

BSc Thesis Applied Mathematics

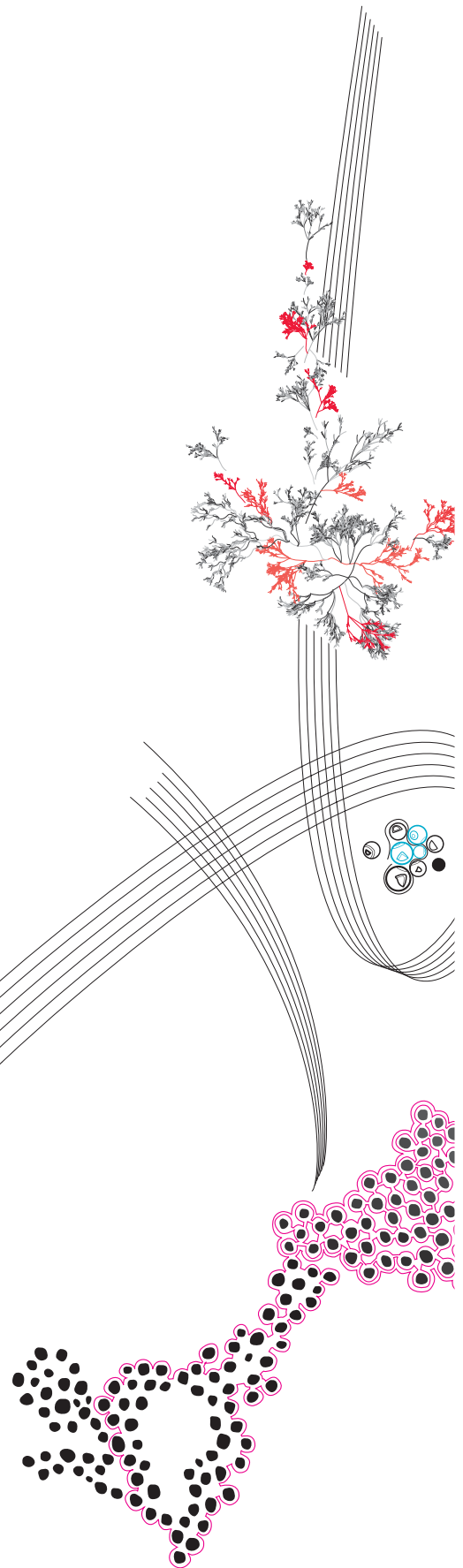
Online route planning in response to traffic incidents

Yoran de Weert

Supervisor: MSc. O. Eikenbroek
Dr. W. Kern

January, 2019

Department of Applied Mathematics
Faculty of Electrical Engineering,
Mathematics and Computer Science



Preface

First of all, I would like to thank to thank my supervisors for all support and feedback throughout the process. Many thanks to O. Eikenbroek (supervisor) for his time, excellent guidance and valuable input and to W. Kern (supervisor) for the time and feedback. Finally, I would like to thank J. Timmer for finding me this project and together with O. Eikenbroek, making this whole project possible.

Enschede, 25 January 2019

Online route planning in response to traffic incidents

Yoran de Weert*

January, 2019

Abstract

Non-recurrent disturbances substantially impact the timeliness of home deliveries. Where recurrent disruptions are predictable and can therefore be incorporated in offline route plans, the en-route adaptation of route plans is required to reduce the impact of non-recurrent disturbances. In this paper, an online re-planning method is developed and evaluated in order to reduce the impact of non-recurrent traffic disturbance. The method is based on two strategies: inter-route switching of customers and the transfer of customers to other vehicles. Results are obtained using an evaluation method that schedules random recurrent congestion and non-recurrent disturbances and compared to an offline solution method.

Keywords: Vehicle routing, time windows, heuristics, non-recurrent disturbances, time dependency.

1 Introduction

In home deliveries, a fleet of vehicles has to serve a number of customers with their products. Each vehicle is assigned to a certain number of customers so that all customers will be served. In addition, sometimes the customers can set a time window in which they would like to be served. The problem here is that there are many possibilities to assign these customers to vehicles and a Vehicle Routing Problem with Time Windows (VRPTW) deals with that problem. Define a route as a sequence of paths that connects a set of customers and a path as a connection between two customers. The goal of a VRPTW is to find an optimal set of routes that minimizes an objective function. An objective could be minimizing the route costs such as distance or the number of used vehicles, but also factors as service quality can be considered.

The VRPTW is typically solved in an offline setting, i.e., the solution is constructed before execution and not adjusted while the vehicles are en route. Sometimes the disturbances are incorporated in the offline solution in order to improve such an offline solution. Two types of disturbances are here distinguished: Recurrent disturbances and non-recurrent disturbances. The recurrent disturbances can typically be predicted and it is possible for offline solution methods of the vehicle routing problem to take this kind of disturbances into account. On the other hand, non-recurrent disturbances in traffic are difficult to predict and therefore almost impossible to be incorporated in an offline solution method. This gives a chance that there is some delay on the routes, which may cause a violation in the time window constraint and that lowers the service quality.

Nowadays, reducing the impact of that non-recurrent disturbances on an offline solution are based on a combination of human experience and real-time monitoring systems, but

*Email: y.r.deweert@student.utwente.nl

those solutions are generally not optimal. To find better improvements in the routes of the vehicles, online re-planning methodologies are necessary. An online re-planning method re-plans the routes after the delivery vehicles are en route. The strategies for such an online re-planning methodology are inter-route switching of trips, inter-route switching of customers, and inter-route helper actions. In this case, the last two strategies will be considered. That is, changing the order of the customers that have to be served in a route and the last strategy is based on transferring customers to other vehicles.

The goal of this paper is to develop and evaluate such an online re-planning methodology to improve the online re-planning in case of non-recurrent traffic disturbances. Developing an re-planning method is to my knowledge not done before and the question that arises is if we can develop such a re-planning method that considers the two strategies mentioned above. These strategies have to be tested and see under what conditions the strategies can achieve good results. Since the routes have to be adjusted during the process, research is also needed of how often and when to update routes.

Evaluating the re-planning method is also not easily done, because of the novelty of this subject. In most cases a test instance is used to evaluate the results. A test instance is test data for a certain problem that can be used to measure or verify the achievements of a solution method for that problem. However, for as far as the knowledge of the author goes, there is no instance available that involves recurrent and non-recurrent disturbance.

This paper is organized as follows. First a literature review will be given about the solution methods in section 2 and after that a detailed description of the problem in section 3. The section 4 describes the used and developed algorithms and the results of the method will be presented in section 5 and discussed in section 6. A conclusion and future directions are given in the last section 7.

2 Literature review

Solving the VRPTW can be done using different solutions methods. These methods can be divided into exact solution methods and heuristics. The exact solution methods give optimal solutions for problems, but requires a lot of time. A less time-consuming approach is the use of heuristics, which are methods that generate solutions that are sufficient for reaching a certain goal.

Cordeau et al. (2007) [3] have made a summary of heuristics for the VRPTW. They distinguish three types of heuristics: Construction-, improvement- and meta-heuristics. The first heuristic constructs routes, the second type tries to improve the set of routes and the last type also tries to improve the set of routes, but after a local minimum is encountered it continues exploring the search space.

The focus in this paper lies on the meta-heuristic tabu search. This method is are often used in Vehicle Routing Problems and achieve good results. Tabu search is a meta-heuristic for guiding search to obtain good solutions in the solution space [6]. It needs an initial solution, then generates a neighbourhood around that current solution and the algorithm searches for an improvement in that neighbourhood. A neighbourhood can be defined as a set of small adaptations with respect to the current solution and is generated by applying moves to the current solution, for example a switch move; it switches in this case two customers and then it checks whether the solution is improved or not. The new solution generated by the switch move is said to be in the neighbourhood of the current solution. The tabu search heuristic often uses a tabu list, a list with moves that are not allowed during the search. This prevents cycling and guides the search into other possible direc-

tions. The heuristic keeps trying to improve solutions until a certain condition is met, for example if the optimal solution is found or if the maximum number of iterations is reached, and then stops.

Bräysy and Gendreau (2002) [1] have made an overview of tabu search heuristics for the VRPTW. An example is the research done by Taillard, Badeau et al. (1997) [14]. They propose a tabu search method based on edge exchange heuristics, called the CROSS exchange. The idea of that exchange is not to switch single customers as in a switch move, but the CROSS exchange switches segments of routes. It uses an adaptive memory to store the best previous visited solutions. The advantage is that good combinations of customers in a route can be maintained, while a single switch move could break that good combination.

A more general tabu search approach is developed by Higgins (1999) [7] and can be used for more optimization problems than only the VRPTW. It uses standard insertion moves and swap moves to generate neighbourhoods and uses a dynamic tabu list to faster generate better solutions. The method also accepts non-improving moves, but returns to its best solution if no better solution is found after doing a certain number of iterations with non-improving moves.

More important for us is a method for solving the dynamic case of the VRPTW (DVRPTW). The difference with the normal VRPTW is that the circumstances change while the delivery vehicles are en route. More concretely, during the process there are sometimes new service request and the set of routes has to be changed in order to serve this customers. Pillac et al. (2013) [11] have made a review of different solution methods for the dynamic vehicle routing problem. For the online deterministic case, they have distinguished two types of solution methods: periodic re-optimization and continuous re-optimization. Periodic re-optimization starts with an initial set of routes at the beginning of the day. After the data changes or after a fixed time interval, the routes will be renewed. This can be accomplished by solving a static problem at the time of updating. The advantage here is the fact that there is a lot of research been conducted about solving the static problem. On the other hand, running an algorithm takes time and a problem could be that traffic conditions could be drastically changed compared to the situation at the starting moment of the algorithm once the algorithm is ended.

Chen and Xu (2006) [2] have developed a periodic re-optimization method for a DVRPTW. Their problem is not the traffic condition, but in their case there are new service requests during the day that have to be scheduled. The approach of their paper is column-generation-based and at each *decision epoch*, which are the update times of the route, a static problem is solved using the column-generation-based algorithm.

The continuous re-optimization uses the interval between events to optimize the routes. It stores information of good solutions in an adaptive memory. When an event occurs, a decision procedure is used to update the current routes using the adaptive memory. The advantage is that the computational capacity is maximized, since the optimization process continues throughout the day. Since the routes could change at any time, it is not known what the next customer of a vehicle is and that could be a disadvantage. Gendreau et al. (1999) [4] tries to solve the same problem as Chen and Xu (2006), but uses a continuous re-optimization approach. Their main idea is to store information of good routes in the adaptive memory. It then uses a parallel tabu search, which partitions all routes of the current solution and optimizes each partition individually. When a new service request arrives, it is checked using the adaptive memory whether the service request should be accepted or rejected.

Evaluating the performance of developed algorithms can be accomplished by using a benchmark. For the VRPTW, the most used benchmark is the one developed by Solomon (1987) [13]. The benchmark exists of a set of test instances and each test instance contains a depot and 100 customers of which each customer has a location, a time window in which the customer has to be served, a demand quantity and a service time. The depot has also a time window, it opens at $t = 0$ and before the time window of the depot closes the vehicles have to be back at the depot. Between each pair of customers, there is a path with its path length equal to the euclidean distance. A solution method for the VRPTW that has the objective to 1) minimize the number of vehicles and 2) minimize the total distance can be evaluated on this instance.

