Constraints (5) take into account that every location can only be visited at most once. Constraints (6) make sure that the route ends within the time window of the depot and constraints (7) and (8) make sure that a location is visited within its time window.

2.3 TOPTWST

The Team Orienteering Problem with Time Windows for Specialized Technicians (TOPTWST) is an extension of TOPTW. In this case, technicians are specialized and are not qualified to visit all locations. Locations are divided into l groups P_l, where $P_l \subset V\backslash\{0\}$. All locations must belong to a group and a location cannot belong to multiple groups, which means that $\bigcup P_l = V\backslash\{0\}$ and $\bigcap P_l = \emptyset$. V_l is equal to the number of routes which belong to group l.

The methodology and the results of TOPTWST are shown in chapter 7.

3 Literature review

The TOPTW belongs to the group of Orienteering Problems (OPs), which has been extensively studied in literature. Vansteenwegen, Souffriau, and Van Oudheusden (2011) discuss some examples of OPs.

One can say that TOPTW is an extension the classic Orienteering Problem, OP. According to Golden, Levy, and Vohra (1987), the OP falls in the category of NP-hard problems. TOPTW is an extension of OP, which means that this problem belongs to the same category of NP-hard problems.

An application of the OP is described in Tsiligirides (1984), where a traveling salesperson does not have enough time to visit all possible cities. Given the revenue of each city, the salesperson visits a combination of cities such that the total revenue is maximized while the travel time is at most a day (or week). Another application is described in Souffriau, Vansteenwegen, Vertommen, Berghe, and Van Oudheusden (2008). Tourists are not always able to visit everything they want to see. Some attractions are also more valuable than other attractions. By solving an OP, a personal tourist plan can be made in order to visit the most valuable attractions given a certain time period.

The Team Orienteering Problem (TOP) is an extension of the OP. Instead of one tour, the TOP allows multiple routes. An application of the TOP is described in Butt and Cavalier (1994). An athlete recruiter has to visit multiple high schools. The recruiter has ranked each school based on its recruiting potential. Given a limited time per day, the recruiter chooses the schools to visit each day such that the recruiting potential is maximized. Another application is described in the article of Tang and Miller-Hooks (2005). A company has multiple technicians and each technician can only work for a finite number of hours per day. It is often not possible to visit all customers per day. The aim is to choose some customers that the technicians should visit that day, while taking customer importance and task urgency into account.

The OP and TOP can be extended to OPTW and TOPTW by adding time windows. Each location can only be visited in a specific time window. The description of the OP application in Souffriau et al. (2008) can be rephrased to an OPTW when some attractions have opening and closing times. The TOP application described in Tang and Miller-Hooks (2005) can rephrase to a TOPTW when a technician can only visit a customer during some hours of the day.

TOPTW has recently gained attention in the literature. Different methods are posed to create solutions for the TOPTW. Montemanni and Gambardella (2009) posed an ant colony system (ACO) to solve the TOPTW. ACO tries to find an optimal path in graph by using multiple workers (ants) and pheromones for communication. Montemanni, Weyland, and Gambardella (2011) and Gambardella, Montemanni, and Weyland (2012) improved their ACO which led to better results in less amount of time.

Vansteenwegen et al. (2009) pose an iterated local search (ILS) algorithm to solve the TOPTW. ILS iteratively generates a sequence of solutions by a fixed heuristic (Lourenço, Martin, and Stützle, 2001). The running times of the ILS are significantly lower than of the ACO, while the solution quality remains similar.

In 2012, an article of Lin and Yu (2012) was published in which two versions of a simulated annealing algorithm for the TOPTW were developed. Simulated annealing is a heuristic based on local search where it is possible to escape from a local optimum by accepting solutions with a small probability. The fast algorithm (FSA) takes several seconds to compute a solution, while the slow algorithm (SSA) takes several minutes to compute better solutions than the solutions of FSA.

Hu and Lim (2014) compared their running times and solution quality of I3CH with the running times and solution quality of ACO (Montemanni and Gambardella, 2009), ILS (Vansteenwegen et al., 2009) and SSA (Lin and Yu, 2012). It follows that both the running times and solution quality of I3CH are better than ACO. The running times of ILS are lower I3CH, but the solution quality of I3CH is better than ILS. This also holds for SSA.

As mentioned in chapter 2.3, technicians are skilled and are not able to visit all customers. Chen,

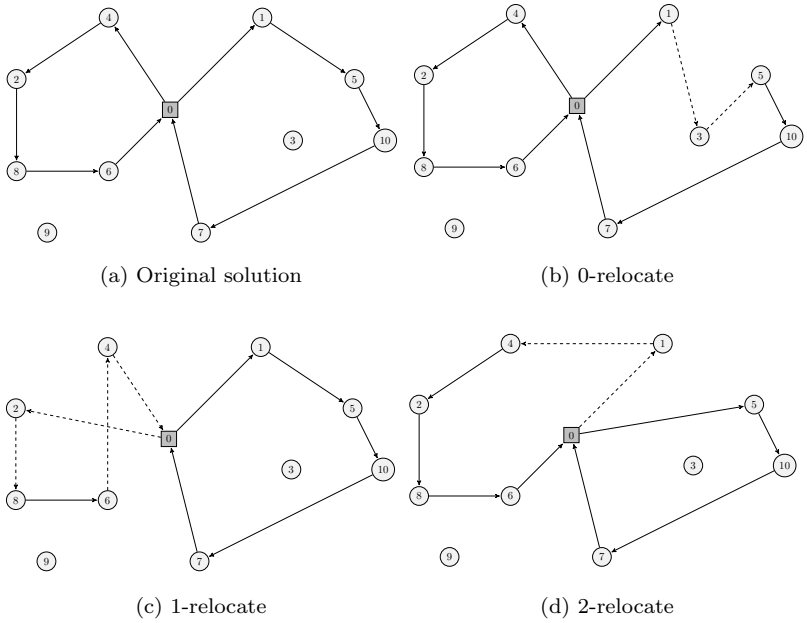
Thomas, and Hewitt (2015) takes this phenomenom into account and models a model of technician routing. The skills and the experience of the technicians are taken into account. Their research is based on the technician routing and scheduling problem (TRSP) and a Markov decision process model.

4 Methodology

The aim of the thesis is to implement the three-component heuristic described in Hu and Lim (2014) and to reproduce their results. This method consists of three components: local search (LS), simulated annealing (SA) and route recombination (RR). Both the local search and simulated annealing will be used as neighborhood search approaches to search in the solution space. The routes obtained during the local search and simulated annealing will be stored into a route pool which is iteratively used by the route recombination.
First, the *eliminator* and *post-processing* operators and a feasibility check for the routes are explained. Second, the local search and simulated annealing components are explained. Third, the route recombination procedure is described. Last, the algorithm of the I3CH is shown.

4.1 *Eliminator* and *post-processing* procedure

Hu and Lim (2014) describe a neighborhood operator called *eliminator* to, given an initial feasible solution, construct neighborhood solutions. The *eliminator* operator can be divided into two stages. In the first stage, given a feasible solution A, locations are randomly removed from some routes and the operator tries to add unvisited locations to the existing routes. A solution B is hereby created, where B is a neighbor of A. The probability that locations with relatively high profit score will be removed from a route is set lower than locations with a relatively lower profit score. The aim is to maximize the total profit scores. It is important to keep as many locations with a relatively high profit score in the existing routes. This procedure will stop when no more locations can be inserted into the routes. Second, a *post-processing* procedure will be used to improve solution B. This procedure consists seven operators. These operators are divided into three types: *relocate*, *exchange* and *2-opt*.



(a) Original solution (b) 0-relocate

(c) 1-relocate (d) 2-relocate

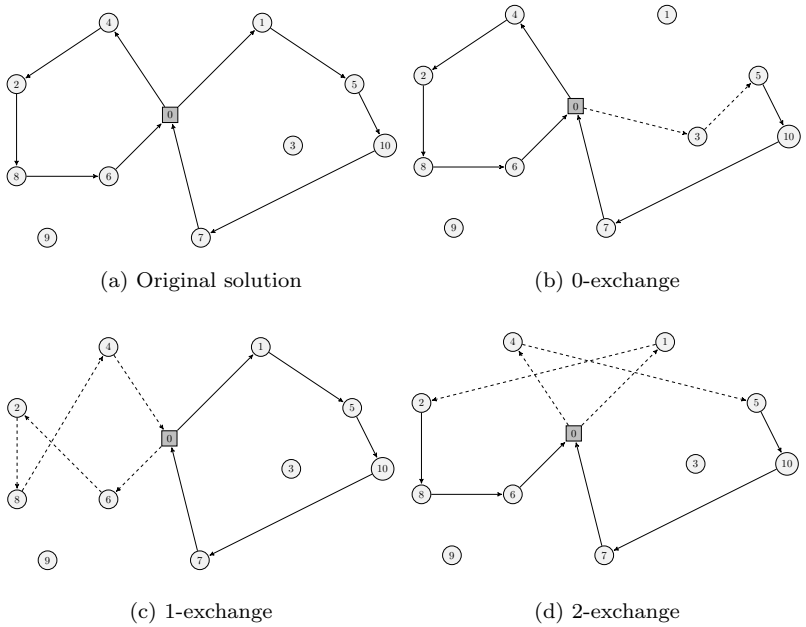
Figure 2: Possible solutions after relocate operators

The *relocate* operators add a location c_j at a feasible position on route r_i. A graphical representation is shown in figure 2. The solution before applying the *relocate* operators is shown in figure 2a.

Figure 2b shows the 0-relocate operator, which inserts an unvisited location c_j into a route. Location 3 used to be unvisited, but is now inserted into a route between location 1 and 5.

Location c_j can be inserted at another position in the same route. This procedure can be done by the 1-relocate operator and can be seen in figure 2c. Location 4 used to be visited at the beginning of the route. After the 1-relocate operator, location 4 will be visited at the end of the route.

The 2-relocate operator inserts a location c_j in another route. This is shown in figure 2d. Location 1 will now be visited by the other route.



(a) Original solution (b) 0-exchange

(c) 1-exchange (d) 2-exchange

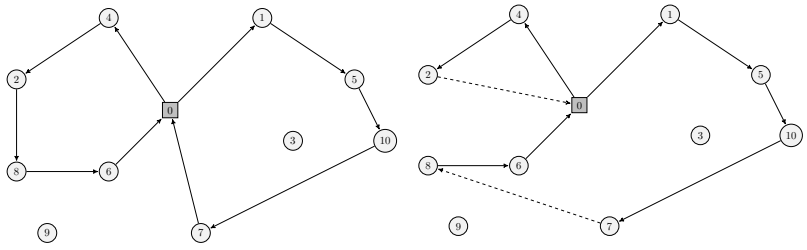
Figure 3: Possible solutions after exchange operators

The *exchange* operators switch two locations c_j and c_k, where $j \neq k$. A graphical representation is shown in figure 3. The solution before applying the *exchange* operator is shown in figure 3a.

Figure 3b shows the 0-exchange operator, which inserts an unvisited location c_j into a route and removes a location c_k from the same route. Location c_k will not be visited in that route. The total profit score increases if the profit of location j is higher than the profit of location k. For example, location 3 is added to a route and location 1 is removed from that route.

Location c_j and c_k can switch positions in a route. This procedure is done by the 1-exchange operator and can be seen in figure 3c. Location 6 is visited first instead of last and location 4 is visited last instead of first.

The 2-exchange operator switches location c_j from route r_i with location c_k from route r_l. This is shown in figure 3d. Location 1 and 4 are now visited by the other route.



(a) Original solution (b) Possible solution after 2-opt operator

Figure 4: Possible solution after 2-opt operators

The seventh operator is the *2-opt* operator. The solution before applying the *2-opt* operator is shown in figure 4a. This operator switches two edges from two different routes. It might be possible to create two new feasible routes. Figure 4b shows the possible new routes generated by the *2-opt* operator.

The *post-processing* procedure will first apply the 2-relocate, 2-opt and 2-exchange operators and then moves to the best found neighbor. After that, the 1-relocate and 1-exchange operator will be used and it moves to the best found neighbor. Last, the 0-relocate and 0-exchange operators are applied. The procedure is repeated when the solution quality cannot be further improved. This means that the profit score cannot increase and the total traveled distance cannot be lower.

Each time a route is changed, we have to check whether this route is still feasible. A method described in Campbell and Savelsbergh (2004) is used to quickly ($O(1)$ time) check whether a move is feasible.

For each location j, its time window $[O_j, C_j]$ is known. O_j is the earliest time the location can be visited and C_j is the latest time the location can be visited. For each location j already assigned to a route, two variables are maintained. These are o_j, the earliest time a location can be visited in a route, and c_j, the latest time a location can be visited in a route. The earliest time a route can start is at 0 and the latest time a route can end is at C_0. When a location is not inserted into a route, o_j is set to O_j and c_j is set to C_j.

Every time a location j is inserted between location $k-1$ and k, the earliest time location j can be visited is

$$o_j = \max\left(O_j, o_{k-1} + T_{k-1} + d_{k-1,j}\right)$$

and the latest time location j can be visited is

$$c_j = \min\left(O_j, o_k - T_j - d_{k,j}\right).$$

An insertion of location j in a route is feasible under TW restrictions when $o_j \leq c_j$.

The insertion of location j in a route can influence the earliest arrival time of locations that still have to be visited and the latest arrival time of locations that have already been visited. We only do this procedure after a move is executed. For all the locations v that still have to be visited, the earliest arrival time will be updated as follows:

$$e_v = \max\left(e_v, e_{v-1} + T_{v-1} + d_{v-1,v}\right).$$

For all the locations w that have already been visited, the latest arrival time will be updated as follows:

$$l_w = \min\left(l_w, l_{w+1} - T_w - d_{w,w+1}\right).$$

By applying this method, it is easy to check whether a location can be inserted at a specified position in a route.

4.2 Local search and simulated annealing

Both the local search and simulated annealing will use the *eliminator* and *post-processing* procedure. The aim of these two components is to improve the current best obtained solution.

The local search procedure explores the neighborhood of a starting solution X. In every iteration, the local search procedure generates N neighbors of X by using the *eliminator* operator and *post-processing* procedure. In the next iteration, the local search procedure starts with the best found solution so far. The procedure terminates after $I_{ls\_no\_impr}$ consecutive iterations without improvement.

The simulated annealing procedure explores the neighborhood of a starting solution Y. In every iteration, the simulated annealing procedure generates only one neighbor Y' of Y by using the *eliminator* operator and *post-processing* procedure. N neighbors are generated in each iteration. If Y' is better than Y, Y' is the best found solution in this iteration. If Y' is not better than Y, Y' is chosen as best found solution in this iteration with a probability

$$P_{sa} := \exp\left\{\frac{1}{T}\frac{Y' - Y}{Y_{sa}}\right\},$$

where T_0 is the initial temperature, α is the cooling speed, $T = \alpha \cdot T$ is the updated temperature after each step and Y_{sa} is the best found solution so far in the simulated annealing procedure. After each iteration, Y will be updated by the best found solution so far. This procedure is terminated after $I_{sa\_no\_impr}$

consecutive iterations without improvement.

Each neighbor that has been found in both the local search and simulated annealing procedure is stored into a pool. This pool consists all previously determined feasible routes. This pool will be used for the route recombination.

4.3 Route recombination

The third component is the route recombination. A set packing formulation is used to determine the best combination of routes stored in the pool which maximizes profit. A set of routes are stored in $pool = \{r_1, r_2, \ldots, r_{S_{pool}}\}$. S_{pool} is the size of pool. The following variables are created:

$$a_{jk} = \begin{cases} 1 & \text{if location } j \text{ is visited on route } k, \\ 0 & \text{otherwise,} \end{cases}$$

$$x_k = \begin{cases} 1 & \text{if route } r_k \text{ is selected,} \\ 0 & \text{otherwise.} \end{cases}$$

The total profit score of route r_k is

$$q_k = \sum_{j \in V \setminus \{0\}} p_j a_{jk}.$$

The set packing formulation is as follows:

$$\max \quad \sum_{k=1}^{S_{pool}} q_k x_k \tag{10}$$

$$\text{s.t.} \quad \sum_{k=1}^{S_{pool}} a_{jk} x_k \leq 1 \quad \forall j \in V \setminus \{0\}, \tag{11}$$

$$\sum_{k=1}^{S_{pool}} x_k \leq m, \tag{12}$$

$$x_k \in \{0,1\} \qquad \forall k \in \{1, 2, \ldots, S_{pool}\}. \tag{13}$$

Equation (10) maximizes the total profit score. Constraints (11) make sure that each location is visited at most once. According to equation (12), not more than m routes are used. First, the LP relaxation is calculated. If the objective value of the LP relaxation is better than the best found profit so far, the set packing formulation will be solved as a MIP using a commercial MIP solver.

4.4 Iterative 3-component heuristic

Combining the three above mentioned components results into the iterative 3-component heuristic. The heuristic is shown in Algorithm 1.
After the initialization, which will be explained in chapter 4.4.1, the RR, LS and SA are applied. The best solution of these three components or the current best solution is chosen and will be the current best solution in the next iteration of the algorithm. This procedure ends after I_{max} iterations or whether all locations are served.

4.4.1 Initialization

The aim of the initialization is to obtain a starting solution A and to cache some feasible routes into the route pool. First, m empty routes and a list of randomly ordered unvisited locations are created. Second, the *eliminator* operator will be applied. During the initialization, the *eliminator* operator is not able to remove locations from routes, but can only add locations to routes. Third, the *post-processing* procedure tries to improve the solution. This whole procedure is repeated $3N$ times and all the $3N$ solutions are cached into the route pool.

Algorithm 1 Iterative 3-Component Heuristic for the TOPTW

Require: maximum number of iterations I_{max};
Require: integer N;
Require: route pool POOL;
1: Apply the *Initialization* procedure and obtain a starting solution A and put the routes into *pool*;
2: Set i = 1;
3: **while** $i \leq I_{max}$ **do**
4: Apply RR over *pool* and obtain Z_{RR};
5: Apply LS, obtain the best solution X_{LS} and save the routes into *pool*;
6: Apply SA, obtain the best solution Y_{LS} and cache the routes into *pool*;
7: Select the best solution B from the collection $\{A, Z_{RR}, X_{LS}, Y_{SA}\}$ breaking ties arbitrarily;
8: $A = B$, $i = i + 1$;
9: If no location is unvisited in A, then set $i = I_{max} + 1$;
10: **end while**
11: **return** the solution of A;

4.4.2 Pool size

During the initialization, LS and SA, routes are cached into the route pool. This means that the number of routes in the pool increases in every iteration. However, a large route pool can effect the computation times of the RR. The computation times will be longer. The strategy that is used to make sure that the size of the route pool is manageable is the least recently strategy. Each time a route is added to the route pool, it gets a value $apr = 0$. Every time a route is used in the final solution of the RR, the value of *apr* will increase to or by 100. If the route is not used in the final solution of the RR, the value of *apr* decreases by 1. Each time the number of routes exceeds its capacity, the route(s) with the lowest value of *apr* will be removed from the route pool. By applying this strategy, the most profitable routes are not removed from the route pool.

5 Data

Different instance benchmark sets are used to test the I3CH. These instances can be found at http://www.mech.kuleuven.be/en/cib/op and are divided into four categories: "Solomon 100", "Solomon 200", "Cordeau 1-10" and "Cordeau 11-20".

The "Solomon 100" and "Solomon 200" instances are based on the 56 test instances in Solomon (1987) for the vehicle routing problem with time windows and are adapted by Montemanni and Gambardella (2009).

The "Cordeau 1-10" and "Cordeau 11-20" instances are based on the 20 test instances in Cordeau, Gendreau, and Laporte (1997) for the multi-depot vehicle routing problems and are adapted by Montemanni and Gambardella (2009).

Each instance is solved 4 times, for 1, 2, 3 and 4 routes. This means that there 304 problems to be solved.

5.1 Data description

Each instance has a depot, where every route must start and end. The number of locations vary between 48 and 288 for the Cordeau instances and are equal to 100 for the Solomon instances. For each location, the x- and y-coordinates of the location, the service time when the location is visited, the profit score of the visited location and time window of the location are given.

The Solomon instances are divided into three categories. The c-data sets contain locations which are clustered. The r-data sets contain locations which are not clustered, but are randomly generated. The rc-data sets contain randomly generated and clustered locations.

5.2 Best known results

Hu and Lim (2014) compare their results with the best known results until 2012. The best known results are based on the ant colony system by Montemanni and Gambardella (2009), the iterated local search algorithm proposed by Vansteenwegen et al. (2009), the variable neighbor search approach developed by Tricoire, Romauch, Doerner, and Hartl (2010), a hybrid metaheuristic combining the greedy randomized adaptive search procedure with evolutionary local search proposed by Labadie, Melechovský, and Wolfler Calvo (2011), a slow version of the simulated annealing heuristic by Lin and Yu (2012) and the LP-based granular variable neighborhood search algorithm proposed by Labadie, Mansini, Melechovsk, and Calvo (2012).

It follows that 31, 26, 44 and 42 of the instances for $m = 1$, $m = 2$, $m = 3$ and $m = 4$ respectively are the same by applying the I3CH from Hu and Lim (2014). 35 new best solutions are found. 4 of them belong the instances calculated with $m = 1$, 6 with $m = 2$, 7 with $m = 3$ and 18 with $m = 4$.

41.45% of the results were worse than the best found solutions.

6 Results

In this chapter, the results of the reproduced I3CH are given. In this thesis, the results of the reproduced I3CH are denoted by GELD. First, the number of iterations I_{max} is determined. Second, the results of GELD are given and these results are compared with the results from Hu and Lim (2014) and with the Best Known Solutions (BKS) available in 2012. Hu and Lim (2014) use these best known solutions in their paper. Third, we investigate which effect the route recombination has on the final solution. Fourth, we check whether a different amount of created routes in the initialization procedure has an impact on the final results.

The I3CH has been implemented in Java on a MacBook Air (1.3 GHz and 8 GB RAM). ILOG CPLEX 12.6.3 is used as IP solver. Hu and Lim (2014) used a different machine for their calculations. This means that the computation times deviate from each other.

Different variables are mentioned in chapter 4. The values of these variables will be the same as described in Hu and Lim (2014). This means that $N = 50$, $I_{ls\_no\_impr} = I_{sa\_no\_impr} = 20$ and the maximum pool size is equal to 1000. The value of I_{max} will be determined in chapter 6.1.

6.1 Number of iterations

Hu and Lim (2014) tried different values for I_{max}, the maximum number of iterations in the I3CH. They determined the average gap between the I3CH and the BKS available in 2012 and the average computation time for $I_{max} = 500, 1000, 2000, 3000, 4000, 5000$. Hu and Lim (2014) chose $I_{max} = 3000$, which is based on a tradeoff between effectiveness and efficiency.

Due to the high computation times of GELD, it is not possible to use the same I_{max}. For multiple instances, the gap between the average profit and the BKS available in 2012 per 5 iterations are calculated. These results are shown in figure 5.

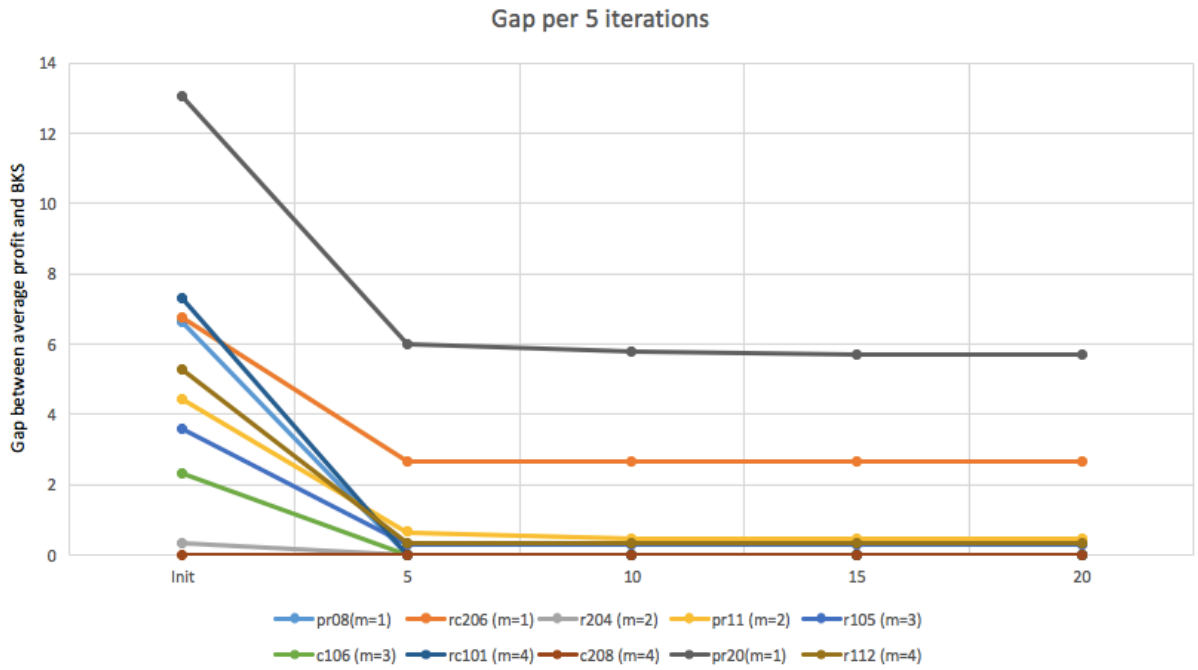


Figure 5: Gap between average profit and BKS available in 2012 per 5 iterations and the initialization

The profit scores of each of the 10 instances are calculated 3 times with 3 different seeds $seed = \{1, 3, 5\}$. The gap is equal to the relative change between the average profit and the BKS available in

2012. For multiple instances, the gap only decreases in the first 10 iterations. For 1 out of 3 runs of the pr20 ($m = 1$) instance, the profit score increases in the fifteenth iteration.

The average gap is equal to 0.99% when $I_{max} = 5$. The number of instances for which the gap does not decrease after 5 iterations is equal to 8.

For $I_{max} = 10$, the average gap is equal to 0.95%. The gap between $I_{max} = 5$ and $I_{max} = 10$ decreases, but the computation time increases with 86.38% by implementing $I_{max} = 10$ instead of 5.

The average gap for $I_{max} = 15$ and $I_{max} = 20$ are equal to 0.94%. The gap is slightly lower than the average gap when $I_{max} = 10$, but the computation time increases with 43.70% when $I_{max} = 15$ is used instead of $I_{max} = 10$.

Taking the average gap and average computation times into account, $I_{max} = 10$ is chosen as the number of iterations in the heuristic. This value of I_{max} is much lower than the value of I_{max} chosen in Hu and Lim (2014). According to our sample, the average gap of 96.67% of the instances does not decrease after 10 iterations. Besides, the average profit score of 8 out of 10 instances from the sample is equal to or higher than the results from Hu and Lim (2014) and the average profit score of 5 out of 10 sample instances is equal to the best known solutions available in 2012.

6.2 Implementation of I3CH

The results of GELD are not the same as the results of Hu and Lim (2014). The performance of GELD compared to I3CH of Hu and Lim (2014) is shown in table 1.

Table 1: Performance of GELD compared to I3CH

Instances	# Better	*# BKS*	*# Better than BKS*	# Same	# Worse	Total
Solomon 100	27	*17*	*3*	71	18	116
Solomon 200	32	*9*	*10*	67	9	108
Cordeau 1-10	12	*4*	*0*	7	21	40
Cordeau 11-20	16	*0*	*3*	3	21	40
Total	87	*30*	*16*	148	69	304

We see that 62.5% of the Solomon instances have the same profit score and that 12.05% of the Solomon instances perform worse than I3CH. The number of Solomon instances that performs better than I3CH is equal to 59. The profit scores of 26 out of 59 Solomon instances that perform better than I3CH are equal to the BKS available in 2012 and the profit scores of 13 out of 59 Solomon instances are higher than the profit scores of the BKS available in 2012. When we compare the profit scores of the 13 instances with the profit scores of the latest BKS (Gunawan, Lau, and Lu, 2015), we conclude that 7 instances perform better and have a higher profit score than the latest BKS.

We see that 12.5% of the Cordeau instances have the same profit score. The percentage of Cordeau instances that performs worse than I3CH is equal to 52.5% and the percentage of Cordeau instances with a higher profit score than I3CH is equal to 35%. The profit scores of 4 out of 28 Cordeau instances that perform better than I3CH are equal to the BKS available in 2012 and the profit scores of 3 out of 28 Cordeau instances are higher than the profit scores of the BKS available in 2012. It follows that the profit scores of these 3 instances are higher than the latest BKS from Gunawan et al. (2015).