Gendreau et al. (1999) have adjusted the Solomon instances in order to evaluate the parallel tabu search in that paper, which is also described above. It uses the same data set, but divides the set in two subsets (each subset with 50 customers). The first subset is the set with customers that are known at the start of day and the other subset contains customers that send a service request during the day. There are scenarios of type 1 and type 2, which actually only differ in the average number service request per minute. A scenario of type 1 has approximately three requests per minute and type 2 approximately one request per minute.

3 Problem description

The VRPTW can be defined as follows: Let $G = (V, E)$ be a graph where $V = \{v_0, v_1, \dots, v_c\}$ is the vertex set and $E = \{(v_i, v_j) : v_i, v_j \in V\}$ the edge set. The vertex set V represents the depot, v_0 , and the customers, v_1, v_2, \dots, v_c . For all $1 \leq i \leq c$ it holds that a customer i is associated with a certain (x_i, y_i) location, a time window $[b_i, e_i]$ in which that customer wants to be served, demand quantity q_i and a service time s_i . An edge between v_i and v_j represents the path between those two customers and each path is associated with a traveltime t_{ij} , the time it takes to get from customer i to j . Note that the recurrent events can be predicted and that the impact of these disturbances can be considered in an offline solution method. The impact changes the traveltime and since these disturbances are not constant throughout a day, the traveltime is dependent of the time.

A fleet of at most n vehicles has to serve these c customers. All customers have to be visited exactly once by one of the vehicles and this has to be done within the time window of each customer. The vehicles do have a maximum capacity Q , so the demand of the customers cannot exceed Q . The vehicles start at the depot and also have to return to the depot before a certain time e_0 . If a vehicle arrives at a customer before the corresponding time window starts, the vehicle has to wait until the time window opens before the service can begin and if it is too late, a lateness penalty is incurred. The objective is to find a set of routes r such that $f(r) = \sum_{i=1}^n T_i + \alpha \sum_{m=1}^c z_m + \beta \sum_{k=1}^n (Q - \sum_{l \in C_k} q_l) + \gamma \sum_{j=1}^c (a_j - e_j)^+$ is minimized, where T_i is the total traveltime of vehicle i , a_j is the arrival time of a vehicle at customer j , C_k is the set of customers that vehicle k has to serve, $(x)^+ = \max(0, x)$ and z_i is 0 if the customer is served on time and 1 otherwise.

4 Online re-planning

In this section, the problem-solving method will be introduced. The idea is to develop a periodic re-optimization method that updates the routes after a certain time period. The

structure of the method is as follows. First an offline solution has to be constructed. After each time period, an improvement algorithm will be applied to the current set of routes. This algorithm exists of two parts which are based on the strategies mentioned in the introduction, the inter-route switching of customers and inter-route helper actions. The method also respects the FIFO property, which is explained below.

4.1 FIFO property

In all cases that the traveltime from customer i to customer j has to be determined, the solution method respects the FIFO property. Ichoua et al. (2003) [10] has defined the FIFO property as follows: the FIFO property guarantees that if a vehicle leaves a customer i for a node j at a given time, any identical vehicle leaving customer i for customer j at a later time will arrive later at customer j . It is possible to violate the property by not checking if traffic conditions change on the path that the vehicle took. For example, the expected traveltime at $t = 10$ is 5 minutes, because of an traffic incident. At $t = 12$, the traffic incident is solved and now the traveltime becomes 1 minute again. Two vehicles took that same path, one of the vehicles drove away at $t = 10$ and the other one at $t = 12$. The first vehicle arrived at $t = 15$, but the vehicle that left later arrived at $t = 13$. This is possible since future conditions that are already known are not checked in a method. In our method, the FIFO property is respected. First, the arrival time at customer j is determined for the earliest departure time at customer i . Then, for each time unit later, it is determined what the arrival time at customer j will be, until the departure time is equal to the arrival time corresponding to the earliest departure time at customer i .

4.2 Constructing initial routes

The offline solution will be based on the tabu search heuristic developed by Higgins [7]. First, an initial solution has to be found for the routing problem and then the tabu search can be used to improve the initial solution. The result of the tabu search is an offline solution that will be used and adjusted during the process of serving customers. The initial solution for the tabu search heuristic is constructed using an algorithm of Ho and Haugland [8], algorithm 1 gives a summary of the construction technique. It starts with a list of unrouted customers U and one vehicle. At each iteration it searches for the nearest unrouted neighbour that can be served the earliest and sets that customer as next customer in the route of a vehicle. A customer only will be inserted if it keeps the route feasible. Feasibility is achieved if the solution respects the capacity and time window, so no capacity overflow or customers that are served outside their time window. If there are no feasible options anymore, the algorithm will add another vehicle and starts again with remaining unrouted customers.

Define \hat{j} as the nearest unrouted neighbour and i as the latest routed customer. The nearest neighbour is based on the customer where the vehicle can start its service as quickest, after the vehicle served customer i :

$$\hat{j} = \min_{j \in U} \{ \max \{ b_j, a_i + s_i + t_{ij}(t) \} \} \quad (1)$$

where s_i is the service time and a_i the arrival time of the vehicle at customer i . $t_{ij}(t)$ is the traveltime between customer i and j and is time-dependent because of the impact of recurrent disturbances. For initializing the route, set $k = 1$ and R_k as the current route, the vehicle has to start at the depot and end at the depot which gives an initial route $R_k = (v_0, v_0)$.

The algorithm described in the paper stops if no feasible insertions can be found or if the

limit of n vehicles is reached, so it is possible that there are still some remaining unrouted customers. In this case, all customers have to be served and therefore the remaining unrouted customers are inserted in a route at the place where the costs of constraint violation is minimized.

Algorithm 1

Step 1 Set $k = 1$.

Step 2 Begin with an empty route k starting from the depot, $R_k = (v_0, v_0)$. Set $\hat{j} = 0$.

Step 3 Set $i = \hat{j}$. Find the nearest unrouted neighbour \hat{j} to customer i that is also feasible in terms of time window and capacity constraints. If there is no feasible unrouted neighbour \hat{j} , set $k = k + 1$ and return to step 2.

Step 4 Insert \hat{j} in route R_k and remove \hat{j} from U . If U is not empty, return to step 3.

Step 5 If the limit of n vehicles is reached or no feasible insertions can be found, insert each customer in a route and place where the costs of constraint violation is minimized.

4.3 The tabu search

To improve the initial solution, the tabu search of Higgins (2001) [7] will be used as a basis. The meta-heuristic starts by generating a sample of the neighbourhood. Using insertion- and swap-moves and applying that to the initial solution or current solution, a neighbourhood can be defined. An insertion move transfers a customer from one vehicle to another vehicle and a swap move switches two customers from different routes with each other. A move is tabu if it is in the tabu list, which is in this case filled during the search with the most recent moves.

The tabu search is initialized by defining the best solution (BS) as the current solution (CS), which is in this case the initial solution found by the algorithm described in section 4.2. It then generates a sample of a neighbourhood and based on the objective function, the tabu search searches for the best solution in that sample. That solution will then be defined as the current solution CS.

After that, the feasibility of the current solution will be determined. If CS is feasible and it is better than the best solution found overall (BS), the best solution will be replaced by CS. If the current solution does not improve BS after a predefined number of iterations, $\phi^\#$, then the current solution will be replaced by the best solution found so far.

It is also possible to have an infeasible solution as CS. This is allowed, because the tabu search is in that way able to move away from a local minimum and explore other minima. If after a predefined number of iterations $\Phi^\#$ the CS is still infeasible, the method then defines the best feasible solution found as the current solution and the algorithm will continue. If no feasible solution has been found yet, the algorithm will return to its initial solution. The heuristic ends when the runtime of the method reaches a predefined amount of time RT. Algorithm 2 summarizes the tabu search.

Algorithm 2

```
Set BS := CS
Repeat
  Set  $\phi := 0$ 
  Set  $\Phi := 0$ 
  Repeat
    Obtain a sample of moves from the neighbourhood of CS
    Set CS := the solution given the best non tabu move in the sample
    If  $f(\text{CS}) < f(\text{BS})$  and CS is feasible then
      Set BS := CS and CFS := CS (Current best feasible solution)
      Set  $\phi := 0$ 
    ElseIf  $f(\text{CS}) \geq f(\text{BS})$  and CS feasible then
      Set  $\phi := \phi + 1$ 
    End If
    If CS is infeasible then
      Set  $\Phi := \Phi + 1$ 
    Else
      Set  $\Phi := 0$ 
    End if
  Until  $\phi := \phi^\#$  or  $\Phi := \Phi^\#$  or  $f(\text{CFS}) < f(\text{CS})$ 
  Set CS = BS and CFS = BS
Until CPU Time reaches RT
```

4.4 Online improvements

As mentioned, two strategies are considered to re-plan the routes of vehicles. The first strategy, the inter-route switching of customers, considers individual vehicle routes and tries to improve these individual routes. If this strategy is applied to such a route and the solution is still not feasible, the second strategy will be applied to find out whether it is possible to obtain a feasible solution or reduce the infeasibility by switching customers between vehicles. If a route is already feasible, this second strategy will not be applied to improve that route. This also improves the speed of the algorithm, since only one strategy is applied instead of two.

4.4.1 Order-changing strategy

This strategy considers each individual route and the goal is to find the shortest possible route with respect to the constraints, so it can be seen as a variant of the Traveling Salesman Problem with Time Windows. The vehicle is at a certain location and it has to serve all customers that are still on the route of that vehicle. The only factor that is changed are the traffic circumstances and because of that, time windows may be violated and cause an infeasible route. The capacity constraint cannot be violated, since the vehicle is already loaded at the beginning of the day and there are not any additional customers to be scheduled in that route.

The technique is based on an insertion heuristic proposed by Gendreau et al. (1998) [5]. Virtually, it picks the starting point of the route and removes the other customers that have to be served from the route. The unrouted customers will be sorted from smallest time window to the widest time window. Then it tries to insert the first customers in the

unrouted list in the route at the best position possible. The advantage of the sorted list is that the customers with the smallest time windows are inserted in a route when there are still a lot possibilities.

At some point, it is possible that there is no feasible insertion left. The algorithm then removes the first customer in the route after the starting point and inserts that customer at the end of the list of unrouted customers. That will continue until a feasible solution is found or after $\hat{\theta}$ iterations of removing customers from the route.

Algorithm 3 stops when a feasible solution is obtained or when the algorithm cannot identify a feasible solution. Only our problem does not always have a feasible solution. In that case, the algorithm will run for one time and inserts the customers in a place where constraint violation is minimized. It also applies afterwards an additional 100 swap moves in order to have a small check if the solution was the best solution found. To convert this algorithm to an online vehicle routing problem not only the minimization problem has been added, also the starting point of the route has to be determined. Normally, the starting point of the route is the depot, but in an online environment this will be the (x, y) location of the vehicle at that time. The endpoint of a vehicle route is normally a customer, but in a vehicle routing problem this last customer is the depot. Algorithm 3 gives a summary of the algorithm.

Algorithm 3

Step 1 Insert all vertices of $V \setminus \{v_0\}$ in an ordered set S in nondecreasing order of their times window width. Construct a first route (v_0, v_0) .

Step 2 If $S = \emptyset$, a feasible solution has been obtained. Otherwise, set $k = 1$ and $s = |S|$.