The percentage of instances with the same profit score is equal to 48.68% and the percentage of instances with a higher profit score is equal to 28.62%. The number of instances that perform better than the BKS available in 2012 is equal to 16 and the number of instances that have a higher profit score than the latest BKS (Gunawan et al., 2015) is equal to 10. The profit scores of these 10 instances can be found in table 2 and the routes can be seen in chapter A.1.9 in the appendix. We have to take into account that the value of I_{max} is not the same. Hu and Lim (2014) use $I_{max} = 3000$, while we use $I_{max} = 10$.

Table 2: New best solutions discovered by GELD

Name	m	New best	Name	m	New best	Name	m	New best
r202	2	1349	rc202	2	1515	rc104	4	1065
r203	2	1422	rc201	3	1699	pr17	4	936
r206	2	1450	pr20	3	1687	pr20	4	2069
r210	2	1428						

More detailed results can be seen in table 3. All the results of the implementation of I3CH can be seen in chapter A.1 in the appendix.

Table 3: Detailed results compared to I3CH and BKS available in 2012

Instances	*num*	I3CH ($I_{\max} = 3000$)		GELD ($I_{\max} = 10$)					
		AG_{BKS} (%)	AT (s)	# Same	AG_{I3CH} (%)	AT (s)	RR (%)	LS (%)	SA (%)
m=1									
Solomon 100	29	0.69	26.71	20	−0.60	60.77	0.21	49.60	47.54
Solomon 200	27	1.34	132.14	9	−0.86	1242.00	0.05	52.97	45.19
Cordeau 1-10	10	1.05	109.01	3	0.86	380.20	0.08	51.32	46.71
Cordeau 11-20	10	3.79	130.23	0	−0.86	729.88	0.05	52.14	45.87
m=2									
Solomon 100	29	0.47	69.31	15	−0.23	257.29	0.11	49.76	47.38
Solomon 200	27	0.43	463.80	7	−0.40	6288.64	0.01	56.35	41.20
Cordeau 1-10	10	0.94	247.05	2	0.30	2207.46	0.02	54.31	43.32
Cordeau 11-20	10	2.69	304.63	1	−0.55	4540.53	0.02	55.17	42.70
m=3									
Solomon 100	29	0.16	135.86	19	0.04	628.83	0.06	51.48	45.72
Solomon 200	27	−0.01	89.24	24	0.00	872.57	0.03	57.27	35.90
Cordeau 1-10	10	0.35	423.99	1	0.94	7013.54	0.01	55.23	42.48
Cordeau 11-20	10	1.00	496.96	1	0.54	11607.45	0.01	55.05	42.52
m=4									
Solomon 100	29	0.06	199.54	17	−0.04	1496.10	0.03	53.29	44.13
Solomon 200	27	0.00	0.17	27	0.00	2.19	0.00	0.00	0.00
Cordeau 1-10	10	0.05	566.49	1	0.66	14483.88	0.01	54.44	43.12
Cordeau 11-20	10	−0.64	728.64	1	0.55	24243.15	0.01	54.24	43.41
Total	304			148					
Average		0.59	200.94		−0.11	3124.52	0.05	48.48	40.00

For the Solomon 100 instances, the average gap (AG) between GELD and I3CH is equal to −0.21%. This means that on average, GELD performs better on the Solomon 100 instances than I3CH. The average gap for the Solomon 200 instances is equal to −0.32%. On average, GELD performs better than I3CH on the Solomon 200 instances.

The average gap between GELD and I3CH for the Cordeau 1-10 instances is equal to 0.69%, which means that GELD performs worse than I3CH on the Cordeau 1-10 instances. For the Cordeau 11-20 instances, the average gap between GELD and I3CH is equal to −0.08%. On average, GELD performs better on the Cordeau 11-20 instances than I3CH.

The gap between GELD and I3CH for all the instances is equal to −0.11%. This means that GELD performs better than I3CH on the 304 benchmark instances. This does not hold for the Cordeau 1-10 instances. Better results for these instances could be obtained by using a higher value of I_{max}.

However, the computation times for all instances are much higher. The average computation times (AT) are approximately 15 times higher than the computation times from Hu and Lim (2014). A pos-

sible explanation for this phenomena is the use of a different computer and the way the heuristic is implemented. During the LS and SA, multiple copies of the initial and best found solution are made. This could have an effect on the memory and speed of the computer.

The RR is a fast procedure. The heuristic only spends 0.05% of its time on the RR. Both the LS and SA procedures are much slower. 48.47% and 40.01% of the total time are spend on the LS and SA procedures. The neighborhood search, checking whether a solution is feasible and caching the best found solution so far may take much time. When we compare these results with the computation times of the components in Hu and Lim (2014), we see that I3CH also spends the least amount of time on the RR. According to Hu and Lim (2014), both the LS and SA components are approximately 2 times slower than RR. In our research, LS is approximately 970 times slower and SA is approximately 800 times slower than RR. This can be explained by using a different computer and the way the heuristic is implemented, which is mentioned above.

For the $m = 4$ Solomon 200 instances, the initialization procedure already found a solution where all locations are visited. This means that the RR, LS and SA components are not used.

In each iteration of the heuristic, the best solution is chosen as input for the next iteration. This solution is the current best solution, the solution obtained by RR, the solution from LS or the solution obtained by SA, breaking ties arbitrarily. The distribution of the chosen solutions for each value of m is shown in figure 6.

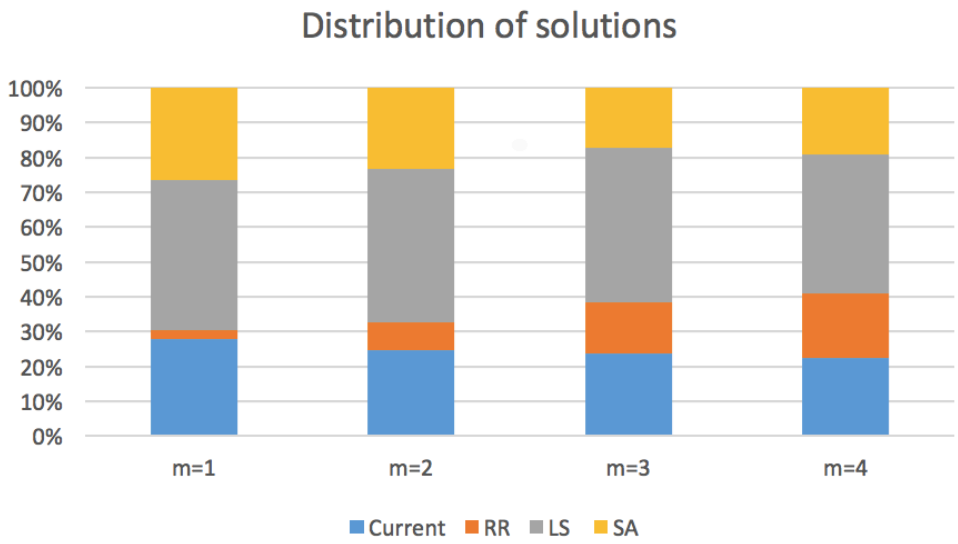


Figure 6: Distribution of chosen solutions

For all m, most better solutions are found by LS. The RR contribute the least in finding better solutions, but the proportion of chosen RR solutions increase for higher values of m. This does not hold for the other components. The proportion of the Current, LS and SA components decrease for higher values of m. We have to take into account that in case of a tie, the component is chosen arbitrarily among the components with the same results.

In chapter 6.3, we investigate the influence of RR on the solution quality.

6.3 I3CH without route recombination

Other papers studying TOPTW, mentioned in the literature review (chapter 3), used a local search or simulating annealing procedure for the TOPTW, but the route recombination has not been used previously. We investigate whether the RR influences the solution quality. For 10 instances, we check whether the solution remains the same when the RR is not used in the heuristic. This procecure is repeated 3 times with different seeds $seed = \{1, 3, 5\}$. For all these instances, the RR component improves the solution quality. The results are shown in figure 7.

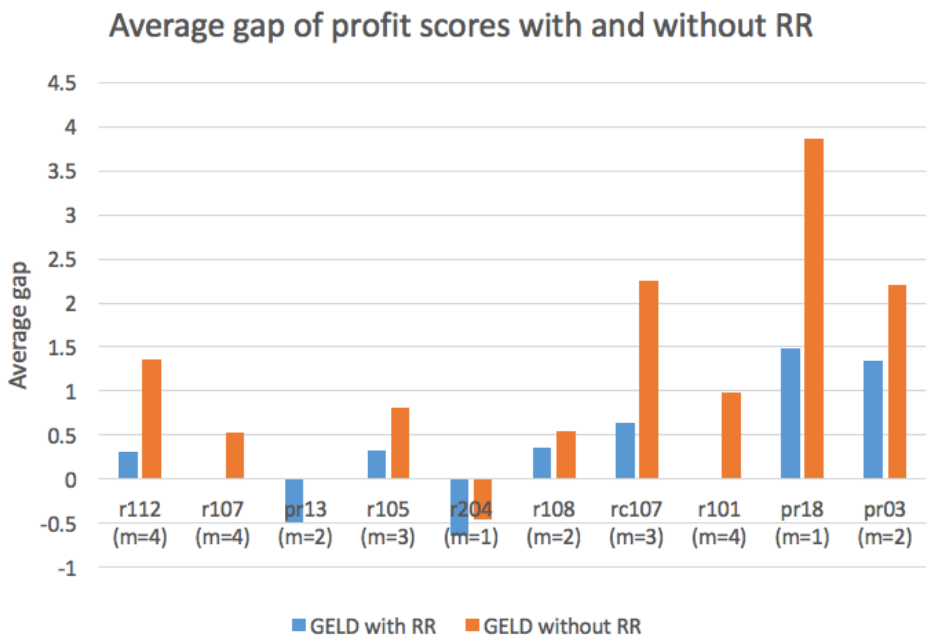


Figure 7: Profit scores for different instances when the RR is and is not implemented

The left blue bars indicate the average gap between GELD with RR and the BKS available in 2012 used in Hu and Lim (2014). The right orange bars indicate the average gap between GELD without RR and the BKS available in 2012 used in Hu and Lim (2014).

For each instance, we see that the average gap between GELD with RR and the BKS available in 2012 is lower than the average gap between GELD without RR and the BKS available in 2012. The average gap between GELD with RR and the BKS available in 2012 is equal to 0.33%, while the average gap between GELD without RR and the BKS available in 2012 is equal to 1.21%. This indicates that the RR component influences the solution quality. In figure 6, we see that the RR is chosen multiple times to improve the solution quality.

Hu and Lim (2014) also determined the average performance of various combinations of components. The average gap between I3CH with RR and the BKS available in 2012 is equal to 0.59%, while the average gap between I3CH without RR and the BKS available in 2012 is equal to 1.01%. We have to take into account that these results are based on all instances, while we base our results on 10 instances which were repeated 3 times.

It follows that the RR component positively influences the solution quality. The average computation time spend on the RR component is according to our results negligible. Hu and Lim (2014) describe similar results.

6.4 Initialization with different number of solutions

The initialization procedure randomly creates $3N$ solutions and the best one is chosen as input for the I3CH. Hu and Lim (2014) do not mention why they choose to create $3N$ solutions. We investigate whether a different number of solutions in the initialization procedure has an effect on the solution quality. The chosen instances are rc102 ($m = 2$), r101 ($m = 4$), pr20 ($m = 1$), c103 ($m = 4$), r112 ($m = 4$), rc208 ($m = 1$), c101 ($m = 2$), rc201 ($m = 1$), c107 ($m = 3$) and c206 ($m = 1$). The heuristic stops after 10 iterations or whether the profit score is at least the profit score obtained while $3N$ solutions are created in the initialization procedure. This is done to check how much time the heuristic takes to get at least the same solution. The results are shown in table 18 in chapter A.3 in the appendix and a summary of the results is shown in table 4.

Table 4: Results with different number of created solutions in the initialization procedure

	Profit after initialization	Profit after GELD	Time
$1N$	0.9887	0.9989	1.1403
$2N$	0.9946	0.9994	1.3777
$3N$	1	1	1
$4N$	1.0012	0.9967	1.2384
$5N$	1.0012	0.9972	1.0038

All the results are compared with the results from $3N$, the original value in the initialization procedure, which are indicated by the base value 1. It follows that the solution after the initialization is better when more possible solutions are created in the initialization procedure. According to our sample, creating more or less solutions in the initialization has a negative effect on the profit score of GELD and the computation times. However, the differences in the profit scores after applying GELD are small. The computation times may be longer for $1N$ and $2N$, because better solutions must be found by the LS, SA and RR components. The computation times may be longer for $4N$ and $5N$, because the initialization procedure is slower.

More detailed results from the $m = 1$ and $m = 4$ instances are shown in figure 5.

Table 5: Results with different number of created solutions in the initialization procedure for $m = 1$ and $m = 4$

$m = 1$	Profit after init	Profit after GELD	Time	$m = 4$	Profit after init	Profit after GELD	Time
$1N$	0.9823	0.9974	1.0695	$1N$	0.9943	1.0000	1.3221
$2N$	0.9914	0.9985	1.3479	$2N$	0.9965	1.0000	1.4466
$3N$	1	1	1	$3N$	1	1	1
$4N$	1.0043	0.9924	1.2791	$4N$	1.0000	1.0000	1.1194
$5N$	1.0043	0.9933	0.8000	$5N$	1.0000	1.0000	1.5382

All these results are compared with the results from $3N$, the original value in the initialization procedure, which are indicated by the base value 1. For $m = 1$, the average profit scores after initialization are lower when less solutions are created in the initialization procedure. This also holds for the average profit scores after GELD is applied. Except for $5N$, computation times are higher.

For $m = 4$, the average profit scores after initialization are lower when less solutions are created in the initialization procedure. This is in line with the other results. The average profit scores after applying GELD are the same for all created solutions in the initialization procedure. This does not hold for the computation times. The lowest computation times belongs to $3N$.

According to the results of this sample, creating less than $3N$ solutions in the initialization procedure does not have a positive effect on the solution quality and computation times, while there is no clear evidence that creating more than $3N$ solution in the initialization procedure has a positive effect on the solution quality and computation times. Creating $3N$ solutions in the initialization procedure results in a relatively good solution quality and relatively fast computation times.

7 TOPTWST

As mentioned in chapter 2.3, technicians visit many customers per day and try to repair the products of the customers. It may be possible that certain technicians are not able to repair all kind of products, because they are specialized in a certain type of product. This means that technicians are not able to visit all customers. Chen et al. (2015) takes this phenomenom into account. Chen et al. (2015) and Hu and Lim (2014) do not have the same solution approach. We only extend the I3CH by taking into account that technicians are not able to visit all customers. We do not take the change in productivity due to learning into account, which is implemented in the model described in Chen et al. (2015).