Step 3 Select the k th vertex from S and attempt to insert it between each pair consecutive customers in the route. A successful attempt to insert the customer is if the route remains feasible after insertion. If no feasible insertion can be identified, set $k = k + 1$; if $k = s + 1$, go to Step 4; if $k \leq s$, repeat Step 3. If at least one feasible insertion can be identified, implement the best, remove the vertex just inserted from S , update the path, and go to Step 2.

Step 4 No feasible insertion could be identified in Step 3. Remove from the path the vertex v_i immediately following v_0 and place it at the end of S . If the number of times v_i is thus removed from the path is equal to $\hat{\theta}$, stop: the algorithm cannot identify a feasible solution. Otherwise, set $k = 1$ and go to Step 3.

Step 5 If no feasible solution has been obtained, insert all vertices of $V \setminus \{v_0\}$ in an ordered set S in nondecreasing order of their times window width. Construct a first path (v_0, v_0) . For all vertices of S : Insert it in the path where constraint violation is minimal. If the path is constructed, do 100 swap moves and pick the best route.

4.4.2 Switching strategy

The idea behind this strategy is that a vehicle transfers a customer between two routes, such that the costs of an infeasible route are minimized or the infeasible route is made feasible. This requires not only a customer to switch between the vehicles, but also a meeting place and time for the two involved vehicles and even enough capacity for the vehicle that

takes over that customer.

First, the search is for the customers of the infeasible route that can be switched. It is possible trying to move all customers one by one, but that takes a lot of time and is not very effective. Since feasibility is the priority, only the customers are chosen that make the route infeasible. In other words, if a customer is removed from the route and the route then becomes feasible, that customer will be considered for moving to another route and added to a list.

If a candidate can be found for placing it in another route, the best vehicle and place in the associated route has to be determined. An option is to search for a vehicle that is geographically close to the concerned vehicle, since intuitively it is more likely that transferring a customer to nearby vehicles has a bigger chance of improving the current solution than to vehicles that are far away from each other. So, the algorithm searches for all customer locations that the other vehicles pass in a certain range from the concerned customer; the concerned customer will be placed in the route before one of those customers. It is possible that the new route of the concerned customer becomes infeasible or already was infeasible before that new customer was inserted, but still that option will be considered. Eventually, it gives an extra option to move a customer and it can still be checked if it is better than the original solution without moving the considered customer.

Still, the demand of the customer has to be transferred between the vehicles and this requires a meeting place and time. The meeting place depends on the place in the new route of the concerned customer, because the meeting between the vehicles must be happened before the concerned customer is served.

It also has to be considered that the route of the vehicle that has taken over the customer is now longer and probably has more tight constraints, since the vehicle has its original route with now the meeting place and extra customer. The original route of the concerned customer has becomes shorter now that the customer has moved to another route and probably has more tolerance for delay. This means that the best option that minimizes the time window violation probably lies somewhere close or even on the route of the vehicle that has taken over the customer.

Therefore, a list will be constructed for all possible meeting places. The algorithm first determines the (x, y) location of the midpoint of the line segment between the two vehicle locations and from that it determines all possible meeting locations within a certain range from that midpoint. After that, the list will be filled with possible meeting locations on the new route of the concerned customer. The possible meeting places then are inserted into both routes and checked whether it is feasible or not. It remembers the best solution found, even if it is unfeasible. At the end of the algorithm the best meeting place is found, it remains to be check if this solution is better than the original solution without moving the concerned customer.

The meeting place is dependent of the meeting time. One vehicle cannot start transferring the products of the concerned customer before the other is arrived, so it gives a synchronizing problem. The optimal solution then can be found by moving the meeting place in both routes and setting the time that the vehicle can start serving as $\max(a_{m_1}, a_{m_2})$, where a_{m_v} is the arrival time at the meeting place of vehicle v . Algorithm 4 gives a summary of the algorithm.

Algorithm 4

Step 1: Set $i := 1$.

Step 2: Remove customer i from the route. If the route becomes feasible, go to step 3. Otherwise, set $i := i + 1$ and return to step 1. If i is bigger than the number of customers in that route, the algorithm ends.

Step 3: Find all customers within a range r_1 of customer i . Add these customers to a list P and set $u := 1$. If there are no customers within this range, repeat step 3 and set $i := i + 1$. If i is bigger than the number of customers in the route of customer i , the algorithm ends.

Step 4: Remove customer i from its current route and insert customer i before customer u in the route of customer u . Construct a list of meeting places M . This list exists of the locations of the customers that have to be served before customer i by the vehicle of customer u and all customer locations within a range r_2 of the location that is exactly in the middle of the location of the original vehicle of i and the current vehicle of i . If there are no locations in M stored, then set $u := u + 1$ and repeat step 4. If u is bigger than the length of list P , set $i = i + 1$ and return to step 3.

Step 5: Insert the meeting place in the new route of customer i at the first position to be served by the vehicle. Insert the meeting place in the original route of customer i also at the first position to be served by the vehicle. Set the opening of the time window of the meeting place at $\max(a_{veh_1}, a_{veh_2})$ and determine $f(r)$, where f is the objective function and r is the current set of routes. Move the meeting place one place up in the original route of i and determine $f(r)$. Continue this until the meeting place is at the last position in the route before the depot. Now move the meeting place one position up in the new route of customer i and set the meeting place in the old route of customer i at the first position to be served by the vehicle. Repeat this until the meeting place reaches the position before the position of customer i in the new route of customer i .

Step 6: Choose the set of routes that belongs to the minimum value found of $f(r)$. If $f_{r_{best}} < f_{r_{orig}}$, where r_{orig} is the original set of routes, then the set of routes belonging to $f_{r_{best}}$ is the new set of routes.

5 Computational results

In this chapter, the developed online re-planning method will be evaluated. The method will be evaluated by comparing the offline solution against the online solution, which is the offline solution that will be adjusted during the process. Since there are no instances available, the instances of Solomon (1987) [13] will be adjusted to develop an instance that considers the (non-)recurrent disturbances. First, the instance together with the adjustments will be explained and after that the results of evaluating the instance will be presented and discussed.

5.1 Test Instances

The standard for evaluating heuristics or algorithms for the VRPTW is using the instances developed by Solomon (1987) [13]. These Solomon instances have 100 customers and each customer i is associated with an (x_i, y_i) location, a demand, time window and a service time. Each customer has to be served at the corresponding location and within their time window. There are paths between each pair of customers and the length of these paths are equal to the Euclidean distance between the pair of customers. Also, the vehicles have a maximum capacity and therefore the demand of all customers in one route cannot exceed the vehicle capacity. The goal is to minimize the number of vehicles and the total distance. Solomon constructed three types of instances where the spreading of the customers is different. The first type is the R-type, where the locations of the customers are randomly spread. The second type is the C-type, where customers are placed in different clusters. The last type is the RC-type and is a mix of the R-type and the C-type, where there are randomly placed and clustered customers. All of these instances distinguish two cases, one that has a long scheduling horizon and one that has a relatively short scheduling horizon. The instances with a short scheduling horizon permit the vehicles to serve about 5 to 10 customers, while the instances with a long scheduling horizon permit vehicles to serve more than 30 customers. These instances will also be adjusted to create an instance that considers the (non-)recurrent disturbances to evaluate the developed method.

5.2 Adjustment of the instances

The Solomon instances use the Euclidean distance as the length of a path between a pair of customers. In this case that distance between two points is considered as the traveltime without any disturbances.

Recurrent congestion in the method is implemented with the use of time intervals. The impact of the recurrent congestion influences the traveltime, the time it takes to travel a path. Each time interval, the original traveltimes or distances are then multiplied with a random factor between 1 and 1.5 and causes a small delay in the system. The width of the time intervals depend on the number of intervals, which has to be set, and the length of the simulation. The number of time intervals is set on 4 and the length of the simulation is equal to the closing time of the depot, e_0 , which differs per instance type. After the closing time of the depot, the impact of the recurrent disturbances is equal to impact of recurrent disturbances in the last time interval, which results in traveltimes that are multiplied with the same factor as in the last interval.

Considering the non-recurrent disturbances, it is assumed that if a non-recurrent disturbance occurs, there is perfect knowledge about the duration and the impact of that disturbance. So at the time of a non-recurrent disturbance, the end time and the impact are also known and can therefore be incorporated in the online solution method. First, the number of non-recurrent disturbances is randomly chosen between one and four disturbances. Also, the place of each non-recurrent disturbance is also randomly generated. For each disturbance, two customers are randomly chosen and the path between those customers is the path where the disturbance happens. It is assumed that if a vehicle is on the same path as the disturbance at the time of the disturbance, the vehicle will experience the impact of the disturbance.

For each disturbance, a random begin- and end-time have to be generated. The start-time and end-time depend on the path where the disturbance happens, because it is desired that the disturbance has a significant impact on the current solution. Keep in mind that a path is a connection between to customers and that those customers have to be served at

some time by a vehicle. If the start-time of the disturbance is earlier than the opening of the time windows of both customers and the end-time later than the closing time of both time windows, it gives an disturbance around the time the vehicle is nearby one of the two customers and causes delay for the vehicle. Therefore, the start-time is chosen to be 1 and 30 time units before the opening of the earliest time window and the end time between 1 and 30 time units after the latest time window of the two concerned customers.

The impact of a non-recurrent disturbance influences the traveltime of the path of the disturbance, but also influences the traveltime of nearby paths. At the path of the disturbance, the traveltime will be multiplied with an impact factor between 1 and 3. The impact factor is calculated by generating a random number, RN , between zero and two and then add 1 to that random number, which gives a random factor between 1 and 3. The impact of the disturbance on nearby paths can now be determined. Within a range of 3 of the endpoints corresponding to the path of the disturbance, the traveltime on that path will be multiplied with a factor $1 + 0.75 * RN$. For paths within a range between 3 and 20 of the endpoints, the traveltime will be multiplied with a factor $1 + 0.5 * RN$.

5.3 Numerical results

First, the offline method will be evaluated. This will be done by running the method using the original Solomon instances and in that way, the achievement of the tabu search heuristic can be compared with the results of other heuristics. After that, the online method can be evaluated by running both the offline method and the online method, with the same initial set of routes, on the same instance.

5.3.1 Evaluating the offline solution method

The tabu search has a few parameters that have to be set. Higgins (2001) [7] has tested the used tabu search on several instances and determined the best parameter values. For ϕ and Φ these are 10 and 50, respectively. The best sample size found is 1600 and the best length of the tabu list is 1000 for significantly improving the final solution. The tabu search is for large problems, which would be a problem with 2000 customers in our case. Therefore, the tabu list is reduced from 1000 to 500. The values of ϕ and Φ are set on the value found by Higgins.

The values α , β and γ corresponding to the penalty of the total number of times arriving too late, capacity overflow constraint and the lateness penalty are set on respectively 5000, 500 and 10. Since feasibility is a must, it is better to have a strong penalty for the total number of times of arriving too late instead of a strong lateness penalty. Namely, there is a chance that the lateness of a vehicle at a customer is close to zero and then it gives a very small penalty. Still the customer is served too late, so instead, if a strong penalty for the number of times of arriving too late is used, it has a much bigger influence on the cost function and therefore the result. Only for the tabu search, an extra penalty of 5000 is added per used vehicles in order to minimize the number used vehicles.