7.1 Modification of I3CH

The I3CH has to be changed in several ways to take the TOPTWST into account. For every location, we have to determine whether the location can be visited by a certain route. The locations are divided into l groups, where $P_l \subset V\backslash\{0\}$, $\bigcup P_l = V\backslash\{0\}$ and $\bigcap P_l = \emptyset$. V_l is equal to the number of routes which belong to group l. This means that a route which belong to group l can only visit locations that belong to the group P_l. The locations are randomly divided among the l groups.

Each time a location is added to a route, a feasibility check determines whether this route is still feasible. We have to implement an extra check. Before we check whether the newly created route is feasible, we have to check whether the added location belongs to the same group as the other locations on the route. If this is not the case, the location cannot be inserted.

Third, we have to modify the set packing formulation. The following variable is created:

$$f_{kl} = \begin{cases} 1 & \text{if route } k \text{ belongs to group } l, \\ 0 & \text{otherwise.} \end{cases}$$

The modified set packing formulation will be as follows:

$$\max \quad \sum_{k=1}^{S_{pool}} q_k x_k \tag{14}$$

$$\text{s.t.} \quad \sum_{k=1}^{S_{pool}} a_{jk} x_k \leq 1 \quad \forall j \in V\backslash\{0\}, \tag{15}$$

$$\sum_{k=1}^{S_{pool}} x_k \leq m, \tag{16}$$

$$\sum_{k=1}^{S_{pool}} f_{kl} x_k = V_l \quad \forall l, \tag{17}$$

$$x_k \in \{0,1\} \qquad \forall k \in \{1,2,\ldots,S_{pool}\}. \tag{18}$$

Constraints (17) make sure that the number of routes that is chosen from the pool which belong to group l is the same as previously determined in the heuristic.

The Solomon 100 rc-instances are used for the TOPTWST. In real life, technicians may visit different customers who lives in the same city, which are represented by the clustered data, and customers who lives in rural area, which are respresented by the random generated data.

7.2 Results

The locations are divided into 2 groups. This means that $l \in \{A, B\}$. The TOPTWST has been implemented twice and repeated for 3 times. First, the number of routes is equal to 4. Two routes belong to group A and two routes belong to group B. The locations are randomly divided into two groups, such that on average 50% of the locations belongs to group A and 50% of the locations belongs to group B.

Second, the number of routes is equal to 3. One route belongs to group A and the other two routes belong to group B. The locations are still randomly divided, but on average 33.33% of the locations belongs to group A and 66.67% of the locations belongs to group B.

For this problem instance, there are no best known solutions available. The results of TOPTWST are shown in table 6 and table 7.

Table 6: Results of TOPTWST for $m = 4$

Instance (m=4)	No specialization		With specialization			
	Profit	AT (s)	Average Profit after			AT (s)
			Init	5	10	
rc101	811	773.77	637.33	787.67	787.67	237.34
rc102	895	1442.17	744.00	852.00	854.67	442.92
rc103	974	1662.33	789.33	904.33	904.33	565.75
rc104	1065	1924.99	851.00	990.67	1001.00	741.97
rc105	875	934.15	703.33	822.00	827.33	296.39
rc106	909	1203.36	728.67	848.00	851.33	393.67
rc107	982	1601.63	814.00	924.33	924.33	500.71
rc108	1025	1981.82	816.00	966.67	966.67	567.52

Table 7: Results of TOPTWST for $m = 3$

Instance (m=3)	No specialization		With specialization			
	Profit	AT (s)	Average Profit after			AT (s)
			Init	5	10	
rc101	621	338.80	497.33	593.33	597.33	152.88
rc102	710	643.77	567.67	771.67	683.67	275.48
rc103	764	633.32	604.00	704.00	705.00	263.51
rc104	834	965.16	662.33	744.67	753.67	329.58
rc105	682	507.14	555.67	624.33	627.00	172.52
rc106	706	423.52	549.67	629.00	629.00	207.38
rc107	768	673.03	619.33	703.67	707.67	286.49
rc108	795	685.99	635.67	715.67	720.67	322.35

For all instances in table 6 and 7, the average profit scores after 10 iterations of the TOPTWST are lower than the profit scores of TOPTW. This also holds for the computation times. The computation times of the instances with specialization are on average 2.76 times lower than the computation times of the instances without specialization. An explanation for the lower profit scores and lower computation times are the extra restrictions that are added to the heuristic. For example, the 2-exchange, 2-relocate and 2-opt operators only works on 2 routes which belong to the same group. This means that there are less possibilities for changes, which decreases the computation times. It is not always possible to add the location with the highest profit score to the route, because this location might belong to another group. It follows that the profit scores with the specialization taken into account can never be higher than the profit scores without the specialization taken into account.

8 Conclusion

The aim of the thesis was to first implement the I3CH of Hu and Lim (2014) and reproduce their results for different instance benchmark sets. We made 2 adjustments and investigated which effect it had on the results. Second, the TOPTW has been extended to the TOPTWST by taking specialized technicians into account which are not able to visit all locations.

The results of GELD, our reproduced heuristic, and I3CH of Hu and Lim (2014) were not the same. Due to high computation times, we were not able to use the same number of iterations I_{max} in the heuristic. We used a different computer and our heuristic made multiple copies of the initial and best found solutions, which could have an effect on the memory and speed of the computer. When we used 10 iterations instead of 3000 iterations, 8 out of 10 instances from our sample were equal to or higher than the results of Hu and Lim (2014) and the average profit scores of 5 out of 10 sample instances were equal to the best known solutions available in 2012. Using more than 10 iterations led to higher computation times, but did not necessarily lead to better results. We made a tradeoff between efficiency and effectiveness and used 10 iterations for our heuristic.

The results of GELD were better than the results of Hu and Lim (2014). The total profit score of GELD was 0.11% lower than their total profit score. The percentage of instances with the same profit score was equal to 48.68% and the percentage of instances with a lower profit score was equal to 22.70%. The number of instances that performed better than I3CH was 87 (28.61%). The number of instances with a higher profit score than the BKS available in 2012 was equal to 16. When we compared these results with the latest BKS (Gunawan et al., 2015), we found 10 new best solutions.

Other papers studying the TOPTW used a local search or simulating annealing procedure for the TOPTW, but the route recombination has not been used previously. According to our research, the RR component did influence the quality of the solutions from our sample. The average gap between GELD with the RR component and the BKS available in 2012 was equal to 0.33%, while the average gap between GELD without the RR component and the BKS available in 2012 was equal to 1.21%. Hu and Lim (2014) had similar results.

Hu and Lim (2014) did not mention why they created $3N$ solutions in the initialization procedure. We investigated whether creating more or less than $3N$ solutions would influence the results from our sample. Enumerating more solutions in the initialization procedure led to higher profit scores after the initialization, but it did not lead to better results or lower computation times. Enumerating less solutions in the initialization procedure led to lower profit scores after initialization and did not lead to better results or lower computation times.

TOPTWST took the specialization of technicians into account. The computation times of TOPTWST were approximately 2.76 times lower than the computation times of TOPTW, because extra restrictions were added to the heuristic.

Our heuristic GELD did not have the same results as I3CH from Hu and Lim (2014), but the profit scores of GELD were on average 0.11% higher than the profit scores of Hu and Lim (2014). We found 16 solutions with a better solution than the BKS available in 2012 and 10 solutions with a better solution than the latest BKS (Gunawan et al., 2015).

9 References

S. Butt and T. Cavalier. A heuristic for the multiple tour maximum collection problem. *Computers and Operations Research*, 21: 101–111, 1994.

A. Campbell and M. Savelsbergh. Efficient insertion heuristics for vehicle routing and scheduling problems. *Transportation Science*, 38(3): 369–378, 2004.

X. Chen, B. Thomas, and M. Hewitt. The technician routing problem with experience-based service times. *Omega*, 2015.

J. Cordeau, M. Gendreau, and G. Laporte. A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Annals of Operations Research*, 63(1): 3–27, 1997.

L. Gambardella, R. Montemanni, and D. Weyland. Coupling ant colony systems with strong local searches. *European Journal of Operational Research*, 220(3): 831–843, 2012.

B. Golden, L. Levy, and R. Vohra. The orienteering problem. *Naval Research Logistics (NRL)*, 34(3): 307–318, 1987.

A. Gunawan, H. Lau, and K. Lu. The latest best known solutions for the team orienteering problem with time windows (toptw) benchmark instances (per 16 june 2015), 2015. URL `http://research.larc.smu.edu.sg/downloads/OPLib/Publications/SupplementTOPTW.pdf`. Accessed: 2016-06-29.

Q. Hu and A. Lim. An iterative three-component heuristic for the team orienteering problem with time windows. *European Journal of Operational Research*, 232(2): 276–286, 2014.

N. Labadie, J. Melechovský, and R. Wolfler Calvo. Hybridized evolutionary local search algorithm for the team orienteering problem with time windows. *Journal of Heuristics*, 17(6): 729–753, 2011.

N. Labadie, R. Mansini, J. Melechovsk, and R. Calvo. The team orienteering problem with time windows: An lp-based granular variable neighborhood search. *European Journal of Operational Research*, 220 (1): 15–27, 2012.

S. Lin and V. Yu. A simulated annealing heuristic for the team orienteering problem with time windows. *European Journal of Operational Research*, 217(1): 94–107, 2012.

H. Lourenço, O. Martin, and T. Stützle. A beginner's introduction to iterated local search. In *Proceeding of the 4th Metaheuristics Internationsl Conference*, pages 1–11, Porto, Portugal, 2001.

R. Montemanni and L. Gambardella. Ant colony system for team orienteering problems with time windows. *Foundations of Computing and Decision Sciences*, 32: 287–306, 2009.

R. Montemanni, D. Weyland, and L. Gambardella. An enhanced ant colony system for the team orienteering problem with time windows. In *2011 International symposium on computer science and society (ISCCS)*, pages 381–384. IEEE, 2011.

M. Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2): 254–265, 1987.

W. Souffriau, P. Vansteenwegen, J. Vertommen, G. Berghe, and D. Van Oudheusden. A personalized tourist trip design algorithm for mobile tourist guides. *Applied Artificial Intelligence*, 22(10): 964–985, 2008.

H. Tang and E. Miller-Hooks. A tabu search heuristic for the team orienteering problem. *Computer and Operations Research*, 32: 1379–1407, 2005.

F. Tricoire, M. Romauch, K. Doerner, and R. Hartl. Heuristics for the multi-period orienteering problem with multiple time windows. *Computers and Operations Research*, 37(2): 351–367, 2010.

T. Tsiligirides. Heuristic methods applied to orienteering. *Journal of the Operational Research Society*, 35(9): 797–809, 1984.

P. Vansteenwegen, W. Souffriau, G. Vanden Berghe, and D. Van Oudheusden. Iterated local search for the team orienteering problem with time windows. *Computers and Operations Research*, 12(36): 3281–3290, 2009.

P. Vansteenwegen, W. Souffriau, and D. Van Oudheusden. The orienteering problem: A survey. *European Journal of Operational Research*, 209(1): 1–10, 2011.

A Appendix

A.1 Results of GELD

A.1.1 Solomon $m = 1$

Table 8: Results Solomon instances for $m = 1$. The BKS are the best known solutions available in 2012 and used in Hu and Lim (2014).

Name	BKS	I3CH		$I_{max} = 3000$	GELD		$I_{max} = 10$	# Best solution chosen				Time spend in component		
		Profit	Gap BKS (%)	Time (s)	Profit	Gap I3CH (%)	Time (s)	# Current	# RR	# LS	# SA	RR (s)	LS (s)	SA (s)
c101	320	320	0.00	21.83	320	0.00	52.68	1	0	2	7	0.16	25.13	25.43
c102	360	360	0.00	28.05	360	0.00	69.87	4	0	5	1	0.11	36.72	27.37
c103	400	400	0.00	27.06	400	0.00	51.91	9	0	1	0	0.09	24.74	24.72
c104	420	420	0.00	27.12	420	0.00	44.16	2	0	5	3	0.06	21.14	21.17
c105	340	340	0.00	23.45	340	0.00	32.77	1	0	5	4	0.11	15.85	15.71
c106	340	340	0.00	23.64	340	0.00	30.25	1	0	9	0	0.13	13.72	15.23
c107	370	370	0.00	24.72	370	0.00	34.66	2	0	4	4	0.14	17.40	15.83
c108	370	370	0.00	24.78	370	0.00	34.75	4	0	4	2	0.12	17.33	15.82
c109	380	380	0.00	26.28	380	0.00	45.93	1	0	8	1	0.15	24.24	19.86
r101	198	198	0.00	20.44	198	0.00	29.74	2	0	4	4	0.12	14.63	14.28
r102	286	286	0.00	29.29	286	0.00	65.05	3	0	4	3	0.13	31.14	32.17
r103	293	293	0.00	28.77	293	0.00	57.64	3	0	2	5	0.14	29.70	25.76
r104	303	298	1.65	27.32	303	-1.68	70.52	3	0	7	0	0.15	36.84	31.74
r105	247	247	0.00	25.95	247	0.00	52.62	4	0	2	4	0.12	25.74	25.75
r106	293	293	0.00	29.41	293	0.00	63.27	4	0	2	4	0.11	30.69	30.81
r107	299	297	0.67	27.80	295	0.67	68.22	2	0	5	3	0.12	33.29	33.07
r108	308	306	0.65	29.67	303	0.98	66.10	2	0	6	2	0.14	33.28	30.96
r109	277	277	0.00	31.06	277	0.00	76.45	5	0	3	2	0.13	36.07	38.87
r110	284	284	0.00	33.90	284	0.00	110.80	4	0	2	2	0.13	54.53	54.32
r111	297	295	0.67	27.70	297	-0.68	90.69	2	0	7	1	0.14	47.90	40.96
r112	298	289	3.02	31.97	298	-3.11	97.15	2	1	4	3	0.19	49.66	45.42
rc101	219	219	0.00	21.76	219	0.00	37.79	3	0	2	5	0.11	18.34	18.58
rc102	266	266	0.00	25.50	266	0.00	59.00	0	0	6	4	0.12	28.79	28.95
rc103	266	266	0.00	27.09	266	0.00	72.04	4	1	3	2	0.14	35.67	34.97
rc104	301	301	0.00	27.20	301	0.00	93.63	2	0	4	4	0.12	46.32	45.98
rc105	244	244	0.00	26.36	241	1.23	71.54	4	0	2	4	0.12	35.10	35.26
rc106	252	250	0.79	25.00	252	-0.80	52.67	2	0	3	5	0.11	26.30	25.19
rc107	277	274	1.08	26.27	277	-1.09	63.53	4	1	2	3	0.13	30.91	31.19
rc108	298	264	11.41	25.12	298	-12.88	66.97	3	0	3	4	0.13	33.06	32.40
c201	870	870	0.00	70.13	870	0.00	236.84	3	0	3	4	0.59	113.48	105.92
c202	930	930	0.00	87.59	930	0.00	308.12	3	0	4	3	0.42	150.28	145.20
c203	960	960	0.00	92.26	960	0.00	452.33	5	0	4	1	0.49	239.43	198.99
c204	980	970	1.02	117.39	970	0.00	538.28	1	0	9	0	0.48	325.94	200.20
c205	910	900	1.10	70.73	910	-1.11	218.30	2	0	5	3	0.43	106.79	103.15
c206	930	920	1.08	75.74	930	-1.09	328.68	3	1	6	0	0.64	174.47	145.25
c207	930	930	0.00	77.39	930	0.00	285.55	4	0	6	0	0.48	140.30	135.02
c208	950	950	0.00	84.05	950	0.00	394.68	4	0	6	0	0.49	213.68	169.96
r201	797	789	1.00	101.83	789	0.00	710.86	3	0	4	3	0.47	359.20	337.59
r202	929	930	-0.11	175.63	924	0.65	1728.24	4	1	4	1	0.73	912.29	785.58
r203	1021	1020	0.10	221.36	1017	0.29	2458.31	1	0	6	3	0.60	1293.48	1126.64
r204	1086	1073	1.20	236.89	1093	-1.86	2194.20	1	2	3	4	0.89	1156.64	996.41
r205	953	946	0.73	129.32	951	-0.53	1160.59	1	0	5	4	0.55	601.98	534.30
r206	1029	1021	0.78	169.27	1032	-1.08	2466.17	3	0	3	4	0.60	1465.89	965.21
r207	1072	1050	2.05	192.81	1071	-2.00	2293.30	3	1	4	2	0.89	1192.05	1061.14
r208	1112	1092	1.80	230.02	1109	-1.56	2131.94	3	0	5	2	0.66	1097.48	994.63
r209	950	948	0.21	136.48	944	0.42	1410.02	2	0	6	2	0.56	725.97	650.25
r210	987	982	0.51	176.91	984	-0.20	1887.44	2	0	3	5	0.62	939.04	911.48
r211	1046	1013	3.15	167.39	1029	-1.58	1899.67	4	0	3	3	0.73	1001.60	852.50
rc201	795	795	0.00	80.94	795	0.00	390.06	3	0	6	1	0.40	211.95	172.35
rc202	936	924	1.28	129.34	938	-1.52	1182.59	1	0	8	1	0.50	608.18	556.34
rc203	1003	966	3.69	134.28	992	-2.69	2238.72	5	0	3	2	0.60	1214.31	1002.81
rc204	1140	1093	4.12	167.54	1136	-3.93	1746.03	0	0	7	3	0.70	959.80	762.45
rc205	859	847	1.40	99.16	845	0.24	864.03	3	0	6	1	0.45	447.56	402.34
rc206	895	863	3.58	98.37	863	0.00	811.27	1	0	6	3	0.46	439.81	356.99
rc207	983	957	2.64	122.04	977	-2.09	1485.32	0	2	7	1	0.76	779.63	684.80
rc208	1053	1003	4.75	123.04	1038	-3.49	1712.37	1	0	4	5	0.63	890.30	797.42

A.1.2 Cordeau $m = 1$

Table 9: Results Cordeau instances for $m = 1$. The BKS are the best known solutions available in 2012 and used in Hu and Lim (2014).

Name	BKS	I3CH		$I_{max} = 3000$	GELD		$I_{max} = 10$	# Best solution chosen				Time spend in component		
		Profit	Gap BKS (%)	Time (s)	Profit	Gap I3CH (%)	Time (s)	# Current	# RR	# LS	# SA	RR (s)	LS (s)	SA (s)
pr01	308	305	0.97	20.8	305	0.00	56.38	4	0	3	3	0.23	28.41	26.35
pr02	404	394	2.48	47.9	404	-2.54	167.48	3	1	3	3	0.34	97.59	75.32
pr03	394	394	0.00	72.9	394	0.00	167.27	3	1	3	3	0.25	81.97	81.02
pr04	489	489	0.00	109.3	454	7.16	348.94	2	1	3	4	0.31	173.51	167.42
pr05	595	594	0.17	185.4	580	2.36	726.41	3	0	5	2	0.31	361.28	348.53
pr06	590	590	0.00	189.9	579	1.86	759.77	3	1	3	3	0.47	411.48	333.44
pr07	298	298	0.00	26.5	291	2.35	56.34	4	0	5	1	0.18	27.24	27.66
pr08	463	454	1.94	77.4	463	-1.98	203.99	5	0	3	2	0.34	104.84	93.08
pr09	493	490	0.61	137.8	493	-0.61	537.95	1	1	5	3	0.45	266.53	262.17
pr10	594	568	4.38	222.2	568	0.00	777.51	5	0	4	1	0.24	398.43	361.06
pr11	351	353	-0.57	30.8	351	0.57	64.34	3	0	2	5	0.19	30.30	31.35
pr12	442	433	2.04	59.8	438	-1.15	176.07	3	1	4	2	0.26	88.86	81.56
pr13	461	466	-1.08	89.5	437	6.22	530.01	4	0	4	2	0.29	270.94	249.59
pr14	567	521	8.11	144.4	550	-5.57	1020.96	4	0	4	2	0.31	536.61	469.46
pr15	685	707	-3.21	248.2	701	0.85	1829.85	1	1	2	6	0.50	932.64	866.82
pr16	674	619	8.16	228.6	617	0.32	1042.18	1	1	6	2	0.51	545.35	469.62
pr17	362	360	0.55	34.7	348	3.33	105.82	2	1	2	5	0.21	51.55	51.92
pr18	539	497	7.79	99.0	531	-6.84	337.91	1	2	4	3	0.42	167.15	161.98
pr19	562	538	4.27	164.6	518	3.72	716.65	4	0	4	2	0.34	365.23	334.39
pr20	667	588	11.84	202.7	647	-10.03	1475.02	5	0	5	0	0.43	817.32	631.08

A.1.3 Solomon $m = 2$

Table 10: Results Solomon instances for $m = 2$. The BKS are the best known solutions available in 2012 and used in Hu and Lim (2014).

Name	BKS	I3CH		$I_{max} = 3000$	GELD		$I_{max} = 10$	# Best solution chosen				Time spend in component		
		Profit	Gap BKS (%)	Time (s)	Profit	Gap I3CH (%)	Time (s)	# Current	# RR	# LS	# SA	RR (s)	LS (s)	SA (s)
c101	590	590	0.00	53.43	590	0.00	152.18	4	0	4	2	0.23	73.48	70.40
c102	660	660	0.00	78.37	660	0.00	250.67	4	0	6	0	0.65	126.36	107.02
c103	720	720	0.00	116.12	720	0.00	302.14	0	0	6	4	0.77	155.03	126.10
c104	760	760	0.00	94.39	750	1.32	215.46	6	0	4	0	1.06	107.46	99.06
c105	640	640	0.00	70.56	640	0.00	151.46	4	1	4	1	0.23	75.42	71.29
c106	620	620	0.00	149.22	620	0.00	133.74	5	0	4	1	0.53	61.51	67.02
c107	670	670	0.00	65.38	670	0.00	169.64	4	0	6	0	0.24	85.95	78.40
c108	680	680	0.00	85.92	680	0.00	180.37	2	0	8	0	0.21	90.58	83.77
c109	720	720	0.00	69.42	720	0.00	197.61	4	0	4	2	0.25	97.61	92.80
r101	349	349	0.00	42.12	349	0.00	81.52	3	0	2	5	0.11	39.04	39.66
r102	508	508	0.00	62.44	508	0.00	219.50	3	0	4	3	0.19	111.83	101.00
r103	522	519	0.57	68.25	518	0.19	307.82	3	0	6	1	0.20	151.58	147.36
r104	552	549	0.54	75.34	546	0.55	422.42	3	1	3	3	0.25	220.62	192.48
r105	453	447	1.32	56.18	453	-1.34	192.61	3	0	3	4	0.17	95.89	92.26
r106	529	529	0.00	63.53	529	0.00	250.15	6	1	3	0	0.18	123.18	119.48
r107	535	533	0.37	63.18	535	-0.38	372.75	3	1	5	1	0.24	184.87	178.78
r108	558	550	1.43	65.66	556	-1.09	388.86	1	2	3	4	0.38	192.74	186.38
r109	506	506	0.00	60.45	505	0.20	251.24	6	1	2	1	0.23	122.41	158.88
r110	525	525	0.00	68.85	525	0.00	327.14	3	0	4	3	0.19	160.04	158.88
r111	544	542	0.37	66.97	544	-0.37	436.67	0	1	4	5	0.53	210.47	187.86
r112	544	534	1.84	62.96	544	-1.87	379.87	3	1	2	4	0.25	191.46	178.93
rc101	427	427	0.00	52.32	427	0.00	136.92	6	1	2	1	0.16	66.67	66.81
rc102	505	505	0.00	59.82	505	0.00	280.19	4	1	3	2	0.20	137.96	136.89
rc103	524	519	0.95	60.32	510	1.73	228.40	3	0	2	5	0.18	113.20	108.95
rc104	575	556	3.30	59.94	574	-3.24	305.79	2	0	6	2	0.21	148.87	150.60
rc105	480	480	0.00	58.61	480	0.00	250.07	0	1	5	4	0.20	123.71	121.41
rc106	483	481	0.41	55.96	483	-0.42	230.34	2	0	3	5	0.19	113.18	112.03
rc107	534	529	0.94	62.93	534	-0.95	301.28	4	1	1	4	0.16	146.60	147.81
rc108	556	547	1.62	61.43	553	-1.10	344.47	3	0	4	3	0.20	185.19	152.61
c201	1460	1450	0.68	321.24	1450	0.00	2476.02	3	0	5	2	0.77	1306.35	1092.93
c202	1470	1470	0.00	405.95	1470	0.00	3304.07	2	0	4	4	0.78	1821.78	1420.75
c203	1480	1470	0.68	458.16	1480	-0.68	3581.15	2	0	5	3	0.90	1981.74	1542.13
c204	1480	1480	0.00	498.52	1480	0.00	3526.28	1	0	5	4	0.90	1938.85	1528.71
c205	1470	1450	1.36	322.19	1470	-1.38	2975.76	3	0	4	3	0.66	1571.39	1353.09
c206	1480	1480	0.00	355.88	1480	0.00	3285.67	2	0	8	0	1.15	1763.60	1467.30
c207	1490	1470	1.34	423.54	1490	-1.36	3321.60	2	1	4	3	1.22	1723.37	1542.38
c208	1490	1470	1.34	424.32	1480	-0.68	3674.84	1	0	7	2	1.18	2099.50	1514.97
r201	1250	1254	-0.32	333.57	1250	0.32	5144.16	4	1	4	1	0.98	2792.31	2253.56
r202	1347	1344	0.22	588.47	1349	-0.37	8263.26	3	0	3	4	0.73	4883.23	3207.30
r203	1414	1416	-0.14	815.90	1422	-0.42	11479.78	1	0	6	3	0.85	6422.02	4789.68
r204	1458	1458	0.00	33.50	1458	0.00	319.01	0	0	0	0	0.00	0.00	0.00
r205	1379	1380	-0.07	606.08	1378	0.14	10564.48	2	2	4	2	1.45	6291.90	4072.62
r206	1440	1427	0.90	739.91	1450	-1.61	14158.28	0	2	4	4	1.52	8245.41	5618.59
r207	1458	1458	0.00	453.80	1458	0.00	2721.44	0	0	1	0	0.04	1795.72	562.84
r208	1458	1458	0.00	4.28	1458	0.00	7.26	0	0	0	0	0.00	0.00	0.00
r209	1405	1404	0.07	600.28	1413	-0.64	12022.33	2	1	5	2	1.29	6907.28	4890.99
r210	1423	1415	0.56	728.49	1428	-0.99	12618.26	1	0	5	4	0.99	7088.27	5271.83
r211	1458	1450	0.55	890.89	1458	-0.55	6128.39	0	0	3	0	0.56	4175.48	1622.77
rc201	1377	1384	-0.51	267.43	1375	0.65	3322.47	5	2	3	0	0.74	1699.91	1553.70
rc202	1509	1500	0.60	386.73	1515	-1.00	5711.65	3	1	1	5	1.18	3139.35	2451.38
rc203	1632	1627	0.31	482.80	1630	-0.18	8935.91	2	1	4	3	1.19	4668.31	4093.20
rc204	1716	1704	0.70	760.75	1707	-0.18	12053.26	3	0	5	2	1.95	6576.91	5273.07
rc205	1458	1452	0.41	311.03	1458	-0.41	6446.16	2	0	5	3	1.35	3803.28	2541.27
rc206	1546	1525	1.36	335.86	1546	-1.38	5748.03	3	0	4	3	0.73	3121.39	2501.75
rc207	1587	1582	0.32	408.56	1574	0.51	7402.24	2	1	5	2	0.87	3800.31	3453.22
rc208	1691	1669	1.30	564.39	1679	-0.60	10601.45	3	0	6	1	0.97	6057.94	4332.07

A.1.4 Cordeau $m = 2$

Table 11: Results Cordeau instances for $m = 2$. The BKS are the best known solutions available in 2012 and used in Hu and Lim (2014).