The tabu search heuristic will be tested for two different values for the limit of the run time, because then the impact of time at the heuristic can be examined. In this case, the values for limit are set on: 200 and 1000 seconds. The results of the tabu search are given in Table 1. Every result in the table is based on 10 repetitions with the corresponding settings, which can be found in Appendix A. The averages are based on the results that gave a feasible solution, since the published solutions are also feasible. Our solutions did not always construct a feasible solution; all R-type problems gave one or more infeasible

Run time (s)	200				1000			
	Average		Best		Average		Best	
	Traveltime	NV*	Traveltime	NV	Traveltime	NV	Traveltime	NV
R101	2700,66 (5)**	20,4	2644,17	20	2689,87 (3)	20,33	2642,5	20
R102	2472,63 (8)	19,5	2386,44	19	2471,69 (8)	19,75	2413,28	19
C101	867,04	10	848,79	10	864	10	848,79	10
C102	1489,02	11	1383,42	11	1491,01	11	1240,59	11

TABLE 1: Results of the tabu search on the Solomon problems

* Number of vehicles

**Average distance(number of feasible solutions)

solutions. The number of feasible solutions for each results are given after the average distance of the corresponding result. If it is not given, all 10 repetitions obtained a feasible solution.

In general, in Table A it can be seen that there is a slight improvement if the run time increases. In fact, all results scored better with a run time of 1000 except for the R102 problem. The average value of both the results with a run time of 200 and the run time of 1000 are about the same, while the best value is around 2% better with a run time of 200. The best improvement can be seen at the C102 problem, which gives an improvement of almost 12 %.

Compared to other heuristics, the developed tabu search does not achieve very good results with respect to the total traveltime. Table 2 compares the best published solutions with the solutions that are achieved here. Most of our solutions here score a higher number of vehicles and a higher total distance. Remarkable are the results of the C-type problems; the best published solutions of the C101 and the C102 are achieved using the same algorithm and the values are almost equal, while our best solution of the C102 is almost 50 % higher than the C101 and has a vehicle more. Also note that the R-instances are even worse in terms of differences compared to the best published solution. Not only is the number of vehicles higher, also the absolute difference between the published solution and our solution is around the 1000 and is a relative big difference.

Problems	Best published solution			Our best solution	
	Number of vehicles	Traveltime	Reference	Number of vehicles	Traveltime
R101	19	1645.79	H*	20	2642.5
R102	17	1486.12	RT**	19	2413.28
C101	10	828.94	RT	10	848.79
C102	10	828.95	RT	11	1240.59

TABLE 2: Offline solution: Best published solution against our best solution

*Hombberger (2013) [9]

**Rochat & Taillard (1995)[12]

5.3.2 Evaluating the online solution method

The online solution method first needs an offline solution as the initial set of routes that will be adjusted during the day. The offline solution is constructed using the tabu search heuristic. The run time for the tabu heuristic is set on 200 seconds, while the other settings of the tabu search are all the same as in the evaluation of the offline method. However, the parameters in the cost function are changed. A decision has to be made which one is more important, the total number of customers that are too late served or the total lateness. The choice here is to give priority to the total lateness and that gives $\alpha = 5000$, $\beta = 500$ and $\gamma = 20000$.

The first part of the online re-planning, the insertion strategy, has only one parameter; that is the number of times a customer can be removed, $\hat{\theta}$. The results in of Gendreau et al. (1998) [5] pointed out that $\hat{\theta} = 4$ is the best option to use.

The second part of online re-planning, customer switching, can only vary in the neighbourhood search for possible new places for the customer and possible meeting places for the two concerned vehicles. However, increasing the range for new possible places or more meeting places does not directly imply a better solution. Increasing the range at some point gives more places to investigate, only the additional new places are further away than the old places before increasing the range. This implies that a longer distance has to be traveled and the chance of feasibility decreases. Because of that, the range is set at 15 for all instances.

The method also needs certain time epochs of when the routes will be updated. Updating the routes will be done after a certain time period or time interval, which gives updating moments that are evenly spaced in time. The parameter values for the width of these intervals will be tested for an interval of 20 time units and 150 time units.

Problems	Offline			Online		
	Avg. traveltime	Avg. total lateness (time)	Avg. Lateness (num. of customers)	Avg. Distance	Avg. total lateness (time)	Avg. Lateness (num. of customers)
R101	3458,78	52,87	8,18	3474,78	44,15	6,84
R102	3101,18	44,20	6,52	3111,07	30,75	4,92
C101	1232,08	10,43	0,74	1239,10	9,86	0,64
C102	1735,27	4,17	0,48	1726,97	3,59	0,30

TABLE 3: Offline method against online method, update interval is set at 20.

Using these settings, the solution methods were tested by doing 50 runs for each instance. For each run, the offline solution is determined and after that the statistics will be collected for offline solution and for the online solution, which is the results of our online re-planning method. The results of the runs can be found in Appendix B.

Table 3 gives the statistics of the 50 runs for each instance and a update interval of 20 time units. It can be noticed that the online method cannot guarantee feasibility, but it can reduce the lateness or total number of customers too late served. The infeasibility reduction comes with a cost, because in all cases the total distance became slightly higher. Also, the improvement of the average lateness here implies an improvement in the number of customers that are too late served.

The same table for the update interval of 150 time units can be found in the Appendix

Problems	Avg. improvement traveltime	Avg. improvement lateness (time)	Avg. Improvement lateness (number of customers)
R101	0*%	14%	17%
	0%	5%	7%
R102	0%	21%	19%
	0%	4%	3%
C101	0%	3%	3%
	0%	0%	0%
C102	0%	12%	12%
	0%	7%	8%

TABLE 4: Online improvements compared with offline solutions

*Time interval 20

Time interval 150

B. The differences in improvements between the update interval of 20 and 150 can be seen in Table 4. It gives the improvement in percentages compared to the offline method. It directly stands out that the improvement in distance is zero, which also can be concluded from Table 3. The total average traveltime increases, but it is less than 1 % . Remarkable is that the online method made small improvements with an update interval of 20 on the C101 problem, but no significant improvements are made with an update interval of 150. The online method has made an improvement on all other problem sets, but it is noticeable that the improvements are most of the times significantly better using the update interval 20. Looking at the problems, it can be seen that the improvements were also slightly better on the R-type problems.

6 Discussion

First, it can be seen that the developed model in general reduces the infeasibility of a set of routes. One important role was the update interval of the model; the online model with an update interval of 150 time units had a significant better improvement than the online model with an interval time of 20 time unit.

Another factor that could have impact on the effect of the online model is the used instance type. In the results it can be seen that the online model achieved much better results on the R-type instances than on the C-type instances. That would imply that this method achieves better results with random customer location than with clustered customer locations. There is only one factor that has to be considered. The offline solution of the R-type instances achieved a result that is 1.6 times the result of the best published solution using the original Solomon instances. The results of the online method tested on the R-type on the adjusted Solomon instances can then be questioned. The most simple reason could be that improvements are very easily made, since the spreading of customers is random and may cause routes to be close to each other which makes it easier to transfer customers to another vehicle.

Another explanation for the better results of the online solution method on the R-type instance could be that the offline solution has left a lot of opportunities for improvement in the routes, since the offline solution did not achieve very good results on R-type instances and the online method has made relatively a lot of improvements. That would imply that the spread of customer location has not much influence on the online solution method. Only the results do not confirm this hypothesis. Other way around, it could be said that

a good offline solution method implies not much improvements in the routes, since a good offline solution method does not leave many opportunities for improvements. This agrees with the results of the online method tested on the C101 instance, but disagrees with the results of the online method on the C102 instance. The offline solution did not achieve good results on the original Solomon instances compared to the best published solution, but there were relatively little improvements. Therefore, further research is needed for the influence of the spread of customer locations on the online solution method.

However, it could be that the instance type has influence on the offline method. The offline method constructs an initial solution for the VRPTW using a nearest neighbour strategy. In clustered instances, the nearest neighbour method probably constructs a route that serves one cluster of customers and gives an initial solution that probably achieves good results without even using the tabu search. On the other hand, the R-type instances have a random spreading of customer locations. The results of the nearest neighbour technique may be not as good as with the clustered instance. A reason for this is the fact that the first constructed routes probably have very good routes, but the last constructed routes may have to deal with customer locations that are widely spread across the field of customer locations since the first constructed routes may have used the most optimal customers. The tabu search then may improve the routes a little, but since there is only one swap at the time, the tabu search may not have that much effect. So, the use of the nearest neighbour strategy may be the reason that the offline solution achieved not well on the R-type instances.

The influence of individual strategies is also not totally clear. In appendix B, it can be seen how many customers are transferred during a run. It can be seen that transferring a customer to another vehicle in the R-type instances is relatively often used compared to the use of transferring customers in C-type instances. In a clustered instance the routes are probably based on serving a whole cluster, especially using the nearest neighbour strategy. It is then probably much harder to switch customers, since the vehicles are probably in another cluster. If that clusters are close to each other, a transfer may be possible. How further away the other clusters, how harder it is probably going to be to transfer a customer to another vehicle. On the other hand there are the R-type instances where there are more customers transferred. It could be that there are more customers in the neighbourhood served by another vehicle than in clustered instances and would imply that it is profitable to transfer a customer in a situation where customer locations are randomly spread. It could also be noticed in appendix B that if there is a significant improvement made, it most of the times involves one or more transfers of a customer. One note should be made when this strategy was used. If the two vehicles arrived at the meeting place, the service time was set on zero. This means that it would take no time to transfer the demands of a customer to the other vehicle and is not realistic.

There are also a few cases where the offline method achieved better results than the online method. An explanation could be that the routes were adjusted in order to avoid the impact of a non-recurrent event, but that a few moments later another recurrent disturbance occurred. It is possible that the second disturbance could have had more impact on that current online solution than on the original offline solution, which could cause more lateness or more customers that were too late served at the online solution.

7 Conclusion

In this paper, a dynamic vehicle routing problem with time windows, recurrent and non-

recurrent disturbances has been studied. The goal was to develop and evaluate an online re-planning model in order to improve a set of routes in case of non-recurrent disturbances. The method that is introduced in this paper constructs an initial set of routes using a tabu search heuristic based on swap- and insertion moves and then tries to improve the set of routes after a set time interval. There are two strategies used to improve the routes online: an insertion heuristic that tries to find the best order of customers in a single route and a strategy that involves another vehicle taking over a customer. Using an adapted version of the Solomon instances, the online route could be tested and compared with an offline model, which was evaluated using the original Solomon instances. The online model is tested with different update intervals on instances with clustered customers and instances with random placed customers. The results of the online model gave improvements compared to the offline model, but only significant improvements under certain circumstances. These circumstances are a relative small update interval and where the vehicle routes are relatively close to each other. Based on that, it can be said that an online re-planning method can be developed by transferring customers and changing the order of customers. Evaluating an online re-planning methodology requires more than scheduling random (non-)recurrent disturbances, it needs more statistics from the instance itself, as the place, time and impact of a non-recurrent disturbance or the number of disturbances, otherwise only speculations instead of explanations about certain events can be given.