Name	BKS	I3CH		$I_{max} = 3000$	GELD		$I_{max} = 10$	# Best solution chosen				Time spend in component		
		Profit	Gap BKS (%)	Time (s)	Profit	Gap I3CH (%)	Time (s)	# Current	# RR	# LS	# SA	RR (s)	LS (s)	SA (s)
pr01	502	502	0.00	51.8	502	0.00	270.07	5	0	4	1	0.24	135.76	129.08
pr02	714	714	0.00	127.7	709	0.70	645.17	3	0	5	2	0.28	320.77	304.84
pr03	742	731	1.48	175.6	732	-0.14	918.16	1	2	4	3	0.41	472.67	418.35
pr04	924	917	0.76	270.1	885	3.49	2312.32	1	1	5	3	0.75	1343.35	916.08
pr05	1090	1101	-1.01	410.0	1087	1.27	4611.00	0	1	7	2	0.84	2525.53	1979.15
pr06	1076	1040	3.35	427.6	1043	-0.29	4194.74	0	2	6	2	0.86	2390.33	1704.13
pr07	566	566	0.00	71.8	566	0.00	299.51	1	0	7	2	0.24	154.16	137.61
pr08	834	824	1.20	184.1	821	0.36	1401.85	2	2	4	2	0.79	722.64	649.42
pr09	905	878	2.98	304.9	898	-2.28	2501.24	3	0	6	1	0.43	1358.14	1078.02
pr10	1124	1117	0.62	447.0	1118	-0.09	4920.50	3	2	3	2	0.69	2564.60	2247.13
pr11	566	559	1.24	71.3	559	0.00	393.53	4	0	2	4	0.31	205.71	179.87
pr12	774	768	0.78	143.6	740	3.65	1426.81	2	0	3	5	0.55	774.99	618.42
pr13	831	832	-0.12	238.6	836	-0.48	2655.43	0	3	6	1	0.89	1469.81	1132.25
pr14	1017	978	3.83	337.3	982	-0.41	5180.21	1	2	6	1	0.73	2796.02	2279.04
pr15	1219	1205	1.15	479.1	1200	0.41	8378.57	1	3	5	1	2.03	4429.45	3758.60
pr16	1231	1124	8.69	500.5	1161	-3.29	7106.44	0	3	5	2	0.99	4133.10	2782.40
pr17	652	639	1.99	117.0	646	-1.10	597.88	1	1	7	1	0.38	295.96	287.25
pr18	938	937	0.11	231.0	922	1.60	2313.18	3	1	3	3	0.58	1222.69	1040.99
pr19	1034	1003	3.00	386.1	1023	-1.99	5597.17	1	4	3	2	1.61	3025.63	2438.46
pr20	1232	1155	6.25	541.6	1200	-3.90	11756.05	1	4	5	0	1.23	6696.77	4869.36

A.1.5 Solomon $m = 3$

Table 12: Results Solomon instances for $m = 3$. The BKS are the best known solutions available in 2012 and used in Hu and Lim (2014).

Name	BKS	I3CH		$I_{max} = 3000$	GELD		$I_{max} = 10$	# Best solution chosen				Time spend in component		
		Profit	Gap BKS (%)	Time (s)	Profit	Gap I3CH (%)	Time (s)	# Current	# RR	# LS	# SA	RR (s)	LS (s)	SA (s)
c101	810	810	0.00	303.12	810	0.00	306.56	6	0	4	0	0.46	150.55	146.31
c102	920	920	0.00	237.50	910	1.09	572.29	6	0	4	0	1.20	290.70	263.65
c103	980	990	-1.02	156.48	990	0.00	515.69	4	1	5	0	1.16	270.77	227.83
c104	1030	1030	0.00	155.90	1030	0.00	530.41	4	0	5	1	0.35	287.76	225.26
c105	870	870	0.00	110.20	870	0.00	353.77	2	1	6	1	0.42	184.82	156.90
c106	870	870	0.00	227.43	870	0.00	314.14	6	0	3	1	0.29	145.03	156.61
c107	910	910	0.00	151.77	910	0.00	387.54	3	1	6	0	0.70	201.96	171.74
c108	920	920	0.00	212.63	920	0.00	428.36	4	0	5	1	0.47	218.05	195.20
c109	970	960	1.03	157.18	960	0.00	433.05	4	1	4	1	0.33	214.00	202.20
r101	484	481	0.62	67.70	484	-0.62	187.44	2	1	5	2	0.20	92.89	87.74
r102	694	691	0.43	111.43	691	0.00	650.71	3	0	5	2	0.30	332.48	301.73
r103	747	740	0.94	125.59	735	0.68	923.75	3	0	3	4	0.52	466.12	433.75
r104	777	777	0.00	128.28	778	-0.13	1174.87	3	1	3	3	0.32	613.69	525.72
r105	620	619	0.16	165.84	618	0.16	404.04	1	3	6	0	0.26	203.51	189.74
r106	729	729	0.00	122.66	729	0.00	722.11	1	2	3	4	0.36	367.75	334.38
r107	760	759	0.13	120.55	752	0.92	958.90	2	0	6	2	0.33	486.50	446.83
r108	797	797	0.00	135.74	797	0.00	1077.39	4	1	3	2	0.33	549.42	499.68
r109	710	710	0.00	103.59	710	0.00	663.05	1	0	7	2	0.25	348.28	297.59
r110	737	736	0.14	109.90	736	0.00	892.39	2	2	4	2	0.30	475.70	394.01
r111	774	773	0.13	118.15	772	0.13	978.10	3	3	2	2	0.43	491.81	462.10
r112	776	776	0.00	110.46	776	0.00	860.89	6	1	2	1	0.29	471.46	362.65
rc101	621	621	0.00	87.59	621	0.00	338.80	3	0	2	5	0.18	167.41	162.41
rc102	714	714	0.00	105.43	710	0.56	643.77	2	2	5	1	0.26	357.51	272.14
rc103	764	764	0.00	105.73	764	0.00	663.32	4	0	2	4	0.23	342.48	303.57
rc104	833	834	-0.12	124.44	834	0.00	965.16	0	1	5	4	0.29	504.79	440.99
rc105	682	682	0.00	90.40	682	0.00	507.14	4	0	3	3	0.20	260.24	234.41
rc106	706	706	0.00	88.07	706	0.00	423.52	0	1	6	3	0.21	213.29	197.20
rc107	773	762	1.42	98.69	768	-0.79	673.03	3	2	3	2	0.33	334.95	321.48
rc108	795	789	0.75	107.39	795	-0.76	685.99	2	1	4	3	0.28	343.62	323.46
c201	1810	1810	0.00	3.84	1810	0.00	3.17	0	0	0	0	0.00	0.00	0.00
c202	1810	1810	0.00	7.22	1810	0.00	8.35	0	0	0	0	0.00	0.00	0.00
c203	1810	1810	0.00	7.92	1810	0.00	77.29	0	0	0	0	0.00	0.00	0.00
c204	1810	1810	0.00	11.16	1810	0.00	137.94	0	0	0	0	0.00	0.00	0.00
c205	1810	1810	0.00	12.55	1810	0.00	6.96	0	0	0	0	0.00	0.00	0.00
c206	1810	1810	0.00	18.03	1810	0.00	16.92	0	0	0	0	0.00	0.00	0.00
c207	1810	1810	0.00	16.86	1810	0.00	4.26	0	0	0	0	0.00	0.00	0.00
c208	1810	1810	0.00	20.61	1810	0.00	34.69	0	0	0	0	0.00	0.00	0.00
r201	1441	1439	0.14	981.16	1441	-0.14	6176.18	0	0	9	1	2.39	3532.11	2414.44
r202	1458	1458	0.00	15.29	1458	0.00	939.70	0	0	1	0	0.14	515.02	200.67
r203	1458	1458	0.00	0.23	1458	0.00	4.53	0	0	0	0	0.00	0.00	0.00
r204	1458	1458	0.00	0.21	1458	0.00	1.12	0	0	0	0	0.00	0.00	0.00
r205	1458	1458	0.00	0.28	1458	0.00	1.14	0	0	0	0	0.00	0.00	0.00
r206	1458	1458	0.00	0.26	1458	0.00	4.73	0	0	0	0	0.00	0.00	0.00
r207	1458	1458	0.00	0.32	1458	0.00	1.01	0	0	0	0	0.00	0.00	0.00
r208	1458	1458	0.00	0.21	1458	0.00	0.96	0	0	0	0	0.00	0.00	0.00
r209	1458	1458	0.00	0.29	1458	0.00	1.26	0	0	0	0	0.00	0.00	0.00
r210	1458	1458	0.00	0.25	1458	0.00	3.40	0	0	0	0	0.00	0.00	0.00
r211	1458	1458	0.00	0.35	1458	0.00	1.11	0	0	0	0	0.00	0.00	0.00
rc201	1698	1693	0.29	575.17	1699	-0.35	6310.57	4	0	4	2	0.98	3478.15	2638.37
rc202	1724	1724	0.00	1.15	1724	0.00	1563.89	0	0	1	0	0.17	1115.15	244.24
rc203	1724	1724	0.00	0.34	1724	0.00	7.63	0	0	0	0	0.00	0.00	0.00
rc204	1724	1724	0.00	0.28	1724	0.00	4.78	0	0	0	0	0.00	0.00	0.00
rc205	1709	1719	-0.59	734.36	1709	0.58	6940.27	5	0	5	0	2.29	4068.69	2671.17
rc206	1724	1724	0.00	0.51	1724	0.00	14.81	0	0	0	0	0.00	0.00	0.00
rc207	1724	1724	0.00	0.33	1724	0.00	1291.35	0	0	1	0	0.44	783.90	288.45
rc208	1724	1724	0.00	0.35	1724	0.00	1.46	0	0	0	0	0.00	0.00	0.00

A.1.6 Cordeau $m = 3$

Table 13: Results Cordeau instances for $m = 3$. The BKS are the best known solutions available in 2012 and used in Hu and Lim (2014).

Name	BKS	I3CH		$I_{max} = 3000$	GELD		$I_{max} = 10$	# Best solution chosen				Time spend in component		
		Profit	Gap BKS (%)	Time (s)	Profit	Gap I3CH (%)	Time (s)	# Current	# RR	# LS	# SA	RR (s)	LS (s)	SA (s)
pr01	622	622	0.00	132.1	622	0.00	487.78	2	2	5	1	0.43	239.99	237.33
pr02	942	936	0.64	260.6	940	-0.43	2176.48	2	0	5	3	0.57	1194.19	927.83
pr03	1010	1010	0.00	301.5	994	1.58	4360.00	1	3	5	1	0.78	2673.54	1608.92
pr04	1294	1286	0.62	442.7	1270	1.24	6944.85	0	1	6	3	1.08	3766.77	3009.60
pr05	1482	1481	0.07	650.3	1466	1.01	12385.11	0	6	4	0	2.19	6423.30	5640.76
pr06	1514	1501	0.86	651.2	1490	0.73	16168.27	2	4	4	0	1.62	9715.50	6142.95
pr07	744	738	0.81	260.4	743	-0.68	1063.48	5	0	4	1	0.34	552.56	488.94
pr08	1138	1139	-0.09	307.0	1114	2.19	3670.03	1	0	6	3	0.57	1996.78	1586.34
pr09	1275	1272	0.24	503.1	1226	3.62	7566.42	3	0	3	4	0.52	3752.48	3603.97
pr10	1573	1567	0.38	731.1	1565	0.13	15312.97	1	4	4	1	1.05	8417.89	6548.86
pr11	654	654	0.00	151.7	654	0.00	666.05	0	2	4	4	0.80	396.53	255.25
pr12	1002	997	0.50	294.3	970	2.71	2589.12	2	4	3	1	1.24	1325.88	1164.27
pr13	1139	1145	-0.53	378.9	1127	1.57	6333.54	2	1	5	2	0.87	3284.56	2895.06
pr14	1372	1315	4.15	533.7	1335	-1.52	13749.11	0	4	6	0	1.47	7664.66	5791.70
pr15	1650	1654	-0.24	708.1	1644	0.60	22320.37	1	3	5	1	1.78	13655.87	8121.46
pr16	1668	1609	3.54	818.1	1596	0.81	19583.44	1	6	2	1	2.01	9848.24	9162.52
pr17	838	841	-0.36	184.3	833	0.95	1896.28	2	1	5	2	0.46	1046.52	798.50
pr18	1281	1276	0.39	386.6	1258	1.41	6602.27	1	4	3	2	1.13	3517.87	2932.48
pr19	1417	1403	0.99	604.1	1395	0.57	15391.37	0	5	5	0	1.27	8496.68	6521.55
pr20	1684	1658	1.54	909.7	1687	-1.75	26942.93	1	2	5	2	1.38	14664.70	11709.73

A.1.7 Solomon $m = 4$

Table 14: Results Solomon instances for $m = 4$. The BKS are the best known solutions available in 2012 and used in Hu and Lim (2014).