As a recommendation, this research showed that the strategy of transferring customers to another vehicle has a good prospect. The strategy is able to significantly reduce the lateness of vehicles under the right circumstances. It is also topic that needs more research. At this moment, the algorithm switches only one customer per infeasible route. There are also possibilities to take over multiple customers or even a whole part of the route. Not only the number of customers to switch can be researched. The expansion can also consider the definition of the neighbourhood. In this case, the neighbourhood exists of locations of customers in the route of the vehicle that takes over the customer and all customer locations in a certain radius. Further research can be done by adding a third strategy to the problem, namely path flexibility. At this moment, the Solomon instances are the most used problem sets to evaluate vehicle routing problems with time windows. However, these instances do not consider any path flexibility, while in real problems path flexibility could give good solutions.

References

- [1] Olli Bräysy and Michel Gendreau. Tabu search heuristics for the vehicle routing problem with time windows. *Top*, 10(2):211–237, 2002.
- [2] Zhi-Long Chen and Hang Xu. Dynamic column generation for dynamic vehicle routing with time windows. *Transportation Science*, 40(1):74–88, 2006.
- [3] Jean-François Cordeau, Gilbert Laporte, Martin WP Savelsbergh, and Daniele Vigo. Vehicle routing. *Handbooks in operations research and management science*, 14:385–428, 2007.
- [4] Michel Gendreau, Francois Guertin, Jean-Yves Potvin, and Eric Taillard. Parallel tabu search for real-time vehicle routing and dispatching. *Transportation science*, 33(4):381–390, 1999.

- [5] Michel Gendreau, Alain Hertz, Gilbert Laporte, and Mihnea Stan. A generalized insertion heuristic for the traveling salesman problem with time windows. *Operations Research*, 46(3):330–335, 1998.
- [6] Fred Glover and Eric Taillard. A user’s guide to tabu search. *Annals of operations research*, 41(1):1–28, 1993.
- [7] Andrew J Higgins. A dynamic tabu search for large-scale generalised assignment problems. *Computers & operations research*, 28(10):1039–1048, 2001.
- [8] Sin C Ho and Dag Haugland. A tabu search heuristic for the vehicle routing problem with time windows and split deliveries. *Computers & Operations Research*, 31(12):1947–1964, 2004.
- [9] Jörg Homberger. *Verteilt-parallele Metaheuristiken zur Tourenplanung: Lösungsverfahren für das Standardproblem mit Zeitfensterrestriktionen*. Springer-Verlag, 2013.
- [10] Soumia Ichoua, Michel Gendreau, and Jean-Yves Potvin. Vehicle dispatching with time-dependent travel times. *European journal of operational research*, 144(2):379–396, 2003.
- [11] Victor Pillac, Michel Gendreau, Christelle Guéret, and Andrés L Medaglia. A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225(1):1–11, 2013.
- [12] Yves Rochat and Éric D Taillard. Probabilistic diversification and intensification in local search for vehicle routing. *Journal of heuristics*, 1(1):147–167, 1995.
- [13] Marius M Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research*, 35(2):254–265, 1987.
- [14] Éric Taillard, Philippe Badeau, Michel Gendreau, François Guertin, and Jean-Yves Potvin. A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation science*, 31(2):170–186, 1997.

A Results of Tabu search algorithm

run	Dis	NV	Cus_ toolate	Time_ toolate	run	Dis*	NV**	Cus_ toolate	Time_ toolate
1	848,79	10	0	0	1	848,79	10	0	0
2	848,79	10	0	0	2	879,20	10	0	0
3	879,20	10	0	0	3	848,79	10	0	0
4	848,79	10	0	0	4	879,20	10	0	0
5	879,20	10	0	0	5	848,79	10	0	0
6	848,79	10	0	0	6	848,79	10	0	0
7	879,20	10	0	0	7	879,20	10	0	0
8	879,20	10	0	0	8	879,20	10	0	0
9	879,20	10	0	0	9	848,79	10	0	0
10	879,20	10	0	0	10	879,20	10	0	0

TABLE 5: Results of tabu search without recurrent disturbances on test instance C101, left table uses a run time of 200 and the right table uses a run time of 1000

*distance

** Number of vehicles

run	Dis	NV	Cus_ toolate	Time_ toolate	run	Dis	NV	Cus_ toolate	Time_ toolate
1	1691,66	11	0	0	1	1441,71	11	0	0
2	1386,48	11	0	0	2	1630,93	11	0	0
3	1454,28	11	0	0	3	1468,67	11	0	0
4	1383,41	11	0	0	4	1462,91	11	0	0
5	1439,25	11	0	0	5	1240,59	11	0	0
6	1556,48	11	0	0	6	1597,04	11	0	0
7	1518,98	11	0	0	7	1572,62	11	0	0
8	1501,39	11	0	0	8	1314,32	11	0	0
9	1540,43	11	0	0	9	1569,76	11	0	0
10	1417,81	11	0	0	10	1611,59	11	0	0

TABLE 6: Results of tabu search on test instance C102, left table uses a run time of 200 and the right table uses a run time of 1000

run	Dis	NV	Cus_ toolate	Time_ toolate
1	2661,42	20,00	0	0,00
2	2650,29	20,00	1	1,66
3	2644,17	20,00	0	0,00
4	2782,92	21,00	0	0,00
5	2667,29	20,00	1	7,00
6	2644,17	20,00	0	0,00
7	2650,29	20,00	1	1,66
8	2770,63	21,00	0	0,00
9	2650,29	20,00	1	1,66
10	2667,29	20,00	1	7,00

run	Dis	NV	Cus_ toolate	Time_ toolate
1	2667,29	20	1	7,00
2	2642,50	20	0	0
3	2661,42	20	0	0
4	2667,29	20	1	7,00
5	2650,29	20	1	1,66
6	2667,29	20	1	7,00
7	2650,29	20	1	1,66
8	2650,29	20	1	1,66
9	2765,68	21	0	0
10	2633,04	20	1	1,66

TABLE 7: Results of tabu search on test instance R101, left table uses a run time of 200 and the right table uses a run time of 1000

run	Dis	NV	Cus_ toolate	Time_ toolate
1	2428,92	19	0	0
2	2538,61	20	0	0
3	2533,25	20	0	0
4	2409,24	19	1	4,24
5	2386,44	19	0	0
6	2538,61	20	0	0
7	2435,78	19	1	3,22
8	2407,21	19	0	0
9	2525,73	20	0	0
10	2422,24	19	0	0

run	Dis	NV	Cus_ toolate	Time_ toolate
1	2530,14	20	0	0
2	2503,20	20	0	0
3	2422,24	19	0	0
4	2447,16	20	0	0
5	2382,73	19	1	3,22
6	2413,25	19	1	3,22
7	2525,73	20	0	0
8	2413,24	19	0	0
9	2489,86	20	0	0
10	2441,96	20	0	0

TABLE 8: Results of tabu search on test instance R102, left table uses a run time of 200 and the right table uses a run time of 1000

B Results of the developed model

run	Dist_off	Dist_on	Perc better	Cus_toolate_off	Cus_toolate_on	Perc better	Time_toolate_off	Time_toolate_on	Perc better	Meet places	Agree
1	1101,32	1101,32	0%	0	0	0%	0,00	0,00	0%	0	1
2	1006,42	1006,42	0%	0	0	0%	0,00	0,00	0%	0	1
3	2167,38	2255,93	-4%	2	0	100%	21,29	0,00	100%	1	0
4	1143,09	1143,09	0%	0	0	0%	0,00	0,00	0%	0	1
5	1000,14	1000,14	0%	0	0	0%	0,00	0,00	0%	0	1
6	1053,33	1053,33	0%	0	0	0%	0,00	0,00	0%	0	1
7	952,84	952,84	0%	0	0	0%	0,00	0,00	0%	0	1
8	1097,63	1097,63	0%	0	0	0%	0,00	0,00	0%	0	1
9	1119,80	1119,80	0%	0	0	0%	0,00	0,00	0%	0	1
10	1048,72	1048,72	0%	0	0	0%	0,00	0,00	0%	0	1
11	1707,51	1707,51	0%	0	0	0%	0,00	0,00	0%	0	1
12	1733,02	1733,02	0%	1	1	0%	1,36	1,36	0%	0	1
13	1023,12	1023,12	0%	0	0	0%	0,00	0,00	0%	0	1
14	991,73	991,73	0%	0	0	0%	0,00	0,00	0%	0	1
15	1056,77	1056,77	0%	0	0	0%	0,00	0,00	0%	0	1
16	982,23	982,23	0%	0	0	0%	0,00	0,00	0%	0	1
17	1792,13	1792,13	0%	0	0	0%	0,00	0,00	0%	0	1
18	1804,29	1804,29	0%	1	1	0%	0,02	0,02	0%	0	1
19	1013,87	1013,87	0%	0	0	0%	0,00	0,00	0%	0	1
20	1089,66	1089,66	0%	1	1	0%	2,69	2,69	0%	0	1
21	1097,29	1097,29	0%	0	0	0%	0,00	0,00	0%	0	1
22	1092,00	1092,00	0%	0	0	0%	0,00	0,00	0%	0	1
23	1073,31	1073,31	0%	0	0	0%	0,00	0,00	0%	0	1
24	1057,53	1057,53	0%	0	0	0%	0,00	0,00	0%	0	1
25	1033,58	1033,58	0%	0	0	0%	0,00	0,00	0%	0	1
26	1793,27	1940,03	-8%	14	12	14%	255,05	248,47	3%	1	0
27	1119,04	1119,04	0%	4	4	0%	11,15	11,15	0%	0	1
28	1116,69	1116,69	0%	0	0	0%	0,00	0,00	0%	0	1
29	984,37	984,37	0%	0	0	0%	0,00	0,00	0%	0	1
30	1015,65	1015,65	0%	0	0	0%	0,00	0,00	0%	0	1
31	1067,01	1067,01	0%	0	0	0%	0,00	0,00	0%	0	1
32	978,36	978,36	0%	0	0	0%	0,00	0,00	0%	0	1
33	1041,22	1041,22	0%	0	0	0%	0,00	0,00	0%	0	1
34	1775,67	1891,58	-7%	2	1	50%	1,39	1,00	28%	1	0
35	971,82	971,82	0%	0	0	0%	0,00	0,00	0%	0	1
36	1044,93	1044,93	0%	0	0	0%	0,00	0,00	0%	0	1
37	1019,10	1019,10	0%	0	0	0%	0,00	0,00	0%	0	1
38	1786,90	1786,90	0%	3	3	0%	6,00	6,00	0%	0	1
39	1051,56	1051,56	0%	0	0	0%	0,00	0,00	0%	0	1
40	1095,42	1095,42	0%	0	0	0%	0,00	0,00	0%	0	1
41	1689,58	1689,58	0%	0	0	0%	0,00	0,00	0%	0	1
42	1671,52	1671,52	0%	0	0	0%	0,00	0,00	0%	0	1
43	1190,48	1190,48	0%	8	8	0%	221,63	221,63	0%	0	1
44	1065,36	1065,36	0%	0	0	0%	0,00	0,00	0%	0	1
45	1092,66	1092,66	0%	0	0	0%	0,00	0,00	0%	0	1
46	1132,95	1132,95	0%	1	1	0%	0,74	0,74	0%	0	1
47	1084,48	1084,48	0%	0	0	0%	0,00	0,00	0%	0	1
48	1081,04	1081,04	0%	0	0	0%	0,00	0,00	0%	0	1
49	1825,39	1825,39	0%	0	0	0%	0,00	0,00	0%	0	1
50	1670,63	1670,63	0%	0	0	0%	0,00	0,00	0%	0	1

TABLE 9: Results of the online model with update interval 20 against the offline model on problem C101

run	Dist_off	Dist_on	Perc_better	Cus_toolate_off	Cus_toolate_on	Perc_better	Time_toolate_off	Time_toolate_on	Perc_better	Meet_places	Agree
1	1742,68	1742,68	0%	0	0	0%	0,00	0,00	0%	0	1
2	1669,59	1669,59	0%	0	0	0%	0,00	0,00	0%	0	1
3	1052,65	1052,65	0%	1	1	0%	3,23	3,23	0%	0	1
4	1151,52	1151,52	0%	0	0	0%	0,00	0,00	0%	0	1
5	1849,02	1849,02	0%	1	1	0%	0,78	0,78	0%	0	1
6	1082,63	1082,63	0%	0	0	0%	0,00	0,00	0%	0	1
7	1118,85	1118,85	0%	9	9	0%	72,21	72,21	0%	0	1
8	1079,10	1079,10	0%	0	0	0%	0,00	0,00	0%	0	1
9	1010,80	1010,80	0%	0	0	0%	0,00	0,00	0%	0	1
10	1780,57	1780,57	0%	1	1	0%	3,02	3,02	0%	0	1
11	948,16	948,16	0%	0	0	0%	0,00	0,00	0%	0	1
12	1002,77	1002,77	0%	0	0	0%	0,00	0,00	0%	0	1
13	1032,73	1032,73	0%	0	0	0%	0,00	0,00	0%	0	1
14	1794,49	1794,49	0%	0	0	0%	0,00	0,00	0%	0	1
15	1129,31	1129,31	0%	5	5	0%	29,70	29,70	0%	0	1
16	991,46	991,46	0%	1	1	0%	2,34	2,34	0%	0	1
17	1031,63	1031,63	0%	0	0	0%	0,00	0,00	0%	0	1
18	1095,91	1095,91	0%	0	0	0%	0,00	0,00	0%	0	1
19	1134,82	1134,82	0%	0	0	0%	0,00	0,00	0%	0	1
20	1039,72	1039,72	0%	0	0	0%	0,00	0,00	0%	0	1
21	1021,08	1021,08	0%	0	0	0%	0,00	0,00	0%	0	1
22	1739,67	1739,67	0%	0	0	0%	0,00	0,00	0%	0	1
23	1056,38	1056,38	0%	0	0	0%	0,00	0,00	0%	0	1
24	996,04	996,04	0%	0	0	0%	0,00	0,00	0%	0	1
25	1020,01	1020,01	0%	0	0	0%	0,00	0,00	0%	0	1
26	2744,58	2744,58	0%	0	0	0%	0,00	0,00	0%	0	1
27	956,55	956,55	0%	0	0	0%	0,00	0,00	0%	0	1
28	1146,62	1146,62	0%	0	0	0%	0,00	0,00	0%	0	1
29	1032,40	1032,40	0%	0	0	0%	0,00	0,00	0%	0	1
30	1783,81	1783,81	0%	2	2	0%	4,51	4,51	0%	0	1
31	1005,41	1005,41	0%	0	0	0%	0,00	0,00	0%	0	1
32	995,55	995,55	0%	0	0	0%	0,00	0,00	0%	0	1
33	1634,76	1634,76	0%	5	5	0%	25,62	25,62	0%	0	1
34	988,62	988,62	0%	0	0	0%	0,00	0,00	0%	0	1
35	2302,32	2302,32	0%	0	0	0%	0,00	0,00	0%	0	1
36	1763,36	1763,36	0%	4	4	0%	16,34	16,34	0%	0	1
37	1024,13	1024,13	0%	0	0	0%	0,00	0,00	0%	0	1
38	1630,14	1630,14	0%	0	0	0%	0,00	0,00	0%	0	1
39	1061,78	1061,78	0%	0	0	0%	0,00	0,00	0%	0	1
40	1173,74	1173,74	0%	1	1	0%	3,42	3,42	0%	0	1
41	1770,68	1770,68	0%	0	0	0%	0,00	0,00	0%	0	1
42	1024,09	1024,09	0%	0	0	0%	0,00	0,00	0%	0	1
43	1692,10	1692,10	0%	0	0	0%	0,00	0,00	0%	0	1
44	1023,84	1023,84	0%	0	0	0%	0,00	0,00	0%	0	1
45	1109,91	1109,91	0%	0	0	0%	0,00	0,00	0%	0	1
46	1071,48	1071,48	0%	0	0	0%	0,00	0,00	0%	0	1
47	1030,88	1030,88	0%	0	0	0%	0,00	0,00	0%	0	1
48	1084,98	1084,98	0%	0	0	0%	0,00	0,00	0%	0	1
49	1052,20	1052,20	0%	0	0	0%	0,00	0,00	0%	0	1
50	1114,78	1114,78	0%	0	0	0%	0,00	0,00	0%	0	1

TABLE 10: Results of the online model with update interval 150 against the offline model on problem C101

run	Dist_off	Dist_on	Perc better	Cus toolate_off	Cus toolate_on	Perc better	Time_toolate_off	Time_toolate_on	Perc better	Meet_places	Agree
1	1530,40	1520,44	1%	0	0	0%	0,00	0,00	0%	0	0
2	1762,20	1828,66	-4%	1	0	100%	0,28	0,00	100%	0	0
3	1775,70	1762,30	1%	0	0	0%	0,00	0,00	0%	0	0
4	1512,17	1508,60	0%	0	0	0%	0,00	0,00	0%	0	0
5	2284,08	2234,94	2%	0	0	0%	0,00	0,00	0%	0	0
6	2095,46	2072,86	1%	0	0	0%	0,00	0,00	0%	0	0
7	1488,10	1487,70	0%	0	0	0%	0,00	0,00	0%	0	0
8	1641,34	1638,23	0%	0	0	0%	0,00	0,00	0%	0	0
9	2001,46	1990,63	0%	0	0	0%	0,00	0,00	0%	0	0
10	1965,29	1962,78	0%	0	0	0%	0,00	0,00	0%	0	0
11	1534,40	1516,89	1%	0	0	0%	0,00	0,00	0%	0	0
12	1471,27	1466,50	0%	1	1	0%	1,91	1,91	0%	0	0
13	2198,80	2198,70	0%	0	0	0%	0,00	0,00	0%	0	0
14	1415,42	1401,30	1%	0	0	0%	0,00	0,00	0%	0	0
15	1365,05	1360,28	0%	1	0	100%	0,29	0,00	100%	0	0
16	1594,57	1594,41	0%	0	0	0%	0,00	0,00	0%	0	0
17	1472,72	1468,77	0%	2	2	0%	3,29	3,29	0%	0	0
18	2284,59	2273,77	0%	0	0	0%	0,00	0,00	0%	0	0
19	1547,34	1544,49	0%	0	0	0%	0,00	0,00	0%	0	0
20	2208,15	2145,78	3%	0	0	0%	0,00	0,00	0%	0	0
21	1569,50	1540,96	2%	0	0	0%	0,00	0,00	0%	0	0
22	1571,18	1691,40	-8%	4	0	100%	22,07	0,00	100%	1	0
23	1341,91	1341,07	0%	0	0	0%	0,00	0,00	0%	0	0
24	1777,21	1756,48	1%	9	9	0%	167,92	167,92	0%	0	0
25	1621,73	1590,95	2%	0	0	0%	0,00	0,00	0%	0	0
26	1575,82	1575,17	0%	0	0	0%	0,00	0,00	0%	0	0
27	1554,43	1536,70	1%	0	0	0%	0,00	0,00	0%	0	0
28	1285,27	1285,27	0%	0	0	0%	0,00	0,00	0%	0	1
29	2284,90	2280,90	0%	0	0	0%	0,00	0,00	0%	0	0
30	1892,60	1890,51	0%	0	0	0%	0,00	0,00	0%	0	0
31	1606,22	1601,60	0%	0	0	0%	0,00	0,00	0%	0	0
32	1422,85	1453,26	-2%	1	0	100%	0,84	0,00	100%	0	0
33	1516,81	1508,24	1%	1	1	0%	2,26	2,26	0%	0	0
34	1566,63	1532,21	2%	0	0	0%	0,00	0,00	0%	0	0
35	2138,39	2129,12	0%	0	0	0%	0,00	0,00	0%	0	0
36	1617,33	1604,07	1%	1	1	0%	1,09	1,09	0%	0	0
37	1437,88	1411,92	2%	0	0	0%	0,00	0,00	0%	0	0
38	1698,29	1695,42	0%	0	0	0%	0,00	0,00	0%	0	0
39	2315,35	2294,86	1%	0	0	0%	0,00	0,00	0%	0	0
40	1947,51	1947,23	0%	0	0	0%	0,00	0,00	0%	0	0
41	1533,53	1527,96	0%	0	0	0%	0,00	0,00	0%	0	0
42	1815,03	1814,06	0%	0	0	0%	0,00	0,00	0%	0	0
43	1727,27	1656,85	4%	0	0	0%	0,00	0,00	0%	0	0
44	1904,62	1902,14	0%	0	0	0%	0,00	0,00	0%	0	0
45	2365,52	2344,37	1%	0	0	0%	0,00	0,00	0%	0	0
46	1411,71	1392,67	1%	0	0	0%	0,00	0,00	0%	0	0
47	1656,70	1599,15	3%	1	1	0%	2,96	2,96	0%	0	0
48	1605,72	1631,76	-2%	1	0	100%	3,97	0,00	100%	0	0
49	1859,99	1829,72	2%	0	0	0%	0,00	0,00	0%	0	0
50	1993,11	2004,33	-1%	1	0	100%	1,61	0,00	100%	0	0