Name	BKS	I3CH		$I_{max} = 3000$	GELD		$I_{max} = 10$	# Best solution chosen				Time spend in component		
		Profit	Gap BKS (%)	Time (s)	Profit	Gap I3CH (%)	Time (s)	# Current	# RR	# LS	# SA	RR (s)	LS (s)	SA (s)
c101	1020	1020	0.00	176.45	1020	0.00	603.25	2	0	8	0	0.48	309.16	274.35
c102	1150	1150	0.00	274.13	1150	0.00	1253.69	5	0	5	0	0.49	649.73	558.65
c103	1200	1210	-0.83	301.08	1210	0.00	1549.93	1	1	5	3	1.17	936.65	581.65
c104	1260	1260	0.00	281.83	1260	0.00	1673.73	5	0	4	1	0.41	946.64	691.46
c105	1070	1060	0.93	333.25	1060	0.00	757.81	4	0	6	0	0.79	385.62	348.61
c106	1080	1080	0.00	179.36	1080	0.00	718.97	5	0	4	1	0.41	356.06	338.63
c107	1120	1120	0.00	209.28	1120	0.00	1001.73	5	0	5	0	0.39	571.60	403.15
c108	1130	1130	0.00	373.13	1130	0.00	904.62	2	0	8	0	0.75	451.13	424.34
c109	1190	1190	0.00	227.58	1190	0.00	1224.97	2	0	6	2	0.44	702.94	488.69
r101	611	608	0.49	93.50	611	-0.49	379.36	6	2	1	1	0.29	188.44	176.34
r102	843	837	0.71	162.62	840	-0.36	1411.55	2	1	3	4	0.49	702.92	642.86
r103	926	928	-0.22	206.92	928	0.00	2293.74	4	0	2	4	0.34	1149.23	1096.64
r104	972	969	0.31	209.07	973	-0.41	2569.97	3	1	4	2	0.42	1498.20	1009.97
r105	778	778	0.00	142.40	776	0.26	914.38	0	1	6	3	0.29	467.68	422.77
r106	905	906	-0.11	185.92	906	0.00	1928.55	4	1	5	0	0.45	1047.55	839.85
r107	945	950	-0.53	221.36	950	0.00	2683.77	0	3	5	2	0.62	1472.36	1144.35
r108	994	994	0.00	220.04	993	0.10	2576.37	2	2	6	0	0.42	1321.19	1190.25
r109	885	885	0.00	145.77	885	0.00	1318.76	3	1	2	4	0.34	727.17	549.81
r110	914	915	-0.11	168.56	915	0.00	1727.66	3	1	4	2	0.28	925.54	752.88
r111	949	952	-0.32	203.47	946	0.63	1954.44	2	1	4	3	0.66	1025.32	878.90
r112	971	967	0.41	251.79	968	-0.10	2415.31	1	4	2	3	0.64	1292.54	1063.73
rc101	811	808	0.37	119.65	811	-0.37	773.77	2	1	4	3	0.23	418.96	335.59
rc102	908	899	0.99	136.72	895	0.44	1442.17	3	1	4	2	0.26	762.02	647.40
rc103	970	974	-0.41	170.16	974	0.00	1662.33	5	0	4	1	0.27	830.13	794.63
rc104	1059	1064	-0.47	194.13	1065	-0.09	1924.99	0	1	7	2	0.37	990.27	891.91
rc105	875	875	0.00	127.03	875	0.00	934.15	2	1	6	1	0.23	489.27	417.92
rc106	909	909	0.00	143.03	909	0.00	1203.36	6	0	3	1	0.25	613.53	562.24
rc107	980	980	0.00	155.35	982	-0.20	1601.63	3	0	5	2	0.35	849.89	716.30
rc108	1025	1020	0.49	173.18	1025	-0.49	1981.82	1	1	5	3	0.33	1039.19	900.85
c201	1810	1810	0.00	0.13	1810	0.00	2.40	0	0	0	0	0	0	0
c202	1810	1810	0.00	0.14	1810	0.00	3.85	0	0	0	0	0	0	0
c203	1810	1810	0.00	0.16	1810	0.00	3.52	0	0	0	0	0	0	0
c204	1810	1810	0.00	0.15	1810	0.00	1.15	0	0	0	0	0	0	0
c205	1810	1810	0.00	0.14	1810	0.00	1.50	0	0	0	0	0	0	0
c206	1810	1810	0.00	0.13	1810	0.00	1.27	0	0	0	0	0	0	0
c207	1810	1810	0.00	0.13	1810	0.00	1.10	0	0	0	0	0	0	0
c208	1810	1810	0.00	0.13	1810	0.00	1.03	0	0	0	0	0	0	0
r201	1458	1458	0.00	0.20	1458	0.00	2.01	0	0	0	0	0	0	0
r202	1458	1458	0.00	0.19	1458	0.00	5.08	0	0	0	0	0	0	0
r203	1458	1458	0.00	0.17	1458	0.00	0.91	0	0	0	0	0	0	0
r204	1458	1458	0.00	0.19	1458	0.00	0.66	0	0	0	0	0	0	0
r205	1458	1458	0.00	0.16	1458	0.00	0.69	0	0	0	0	0	0	0
r206	1458	1458	0.00	0.15	1458	0.00	0.71	0	0	0	0	0	0	0
r207	1458	1458	0.00	0.16	1458	0.00	0.60	0	0	0	0	0	0	0
r208	1458	1458	0.00	0.18	1458	0.00	0.70	0	0	0	0	0	0	0
r209	1458	1458	0.00	0.16	1458	0.00	0.98	0	0	0	0	0	0	0
r210	1458	1458	0.00	0.15	1458	0.00	1.00	0	0	0	0	0	0	0
r211	1458	1458	0.00	0.15	1458	0.00	0.83	0	0	0	0	0	0	0
rc201	1724	1724	0.00	0.23	1724	0.00	8.72	0	0	0	0	0	0	0
rc202	1724	1724	0.00	0.22	1724	0.00	1.24	0	0	0	0	0	0	0
rc203	1724	1724	0.00	0.19	1724	0.00	1.17	0	0	0	0	0	0	0
rc204	1724	1724	0.00	0.17	1724	0.00	1.06	0	0	0	0	0	0	0
rc205	1724	1724	0.00	0.21	1724	0.00	11.43	0	0	0	0	0	0	0
rc206	1724	1724	0.00	0.21	1724	0.00	1.41	0	0	0	0	0	0	0
rc207	1724	1724	0.00	0.22	1724	0.00	2.43	0	0	0	0	0	0	0
rc208	1724	1724	0.00	0.18	1724	0.00	1.58	0	0	0	0	0	0	0

A.1.8 Cordeau $m = 4$

Table 15: Results Cordeau instances for $m = 4$. The BKS are the best known solutions available in 2012 and used in Hu and Lim (2014).

Name	BKS	I3CH		$I_{max} = 3000$	GELD		$I_{max} = 10$	# Best solution chosen				Time spend in component		
		Profit	Gap BKS (%)	Time (s)	Profit	Gap I3CH (%)	Time (s)	# Current	# RR	# LS	# SA	RR (s)	LS (s)	SA (s)
pr01	657	657	0.00	0.1	657	0.00	64.12	0	0	1	0	0.44	29.16	17.22
pr02	1079	1073	0.56	380.6	1065	0.75	4780.98	2	1	5	2	1.03	2701.87	1944.87
pr03	1222	1232	-0.82	436.6	1231	0.08	8085.25	3	1	3	3	0.71	4552.82	3357.81
pr04	1557	1585	-1.80	603.6	1543	2.65	13189.83	2	6	2	0	1.72	7008.78	5799.13
pr05	1833	1838	-0.27	902.9	1800	2.07	27039.82	2	4	3	1	1.90	13911.46	12474.14
pr06	1860	1835	1.34	939.6	1859	-1.31	30462.11	1	3	4	2	1.62	16985.61	12756.09
pr07	876	872	0.46	228.9	876	-0.46	1964.31	2	2	2	4	0.63	1011.93	906.76
pr08	1382	1377	0.36	429.9	1355	1.60	7272.78	0	3	4	3	1.13	3973.00	3112.53
pr09	1619	1604	0.93	698.5	1608	-0.25	21468.43	2	2	3	3	1.09	12568.33	8441.65
pr10	1939	1943	-0.21	1044.4	1914	1.49	30511.16	2	5	2	1	1.78	16105.78	13648.37
pr11	657	657	0.00	0.1	657	0.00	1.77	0	0	0	0	0.00	0.00	0.00
pr12	1132	1120	1.06	477.1	1128	-0.71	5608.03	0	2	3	5	1.24	2882.73	2524.84
pr13	1364	1386	-1.61	672.0	1361	1.80	12379.71	3	3	3	1	1.30	6671.00	5389.49
pr14	1670	1651	1.14	783.2	1656	-0.30	25190.55	0	4	3	3	1.80	14180.57	10407.94
pr15	1958	2065	-5.46	1161.7	2047	0.87	42374.87	0	4	4	2	3.53	23677.59	17607.51
pr16	2065	2017	2.32	1183.8	2002	0.74	49793.99	1	6	1	2	2.31	26740.71	21850.40
pr17	933	934	-0.11	332.8	936	-0.21	3157.65	1	3	4	2	0.89	1741.59	1335.23
pr18	1525	1539	-0.92	559.5	1515	1.56	12553.04	1	5	2	2	1.79	6975.77	5287.85
pr19	1723	1750	-1.57	919.4	1714	2.06	35442.54	1	5	1	3	1.56	17968.43	16722.29
pr20	2037	2062	-1.23	1196.6	2069	-0.34	55929.40	0	4	4	2	2.25	30654.19	24122.22

A.1.9 New best known solutions

Table 16: Routes and profit scores of new BKS

r202 (m=2) 1349	Route 1	0, 92, 42, 14, 91, 45, 47, 36, 63, 64, 11, 19, 62, 88, 30, 71, 51, 9, 81, 79, 53, 82, 49, 48, 7, 10, 20, 32, 66, 35, 68, 12, 13, 89, 60, 17, 83, 5, 93, 85, 100, 58, 0
	Route 2	0, 1, 65, 34, 33, 27, 28, 26, 39, 23, 75, 15, 38, 44, 16, 61, 86, 99, 87, 2, 22, 41, 57, 37, 98, 59, 96, 94, 95, 97, 43, 56, 55, 54, 21, 72, 74, 4, 25, 24, 80, 77, 3, 50, 70, 31, 52, 0
r203 (m=2) 1422	Route 1	0, 18, 60, 83, 45, 5, 61, 98, 42, 57, 72, 39, 23, 15, 43, 38, 44, 85, 99, 96, 40, 53, 87, 2, 22, 41, 73, 21, 26, 12, 24, 55, 56, 75, 74, 4, 54, 68, 29, 34, 35, 51, 70, 1, 50, 3, 77, 80, 28, 58, 0
	Route 2	0, 20, 65, 71, 81, 33, 27, 69, 31, 64, 11, 62, 88, 30, 76, 67, 79, 78, 9, 49, 10, 66, 32, 90, 63, 19, 47, 48, 82, 7, 52, 89, 94, 13, 95, 59, 93, 16, 86, 14, 100, 37, 97, 0
r206 (m=2) 1450	Route 1	0, 42, 14, 45, 47, 36, 64, 63, 11, 19, 62, 88, 30, 51, 71, 9, 81, 78, 79, 76, 53, 40, 73, 22, 41, 87, 97, 43, 57, 2, 21, 72, 74, 75, 56, 4, 25, 55, 24, 54, 26, 58, 94, 96, 83, 60, 89, 52, 31, 70, 0
	Route 2	0, 50, 33, 65, 1, 69, 27, 28, 29, 39, 67, 23, 15, 44, 38, 86, 16, 61, 99, 6, 5, 84, 18, 8, 82, 7, 48, 49, 10, 32, 20, 66, 35, 34, 3, 77, 68, 80, 12, 13, 95, 59, 98, 37, 100, 91, 85, 93, 0
r210 (m=2) 1428	Route 1	0, 5, 45, 83, 52, 27, 21, 73, 72, 75, 23, 15, 42, 14, 44, 38, 86, 16, 61, 99, 87, 98, 85, 84, 8, 18, 82, 19, 49, 47, 48, 7, 10, 32, 20, 66, 35, 34, 3, 68, 24, 25, 54, 80, 77, 50, 1, 70, 0
	Route 2	0, 28, 12, 29, 33, 65, 63, 64, 11, 62, 88, 30, 76, 39, 67, 22, 40, 53, 79, 81, 9, 71, 51, 31, 89, 94, 95, 97, 43, 57, 2, 74, 56, 4, 26, 13, 96, 59, 93, 37, 100, 91, 17, 60, 58, 0
rc202 (m=2) 1515	Route 1	0, 82, 65, 91, 92, 95, 50, 33, 28, 26, 27, 29, 31, 34, 30, 71, 44, 40, 38, 41, 78, 79, 8, 6, 7, 46, 5, 3, 1, 43, 37, 35, 54, 96, 94, 89, 48, 21, 25, 77, 58, 83, 0
	Route 2	0, 36, 39, 42, 14, 47, 15, 11, 69, 64, 19, 23, 18, 76, 51, 22, 57, 86, 87, 9, 59, 75, 97, 10, 66, 56, 68, 55, 2, 4, 60, 17, 12, 98, 100, 70, 80, 0
rc201 (m=3) 1699	Route 1	0, 42, 36, 39, 5, 45, 2, 69, 98, 12, 15, 16, 11, 99, 88, 90, 7, 8, 79, 78, 6, 46, 3, 96, 54, 37, 35, 43, 1, 4, 60, 17, 100, 70, 0
	Route 2	0, 72, 92, 95, 63, 33, 28, 27, 29, 31, 30, 62, 67, 71, 61, 38, 40, 41, 81, 94, 84, 50, 34, 26, 32, 56, 55, 68, 93, 91, 80, 0
	Route 3	0, 65, 59, 47, 14, 82, 83, 64, 75, 23, 21, 19, 18, 76, 51, 22, 86, 87, 9, 57, 49, 20, 66, 10, 97, 13, 74, 24, 89, 48, 25, 77, 58, 0
pr20 (m=3) 1687	Route 1	0, 35, 205, 232, 115, 185, 243, 27, 72, 155, 88, 56, 101, 242, 208, 42, 34, 134, 20, 141, 275, 87, 94, 288, 23, 142, 178, 36, 154, 96, 254, 0
	Route 2	0, 9, 260, 160, 118, 179, 100, 108, 126, 86, 249, 180, 119, 226, 219, 114, 25, 97, 267, 221, 181, 109, 128, 158, 258, 2, 6, 16, 71, 110, 271, 0
	Route 3	0, 51, 161, 146, 246, 206, 14, 54, 65, 184, 112, 144, 173, 261, 166, 129, 277, 17, 233, 263, 244, 98, 62, 80, 64, 50, 229, 138, 103, 123, 77, 48, 177, 190, 143, 207, 0
rc104 (m=4) 1065	Route 1	0, 80, 94, 96, 54, 41, 38, 37, 35, 36, 40, 43, 42, 61, 0
	Route 2	0, 98, 88, 60, 78, 79, 7, 6, 5, 4, 2, 70, 68, 0
	Route 3	0, 92, 56, 51, 76, 89, 18, 19, 49, 22, 64, 66, 0
	Route 4	0, 12, 16, 15, 11, 10, 9, 97, 59, 86, 57, 65, 0
pr17 (m=4) 936	Route 1	0, 72, 5, 63, 28, 46, 53, 15, 16, 56, 54, 22, 32, 44, 8, 34, 62, 45, 3, 0
	Route 2	0, 33, 11, 70, 27, 31, 20, 36, 71, 35, 39, 38, 2, 18, 65, 10, 0
	Route 3	0, 43, 7, 51, 59, 13, 52, 29, 4, 55, 21, 48, 14, 67, 60, 9, 19, 42, 30, 66, 69, 25, 0
	Route 4	0, 23, 1, 50, 41, 17, 12, 47, 64, 26, 68, 49, 61, 37, 58, 0
pr20 (m=4) 2069	Route 1	0, 46, 232, 205, 249, 35, 86, 126, 180, 119, 226, 114, 25, 159, 172, 133, 12, 73, 128, 152, 158, 258, 2, 6, 16, 71, 110, 271, 0
	Route 2	0, 206, 14, 246, 83, 146, 288, 13, 94, 275, 141, 20, 134, 34, 42, 185, 115, 109, 221, 267, 97, 219, 284, 228, 168, 251, 177, 190, 143, 207, 0
	Route 3	0, 170, 208, 155, 44, 243, 27, 72, 242, 7, 56, 101, 150, 105, 89, 174, 285, 245, 11, 39, 254, 36, 142, 178, 181, 0
	Route 4	0, 161, 51, 108, 100, 179, 118, 160, 112, 9, 260, 144, 173, 261, 192, 166, 129, 277, 17, 233, 263, 122, 244, 98, 62, 54, 123, 65, 184, 77, 103, 138, 229, 50, 52, 154, 96, 0