TABLE 11: Results of the online model with update interval 20 against the offline model on problem C102

run	Dist_off	Dist_on	Perc_better	Cus_toolate_off	Cus_toolate_on	Perc_better	Time_toolate_off	Time_toolate_on	Perc_better	Meet_places	Agree
1	2280,45	2278,74	0%	0	0	0%	0,00	0,00	0%	0	0
2	1798,35	1797,19	0%	1	1	0%	0,56	0,56	0%	0	0
3	1682,65	1676,43	0%	0	0	0%	0,00	0,00	0%	0	0
4	1595,87	1594,90	0%	6	6	0%	30,32	30,32	0%	0	0
5	1505,80	1440,48	4%	0	0	0%	0,00	0,00	0%	0	0
6	1723,03	1715,68	0%	0	0	0%	0,00	0,00	0%	0	0
7	1523,69	1591,08	-4%	7	6	14%	48,81	47,71	2%	0	0
8	1910,31	1889,52	1%	0	0	0%	0,00	0,00	0%	0	0
9	1539,26	1507,73	0%	0	0	0%	0,00	0,00	0%	0	0
10	2166,38	2166,38	0%	0	0	0%	0,00	0,00	0%	0	0
11	1270,00	1254,76	1%	0	0	0%	0,00	0,00	0%	0	0
12	1911,33	1859,18	3%	0	0	0%	0,00	0,00	0%	0	0
13	1921,41	1919,46	0%	1	1	0%	0,61	0,61	0%	0	0
14	1855,83	1845,37	1%	2	2	0%	9,17	9,17	0%	0	0
15	2351,71	2351,71	0%	0	0	0%	0,00	0,00	0%	0	1
16	1490,83	1592,36	-7%	10	7	30%	78,61	32,24	59%	0	0
17	2290,77	2230,86	3%	0	0	0%	0,00	0,00	0%	0	0
18	1371,27	1353,57	1%	0	0	0%	0,00	0,00	0%	0	0
19	1673,61	1671,65	0%	0	0	0%	0,00	0,00	0%	0	0
20	1915,11	1911,73	0%	2	2	0%	3,90	3,90	0%	0	0
21	1954,58	1954,58	0%	2	2	0%	8,53	8,53	0%	0	1
22	1881,05	1881,00	0%	1	1	0%	4,84	4,84	0%	0	0
23	2000,85	1993,85	0%	0	0	0%	0,00	0,00	0%	0	0
24	1624,65	1571,99	3%	0	0	0%	0,00	0,00	0%	0	0
25	1880,07	1872,77	0%	0	0	0%	0,00	0,00	0%	0	0
26	1398,37	1353,85	3%	6	6	0%	87,84	87,84	0%	0	0
27	1662,28	1658,50	0%	0	0	0%	0,00	0,00	0%	0	0
28	1731,20	1728,29	0%	0	0	0%	0,00	0,00	0%	0	0
29	1692,40	1687,97	0%	0	0	0%	0,00	0,00	0%	0	0
30	1544,18	1513,99	2%	1	1	0%	0,63	0,63	0%	0	0
31	2583,85	2571,14	0%	0	0	0%	0,00	0,00	0%	0	0
32	1747,17	1783,50	-2%	12	5	58%	603,50	404,74	33%	1	0
33	1588,08	1564,09	2%	6	4	33%	27,58	26,18	5%	0	0
34	1923,26	1939,47	-1%	5	3	40%	46,28	24,86	46%	0	0
35	2336,50	2315,37	1%	1	1	0%	0,25	0,25	0%	0	0
36	1640,26	1591,69	3%	6	6	0%	49,58	49,58	0%	0	0
37	1942,19	2100,24	-8%	2	0	100%	1,43	0,00	100%	0	0
38	1586,34	1574,98	1%	0	0	0%	0,00	0,00	0%	0	0
39	1376,63	1374,46	0%	0	0	0%	0,00	0,00	0%	0	0
40	1619,73	1614,16	0%	2	2	0%	2,75	2,75	0%	0	0
41	1653,72	1640,11	1%	0	0	0%	0,00	0,00	0%	0	0
42	2373,92	2366,08	0%	0	0	0%	0,00	0,00	0%	0	0
43	1748,53	1745,08	0%	1	1	0%	3,99	3,99	0%	0	0
44	1531,52	1528,25	0%	1	1	0%	1,27	1,27	0%	0	0
45	1602,64	1564,50	2%	0	0	0%	0,00	0,00	0%	0	0
46	1713,94	1604,17	6%	3	0	100%	61,68	0,00	100%	0	0
47	2142,37	2087,03	3%	0	0	0%	0,00	0,00	0%	0	0
48	1486,26	1459,67	2%	0	0	0%	0,00	0,00	0%	0	0
49	1894,08	1891,23	0%	3	3	0%	10,21	10,21	0%	0	0
50	3050,42	3050,42	0%	0	0	0%	0,00	0,00	0%	0	1

TABLE 12: Results of the online model with update interval 150 against the offline model on problem C102

run	Dist_off	Dist_on	Perc_better	Cus_toolate_off	Cus_toolate_on	Perc_better	Time_toolate_off	Time_toolate_on	Perc_better	Meet_places	Agree
1	3286,94	3283,90	0%	20	15	25%	144,89	87,78	39%	2	0
2	3622,95	3642,10	-1%	12	11	8%	161,61	128,75	20%	1	0
3	3616,98	3656,91	-1%	9	8	11%	16,06	11,92	26%	1	0
4	3564,56	3564,56	0%	6	6	0%	33,17	33,17	0%	0	1
5	3584,10	3614,68	-1%	10	9	10%	24,06	22,54	6%	1	0
6	3329,61	3329,61	0%	8	8	0%	39,58	39,58	0%	0	1
7	3568,99	3582,82	0%	10	9	10%	78,03	75,84	3%	1	0
8	3355,76	3363,39	0%	4	1	75%	20,60	6,35	69%	1	0
9	2844,41	2844,41	0%	0	0	0%	0,00	0,00	0%	0	1
10	3225,12	3225,12	0%	12	12	0%	92,78	92,78	0%	0	1
11	3381,28	3420,64	-1%	2	1	50%	5,42	3,54	35%	1	0
12	3162,03	3162,03	0%	10	10	0%	76,14	76,14	0%	0	1
13	3503,01	3503,01	0%	3	3	0%	2,34	2,34	0%	0	1
14	3791,62	3791,62	0%	4	4	0%	15,58	15,58	0%	0	1
15	3338,55	3336,16	0%	13	12	8%	98,77	95,56	3%	1	0
16	3321,15	3352,63	-1%	11	8	27%	91,99	61,23	33%	2	0
17	3383,73	3415,27	-1%	7	3	57%	58,13	53,73	8%	1	0
18	3690,41	3704,13	0%	7	6	14%	14,23	11,93	16%	1	0
19	3468,30	3482,14	0%	7	6	14%	38,89	34,28	12%	1	0
20	3525,06	3553,51	-1%	18	12	33%	55,46	35,66	36%	4	0
21	3551,70	3551,70	0%	4	4	0%	19,38	19,38	0%	0	1
22	3549,04	3549,04	0%	7	7	0%	29,35	29,35	0%	0	1
23	3378,40	3378,40	0%	8	8	0%	42,82	42,82	0%	0	1
24	3044,81	3038,55	0%	11	8	27%	81,03	53,10	34%	1	0
25	3383,56	3446,71	-2%	4	2	50%	2,71	1,72	37%	2	0
26	3529,07	3529,07	0%	4	4	0%	18,92	18,92	0%	0	1
27	3787,36	3802,23	0%	7	6	14%	17,64	16,27	8%	1	0
28	3781,62	3810,86	-1%	8	5	38%	49,63	39,40	21%	1	0
29	3791,55	3791,55	0%	9	9	0%	92,18	92,18	0%	0	1
30	3650,62	3650,62	0%	9	9	0%	22,92	22,92	0%	0	1
31	3369,49	3379,99	0%	17	13	24%	158,29	167,95	-6%	1	0
32	3553,45	3608,12	-2%	11	9	18%	23,44	16,72	29%	2	0
33	3184,18	3223,59	-1%	12	11	8%	46,34	39,03	16%	1	0
34	3398,36	3411,97	0%	30	22	27%	445,99	295,12	34%	2	0
35	3428,06	3428,06	0%	4	4	0%	17,33	17,33	0%	0	1
36	3633,34	3633,34	0%	1	1	0%	5,32	5,32	0%	0	1
37	3770,71	3788,25	0%	10	10	0%	43,82	46,94	-7%	1	0
38	3130,55	3131,72	0%	6	3	50%	42,78	26,05	39%	1	0
39	3858,57	3912,89	-1%	11	10	9%	77,97	74,76	4%	1	0
40	2992,75	2992,75	0%	3	3	0%	11,01	11,01	0%	0	1
41	3535,23	3560,80	-1%	5	1	80%	59,12	25,88	56%	1	0
42	3138,28	3138,28	0%	11	11	0%	38,62	38,62	0%	0	1
43	3492,38	3538,30	-1%	4	3	25%	11,10	10,01	10%	1	0
44	2957,28	2968,21	0%	1	0	100%	1,57	0,00	100%	1	0
45	3497,81	3497,81	0%	2	2	0%	14,93	14,93	0%	0	1
46	3485,90	3485,90	0%	10	10	0%	55,35	55,35	0%	0	1
47	3596,65	3603,47	0%	3	3	0%	8,86	8,03	9%	1	0
48	3748,08	3748,08	0%	8	8	0%	61,69	61,69	0%	0	1
49	3441,80	3523,46	-2%	10	8	20%	45,18	42,18	7%	2	0
50	3713,73	3786,53	-2%	6	4	33%	30,65	25,69	16%	2	0

TABLE 13: Results of the online model with update interval 20 against the offline model on problem R101

run	Dist_off	Dist_on	Perc better	Cus_toolate_off	Cus_toolate_on	Perc better	Time_toolate_off	Time_toolate_on	Perc better	Meet places	Agree
1	3032,59	3044,36	0%	12	11	8%	54,05	54,03	0%	1	0
2	3670,40	3689,85	-1%	5	4	20%	10,58	8,40	21%	1	0
3	3098,55	3108,34	0%	5	4	20%	27,61	25,04	9%	1	0
4	3837,34	3837,34	0%	5	5	0%	32,09	32,09	0%	0	1
5	3430,47	3430,47	0%	2	2	0%	37,79	37,79	0%	0	1
6	3769,41	3769,41	0%	2	2	0%	8,88	8,88	0%	0	1
7	3785,15	3785,15	0%	11	11	0%	55,32	55,32	0%	0	1
8	3101,27	3114,56	0%	13	12	8%	88,62	86,74	2%	1	0
9	3464,58	3464,58	0%	7	7	0%	32,67	32,67	0%	0	1
10	3688,61	3727,75	-1%	9	8	11%	42,81	22,36	48%	1	0
11	3660,15	3660,15	0%	9	9	0%	35,23	35,23	0%	0	1
12	3677,29	3688,47	0%	11	10	9%	48,77	46,76	4%	1	0
13	3241,21	3241,21	0%	2	2	0%	12,60	12,60	0%	0	1
14	2972,23	2984,70	0%	12	9	25%	59,15	38,18	35%	1	0
15	3573,91	3573,91	0%	4	4	0%	21,96	21,96	0%	0	1
16	3376,87	3397,66	-1%	3	1	67%	14,68	13,50	8%	1	0
17	3175,05	3195,59	-1%	9	8	11%	44,27	42,02	5%	1	0
18	3826,20	3867,02	-1%	13	12	8%	63,43	55,66	12%	1	0
19	3716,26	3716,26	0%	9	9	0%	51,90	51,90	0%	0	1
20	3410,76	3422,65	0%	5	4	20%	21,84	21,39	2%	1	0
21	3578,78	3601,98	-1%	6	5	17%	30,23	29,58	2%	1	0
22	3663,56	3663,56	0%	9	9	0%	52,84	52,84	0%	0	1
23	3137,91	3137,91	0%	8	8	0%	42,76	42,76	0%	0	1
24	3332,38	3332,38	0%	3	3	0%	6,08	6,08	0%	0	1
25	3045,28	3045,28	0%	11	11	0%	60,14	60,14	0%	0	1
26	3314,12	3314,12	0%	9	9	0%	22,65	22,65	0%	0	1
27	3543,29	3543,29	0%	13	13	0%	61,08	61,08	0%	0	1
28	3236,98	3236,98	0%	2	2	0%	2,24	2,24	0%	0	1
29	3055,35	3055,35	0%	8	8	0%	40,68	40,68	0%	0	1
30	3932,11	3930,65	0%	7	6	14%	32,88	32,73	0%	1	0
31	3369,79	3395,37	-1%	18	17	6%	156,41	155,33	1%	1	0
32	3341,26	3341,26	0%	9	9	0%	53,66	53,66	0%	0	1
33	2826,17	2826,17	0%	12	12	0%	134,23	134,23	0%	0	1
34	3340,17	3340,17	0%	9	9	0%	54,72	54,72	0%	0	1
35	3493,11	3493,11	0%	8	8	0%	48,80	48,80	0%	0	1
36	3027,59	3041,49	0%	5	4	20%	10,17	8,16	20%	1	0
37	3271,67	3271,67	0%	6	6	0%	69,96	69,96	0%	0	1
38	2998,18	3018,31	-1%	12	11	8%	139,92	139,62	0%	1	0
39	3085,22	3098,08	0%	10	9	10%	27,05	25,45	6%	1	0
40	3768,31	3780,84	0%	8	6	25%	60,24	58,53	3%	1	0
41	3515,33	3515,33	0%	5	5	0%	35,82	35,82	0%	0	1
42	3273,17	3302,89	-1%	4	3	25%	5,73	3,24	43%	1	0
43	3664,81	3664,81	0%	11	11	0%	116,91	116,91	0%	0	1
44	3298,94	3298,94	0%	0	0	0%	0,00	0,00	0%	0	1
45	3572,72	3572,72	0%	10	10	0%	50,07	50,07	0%	0	1
46	3279,55	3279,55	0%	9	9	0%	27,53	27,53	0%	0	1
47	3151,65	3151,65	0%	4	4	0%	38,66	38,66	0%	0	1
48	3724,05	3724,05	0%	3	3	0%	0,38	0,38	0%	0	1
49	3505,07	3514,88	0%	4	3	25%	2,40	2,10	13%	1	0
50	3076,60	3076,60	0%	12	12	0%	63,66	63,66	0%	0	1

TABLE 14: Results of the online model with update interval 150 against the offline model on problem R101

run	Dist_off	Dist_on	Perc better	Cus_toolate_off	Cus_toolate_on	Perc better	Time_toolate_off	Time_toolate_on	Perc better	Meet places	Agree
1	3121,71	3117,91	0%	4	2	50%	28,50	5,89	79%	1	0
2	2799,14	2851,72	-2%	9	8	11%	35,84	34,36	4%	1	0
3	2860,36	2860,36	0%	3	3	0%	2,23	2,23	0%	0	0
4	3311,28	3417,43	-3%	24	17	29%	239,61	144,04	40%	4	0
5	2712,75	2696,62	1%	10	10	0%	59,76	59,76	0%	0	0
6	2858,68	3000,44	-5%	28	16	43%	231,39	137,69	40%	4	0
7	2855,88	2880,81	-1%	9	8	11%	51,18	50,97	0%	0	0
8	2886,70	2897,49	0%	9	6	33%	44,64	36,61	18%	1	0
9	3512,69	3516,67	0%	7	5	0%	24,48	15,19	0%	1	0
10	3175,71	3228,32	-2%	12	2	83%	108,13	23,64	78%	2	0
11	3354,32	3380,67	-1%	5	7	-40%	19,52	34,94	-79%	1	0
12	2883,15	2875,84	0%	5	5	0%	29,87	29,87	0%	0	0
13	3430,59	3430,59	0%	10	10	0%	44,00	44,00	0%	0	1
14	2842,98	2842,57	0%	3	2	33%	3,64	1,75	52%	1	0
15	3243,13	3243,13	0%	1	1	0%	1,09	1,09	0%	0	0
16	3523,65	3539,53	0%	9	7	22%	49,54	45,52	8%	1	0
17	3302,36	3302,36	0%	0	0	0%	0,00	0,00	0%	0	0
18	3639,04	3639,04	0%	8	8	0%	56,26	56,26	0%	0	0
19	3414,98	3414,13	0%	6	6	0%	18,25	18,25	0%	0	0
20	3071,11	3081,65	0%	15	11	27%	258,40	156,19	40%	1	0
21	3236,19	3236,19	0%	3	3	0%	11,48	11,48	0%	0	0
22	3168,78	3177,27	0%	10	6	40%	42,52	38,36	10%	3	0
23	2955,57	2955,57	0%	4	4	0%	15,82	15,82	0%	0	0
24	2549,90	2549,90	0%	0	0	0%	0,00	0,00	0%	0	0
25	3186,74	3194,83	0%	5	4	20%	41,27	14,28	65%	1	0
26	2862,05	2862,05	0%	1	1	0%	3,14	3,14	0%	0	0
27	3207,54	3207,54	0%	2	2	0%	10,07	10,07	0%	0	0
28	2586,70	2593,22	0%	1	1	0%	11,53	4,16	64%	0	0
29	2864,16	2864,16	0%	1	1	0%	2,00	2,00	0%	0	0
30	3102,58	3110,30	0%	6	6	0%	67,67	59,91	11%	1	0
31	3280,30	3280,30	0%	2	2	0%	1,48	1,48	0%	0	1
32	2761,54	2770,27	0%	8	6	25%	52,23	38,63	26%	0	0
33	3222,68	3229,25	0%	2	0	100%	4,83	0,00	100%	0	0
34	3002,20	2992,10	0%	5	1	80%	33,16	1,30	96%	1	0
35	3397,38	3397,38	0%	13	13	0%	117,26	117,26	0%	0	1
36	2920,67	2909,75	0%	17	9	47%	100,10	64,75	35%	2	0
37	3299,94	3299,94	0%	8	8	0%	37,84	37,84	0%	0	0
38	3336,57	3320,14	0%	9	6	33%	50,52	18,01	64%	1	0
39	3319,81	3319,81	0%	1	1	0%	2,05	2,05	0%	0	0
40	3032,04	3049,43	-1%	4	2	50%	26,21	11,74	55%	1	0
41	3207,75	3207,75	0%	6	6	0%	22,33	22,33	0%	0	0
42	3014,47	3014,47	0%	2	2	0%	12,24	12,24	0%	0	0
43	3127,62	3143,68	-1%	6	2	67%	57,81	11,44	80%	0	0
44	3108,39	3108,39	0%	2	2	0%	1,26	1,26	0%	0	0
45	2958,06	2955,90	0%	4	4	0%	12,58	12,58	0%	0	0
46	3156,17	3156,17	0%	9	9	0%	28,56	28,56	0%	0	0
47	3062,48	3062,48	0%	5	5	0%	70,46	70,46	0%	0	1
48	3260,79	3260,79	0%	4	4	0%	23,99	23,99	0%	0	0
49	3074,29	3056,83	1%	5	2	60%	29,09	4,19	86%	1	0
50	2995,26	3050,26	-2%	4	0	100%	13,95	0,00	100%	1	0

TABLE 15: Results of the online model with update interval 20 against the offline model on problem R102

run	Dist_off	Dist_on	Perc_better	Cus_toolate_off	Cus_toolate_on	Perc_better	Time_toolate_off	Time_toolate_on	Perc_better	Meet_places	Agree
1	3131,73	3131,73	0%	1	1	0%	0,12	0,12	0%	2	0
2	3274,85	3274,85	0%	10	10	0%	114,39	114,39	0%	1	0
3	3103,00	3103,00	0%	8	8	0%	44,06	44,06	0%	1	0
4	3393,66	3393,66	0%	9	9	0%	50,17	50,17	0%	0	1
5	2994,72	2994,72	0%	9	9	0%	34,28	34,28	0%	1	0
6	3169,21	3169,21	0%	9	9	0%	89,81	89,81	0%	0	1
7	2924,02	2924,02	0%	3	3	0%	4,70	4,70	0%	1	0
8	3324,56	3324,56	0%	7	7	0%	37,13	37,13	0%	1	0
9	2854,54	2854,54	0%	16	16	0%	109,35	109,35	0%	0	1
10	2872,20	2872,20	0%	4	4	0%	11,31	11,31	0%	0	1
11	2944,73	2944,73	0%	6	6	0%	37,45	37,45	0%	1	0
12	2750,45	2750,45	0%	2	2	0%	5,35	5,35	0%	0	1
13	2763,13	2763,13	0%	1	1	0%	0,18	0,18	0%	0	1
14	3238,37	3251,61	0%	7	5	29%	12,29	8,28	33%	0	1
15	3206,57	3206,57	0%	5	5	0%	12,70	12,70	0%	1	0
16	2778,10	2778,10	0%	5	5	0%	21,93	21,93	0%	2	0
17	2998,25	2998,25	0%	12	12	0%	103,36	103,36	0%	1	0
18	3437,00	3437,00	0%	13	13	0%	126,86	126,86	0%	1	0
19	2944,08	2944,08	0%	8	8	0%	40,76	40,76	0%	1	0
20	2913,31	2913,31	0%	11	11	0%	144,96	144,96	0%	4	0
21	3104,89	3116,69	0%	8	7	13%	51,81	46,42	10%	0	1
22	3237,08	3231,89	0%	4	2	50%	29,28	8,34	72%	0	1
23	3523,47	3523,47	0%	5	5	0%	8,59	8,59	0%	0	1
24	3174,97	3174,97	0%	5	5	0%	10,99	10,99	0%	1	0
25	3205,66	3205,66	0%	6	6	0%	41,02	41,02	0%	2	0
26	2969,44	2969,44	0%	8	8	0%	57,85	57,85	0%	0	1
27	3134,63	3134,63	0%	6	6	0%	15,38	15,38	0%	1	0
28	3513,67	3513,67	0%	5	5	0%	12,49	12,49	0%	1	0
29	3472,43	3472,43	0%	5	5	0%	10,14	10,14	0%	0	1
30	2762,33	2762,33	0%	1	1	0%	5,60	5,60	0%	0	1
31	3010,09	3010,09	0%	9	9	0%	40,94	40,94	0%	1	0
32	3255,69	3255,69	0%	6	6	0%	68,92	68,92	0%	2	0
33	2717,10	2717,10	0%	13	13	0%	58,13	58,13	0%	1	0
34	3565,99	3565,99	0%	10	10	0%	31,54	31,54	0%	2	0
35	3012,75	3012,75	0%	6	6	0%	13,21	13,21	0%	0	1
36	2679,13	2679,13	0%	4	4	0%	18,50	18,50	0%	0	1
37	2768,14	2768,14	0%	3	3	0%	16,23	16,23	0%	1	0
38	3275,20	3275,20	0%	4	4	0%	4,98	4,98	0%	1	0
39	3283,70	3283,70	0%	9	9	0%	48,80	48,80	0%	1	0
40	3663,91	3663,91	0%	36	36	0%	602,48	602,48	0%	0	1
41	3221,06	3221,06	0%	6	6	0%	41,98	41,98	0%	1	0
42	3242,05	3242,05	0%	1	1	0%	4,20	4,20	0%	0	1
43	3363,26	3384,44	-1%	12	12	0%	72,96	72,57	1%	1	0
44	3199,20	3235,79	-1%	2	1	50%	1,97	0,86	57%	1	0
45	3261,92	3261,92	0%	0	0	0%	0,00	0,00	0%	0	1
46	3020,26	3011,01	0%	9	8	11%	134,42	110,53	18%	0	1
47	2772,02	2772,02	0%	6	6	0%	53,71	53,71	0%	1	0
48	3383,09	3383,09	0%	9	9	0%	75,16	75,16	0%	0	1
49	3137,42	3137,50	0%	13	12	8%	136,09	127,81	6%	2	0
50	2869,78	2869,78	0%	7	7	0%	38,86	38,86	0%	2	0

TABLE 16: Results of the online model with update interval 150 against the offline model on problem R102