A.2 Results without route recombination

Table 17: Results without the RR component

m	Name	Geld with RR $I_{max} = 10$		Geld without RR $I_{max} = 10$			# Best solution chosen				Time spend in component		
		Profit	Time (s)	Profit	Gap I3CH (%)	Time (s)	# Current	# RR	# LS	# SA	RR (s)	LS (s)	SA (s)
$m = 4$	r112	968	2415.31	958	1.03	1973.088	2	0	6	2	0.00	980.38	919.98
$m = 4$	r107	982	1601.63	945	3.77	1932.142	4	0	3	3	0.00	956.40	924.28
$m = 2$	pr13	836	2655.43	832	0.48	2513.812	2	0	7	1	0.00	1360.33	1100.40
$m = 3$	r105	618	404.04	615	0.49	428.776	2	0	3	5	0.00	213.04	205.78
$m = 1$	r204	1093	2194.2	1091	0.18	1768.361	3	0	4	3	0.00	921.64	814.81
$m = 2$	r108	556	388.86	555	0.18	367.93	1	0	2	7	0.00	197.57	148.23
$m = 3$	rc107	768	673.03	756	1.56	607.151	2	0	6	2	0.00	312.58	280.74
$m = 4$	r101	611	379.36	605	0.98	377.815	4	0	4	2	0.00	187.39	178.36
$m = 1$	pr18	531	337.91	519	2.26	404.106	3	0	4	3	0.00	211.36	180.60
$m = 2$	pr03	732	918.16	726	0.82	887.031	6	0	3	1	0.00	450.23	406.18

A.3 Results from initialization with different number of solutions

Table 18: Results for multiple instances when different number of solutions are created in the initialization procedure

Instance	# Solutions	Profit after		Time(s)
		Init	GELD	
rc102	$1N$	466	505	86.68
$m = 2$	$2N$	477	505	139.19
	$3N$	490	505	55.19
	$4N$	490	505	84.51
	$5N$	490	505	127.56
r101	$1N$	581	611	248.34
$m = 4$	$2N$	587	611	108.2
	$3N$	597	611	82.96
	$4N$	597	611	85.07
	$5N$	597	611	89.82
pr20	$1N$	573	644	1397.05
$m = 1$	$2N$	573	644	1538.32
	$3N$	573	647	1691.22
	$4N$	583	624	1566.77
	$5N$	583	624	1377.35
c103	$1N$	1180	1210	605.53
$m = 4$	$2N$	1180	1210	793.00
	$3N$	1180	1210	498.97
	$4N$	1180	1210	581.57
	$5N$	1180	1210	883.46
c105	$1N$	1040	1060	99.44
$m = 4$	$2N$	1040	1060	141.80
	$3N$	1040	1060	139.09
	$4N$	1040	1060	140.46
	$5N$	1040	1060	135.83
rc208	$1N$	992	1038	651.64
$m = 1$	$2N$	992	1036	1176.25
	$3N$	992	1038	380.41
	$4N$	992	1035	1169.22
	$5N$	992	1038	195.84
c101	$1N$	590	590	11.88
$m = 2$	$2N$	590	590	13.12
	$3N$	590	590	14.18
	$4N$	590	590	15.21
	$5N$	590	590	16.41
rc201	$1N$	709	789	303.88
$m = 1$	$2N$	730	795	257.01
	$3N$	750	795	119.78
	$4N$	750	795	107.23
	$5N$	750	795	137.05
c107	$1N$	880	910	82.15
$m = 3$	$2N$	890	910	49.12
	$3N$	890	910	64.07
	$4N$	890	910	59.50
	$5N$	890	910	64.64
c206	$1N$	920	930	61.91
$m = 1$	$2N$	920	930	71.42
	$3N$	920	930	66.10
	$4N$	920	930	44.33
	$5N$	920	930	95.70

A.4 Results of TOPTWST

Table 19: Results TOPTWST for $m = 4$. The BKS are the best known solutions available in 2012 and used in Hu and Lim (2014).

Name	BKS	I3CH	GELD	$I_{max} = 10$	GELD TOPTWST	$I_{max} = 10$	# Best solution chosen				Time spend in component		
		Profit	Profit	Time (s)	Profit	Time (s)	# Current	# RR	# LS	# SA	RR (s)	LS (s)	SA (s)
rc101	811	808	811	773.77	749	263.17	3	2	3	2	0.46	131.42	110.50
rc102	908	899	895	1442.17	791	327.74	5	0	4	1	0.28	161.15	152.90
rc103	970	974	974	1662.33	901	469.71	1	3	3	3	0.41	263.82	193.21
rc104	1059	1064	1065	1924.99	1045	585.42	1	4	5	0	0.63	314.56	256.52
rc105	875	875	875	934.15	780	251.53	4	1	3	2	0.19	124.91	117.30
rc106	909	909	909	1203.36	786	395.91	2	1	4	3	0.23	205.95	180.30
rc107	980	980	982	1601.63	888	492.06	2	2	3	3	0.36	253.04	226.91
rc108	1025	1020	1025	1981.82	933	581.98	0	4	2	4	0.35	307.97	259.89

Name	Profit										
	Init	1	2	3	4	5	6	7	8	9	10
rc101	644	701	729	749	749	749	749	749	749	749	749
rc102	762	791	791	791	791	791	791	791	791	791	791
rc103	796	838	854	878	901	901	901	901	901	901	901
rc104	861	923	926	985	1014	1014	1014	1017	1045	1045	1045
rc105	727	757	780	780	780	780	780	780	780	780	780
rc106	742	778	780	786	786	786	786	786	786	786	786
rc107	829	858	883	883	883	888	888	888	888	888	888
rc108	829	900	907	928	932	933	933	933	933	933	933

Name	BKS	I3CH	GELD	$I_{max} = 10$	GELD TOPTWST	$I_{max} = 10$	# Best solution chosen				Time spend in component		
		Profit	Profit	Time (s)	Profit	Time (s)	# Current	# RR	# LS	# SA	RR (s)	LS (s)	SA (s)
rc101	811	808	811	773.77	806	203.728	2	0	4	4	0.30	104.93	91.01
rc102	908	899	895	1442.17	893	551.786	3	1	5	1	0.47	312.09	225.73
rc103	970	974	974	1662.33	905	584.229	1	0	6	3	0.32	307.12	263.82
rc104	1059	1064	1065	1924.99	991	921.04	2	1	3	4	0.48	553.90	352.70
rc105	875	875	875	934.15	842	340.125	4	1	4	1	0.28	180.48	150.88
rc106	909	909	909	1203.36	876	331.658	1	0	7	2	0.25	176.44	146.42
rc107	980	980	982	1601.63	924	475.218	1	0	7	2	0.26	245.13	218.88
rc108	1025	1020	1025	1981.82	1012	608.799	1	1	7	1	0.40	330.55	265.56

Name	Profit										
	Init	1	2	3	4	5	6	7	8	9	10
rc101	640	806	806	806	806	806	806	806	806	806	806
rc102	753	874	874	874	876	885	893	893	893	893	893
rc103	811	905	905	905	905	905	905	905	905	905	905
rc104	867	958	985	991	991	991	991	991	991	991	991
rc105	714	811	811	811	81	826	842	842	842	842	842
rc106	726	874	875	875	875	875	875	875	876	876	876
rc107	810	924	924	924	924	924	924	924	924	924	924
rc108	824	929	944	1012	1012	1012	1012	1012	1012	1012	1012

Name	BKS	I3CH	GELD	$I_{max} = 10$	GELD TOPTWST	$I_{max} = 10$	# Best solution chosen				Time spend in component		
		Profit	Profit	Time (s)	Profit	Time (s)	# Current	# RR	# LS	# SA	RR (s)	LS (s)	SA (s)
rc101	811	808	811	773.77	808	245.12	5	1	3	1	0.23	126.78	112.47
rc102	908	899	895	1442.17	880	449.226	1	2	5	2	0.45	252.85	186.89
rc103	970	974	974	1662.33	907	643.31	1	2	5	2	0.42	378.14	254.17
rc104	1059	1064	1065	1924.99	967	719.445	2	0	5	3	0.31	398.13	308.85
rc105	875	875	875	934.15	860	297.523	2	1	6	1	0.30	171.18	118.14
rc106	909	909	909	1203.36	892	453.432	2	1	5	2	0.38	258.06	186.48
rc107	980	980	982	1601.63	961	534.857	2	2	4	2	0.40	312.58	210.94
rc108	1025	1020	1025	1981.82	955	511.789	3	1	5	1	0.56	281.10	217.69

Name	Profit										
	Init	1	2	3	4	5	6	7	8	9	10
rc101	628	777	777	783	808	808	808	808	808	808	808
rc102	717	808	821	838	846	880	880	880	880	880	880
rc103	788	902	906	907	907	907	907	907	907	907	907
rc104	825	940	857	967	967	967	967	967	967	967	967
rc105	669	828	832	860	860	860	860	860	860	860	860
rc106	718	846	861	861	883	883	885	885	885	892	892
rc107	803	896	904	913	951	961	961	961	961	961	961
rc108	795	890	908	913	955	955	955	955	955	955	955

Table 20: Results TOPTWST for $m = 3$. The BKS are the best known solutions available in 2012 and used in Hu and Lim (2014).

Name	BKS	I3CH	GELD	$I_{max} = 10$	GELD TOPTWST	$I_{max} = 10$	# Best solution chosen				Time spend in component		
		Profit	Profit	Time (s)	Profit	Time (s)	# Current	# RR	# LS	# SA	RR (s)	LS (s)	SA (s)
rc101	621	621	621	338.80	586	176.58	1	1	6	2	0.36	94.11	71.85
rc102	714	714	710	643.77	662	273.45	2	2	4	2	0.33	140.94	123.90
rc103	764	764	764	663.32	716	222.69	0	2	6	2	0.28	111.25	104.25
rc104	833	834	834	965.16	692	395.19	2	1	3	4	0.21	206.72	177.47
rc105	682	682	682	507.14	599	194.70	3	2	4	1	0.24	99.79	89.51
rc106	706	706	706	423.52	581	199.22	4	1	4	1	0.19	97.40	96.59
rc107	773	762	768	673.03	650	272.67	3	1	3	3	0.20	138.08	127.68
rc108	795	789	795	685.99	657	330.27	2	1	4	3	0.29	169.89	152.70

Name	Profit										
	Init	1	2	3	4	5	6	7	8	9	10
rc101	492	525	586	586	586	586	586	586	586	586	586
rc102	580	592	644	654	662	662	662	662	662	662	662
rc103	601	628	698	707	716	716	716	716	716	716	716
rc104	663	684	692	692	692	692	692	692	692	692	692
rc105	549	572	597	599	599	599	599	599	599	599	599
rc106	537	570	581	581	581	581	581	581	581	581	581
rc107	625	650	650	650	650	650	650	650	650	650	650
rc108	633	656	657	657	657	657	657	657	657	657	657

Name	BKS	I3CH	GELD	$I_{max} = 10$	GELD TOPTWST	$I_{max} = 10$	# Best solution chosen				Time spend in component		
		Profit	Profit	Time (s)	Profit	Time (s)	# Current	# RR	# LS	# SA	RR (s)	LS (s)	SA (s)
rc101	621	621	621	338.80	600	150.235	2	1	6	1	0.42	81.45	58.04
rc102	714	714	710	643.77	679	296.188	1	2	7	0	0.37	165.87	120.02
rc103	764	764	764	663.32	723	318.902	1	3	5	1	0.38	180.83	132.14
rc104	833	834	834	965.16	812	346.702	1	1	5	3	0.29	188.72	149.16
rc105	682	682	682	507.14	633	142.504	2	1	6	0	0.26	66.32	71.65
rc106	706	706	706	423.52	659	199.12	1	2	5	2	0.32	100.88	92.52
rc107	773	762	768	673.03	767	348.522	2	1	3	4	0.31	202.48	140.08
rc108	795	789	795	685.99	730	314.752	0	3	6	1	0.32	160.63	147.85

Name	Profit										
	Init	1	2	3	4	5	6	7	8	9	10
rc101	505	588	588	588	588	588	593	593	600	600	600
rc102	556	600	659	973	973	973	973	675	679	679	679
rc103	612	686	696	715	718	720	723	723	723	723	723
rc104	653	788	788	788	788	788	788	795	812	812	812
rc105	561	621	633	633	633	633	633	633	633	633	633
rc106	554	648	50	650	659	659	659	659	659	659	659
rc107	620	756	756	756	756	756	757	758	767	767	767
rc108	620	708	708	709	715	715	716	725	725	725	730

Name	BKS	I3CH	GELD	$I_{max} = 10$	GELD TOPTWST	$I_{max} = 10$	# Best solution chosen				Time spend in component		
		Profit	Profit	Time (s)	Profit	Time (s)	# Current	# RR	# LS	# SA	RR (s)	LS (s)	SA (s)
rc101	621	621	621	338.80	606	131.834	3	0	3	4	0.18	65.98	62.54
rc102	714	714	710	643.77	710	253.806	1	2	5	2	0.51	139.11	109.51
rc103	764	764	764	663.32	676	248.932	3	1	4	2	0.30	123.76	112.70
rc104	833	834	834	965.16	757	246.839	3	1	3	3	0.25	140.39	99.85
rc105	682	682	682	507.14	649	180.338	2	3	4	1	0.26	91.90	83.68
rc106	706	706	706	423.52	647	223.799	1	0	6	3	0.25	115.47	103.73
rc107	773	762	768	673.03	706	238.274	2	0	3	5	0.24	122.24	110.63
rc108	795	789	795	685.99	775	322.023	1	2	6	1	0.46	162.81	152.86

Name	Profit										
	Init	1	2	3	4	5	6	7	8	9	10
rc101	495	591	591	606	606	606	606	606	606	606	606
rc102	567	647	649	652	655	680	707	707	707	707	707
rc103	599	659	670	676	676	676	676	676	676	676	676
rc104	671	744	744	744	744	754	754	754	754	754	757
rc105	557	621	628	635	635	641	641	649	649	649	649
rc106	558	644	646	647	647	647	647	647	647	647	647
rc107	613	705	705	705	705	705	706	706	706	706	706
rc108	654	692	724	753	768	775	775	775	775	775	775