An approach to solve the Multi Depot Vehicle Routing Problem with Time Windows (MDVRPTW) in static and dynamic scenarios

Thomas Timmermann[1] and René Schumann[2]

[1] Goethe University, Frankfurt, Germany,
[2] Goethe University, Information Systems and Simulation, Robert-Mayer-Str. 10, 60325 Frankfurt am Main, Germany
[timmerm|reschu]@informatik.uni-frankfurt.de

Abstract. In this article we present an approach to solve the Multi Depot Vehicle Routing Problem with Time Windows (MDVRPTW) in different scenarios. First we present the optimization in a static scenario using an adaptive neighborhood search combined with a tabu search heuristic. In a dynamic scenario events occur that change the data the actual plan is based on, like the customers demand, the number of customers or the needed traveling time between two customers. These events have to be integrated into the current plan, to provide a valid route for all trucks. The well-known MDVRPTW instances by Cordeau et al. [4] are used in this approach. To evaluate the presented planning approach in dynamic environments we use a c-competitive analysis, which emphasizes the efficiency of the repair mechanisms.

1 Introduction

Transportation is an integral part in today's economic planning. The German Federal Ministry of Transport, Building and Urban Affairs calculated the transportation of freight of 434,1 billions tonne kilometers on German roads in 2006. Experts predict an increase of 12.4 % to 488 billions tonne kilometers until 2010 [14]. This situation makes it necessary to optimize routes, especially considering the increasing prices for fossil fuels. It is essential for logistic planning systems to consider additional restrictions such as delivery time windows of customers or maximum duration of routes. During the last decades there has been an increased demand for tour planning systems capable of integrating updated customer data and tracking and tracing information [9]. The following section presents different scenarios and challenges of routing problems regarding the aforementioned restrictions. In section 3 an algorithm for the computation of the route plan will be shown. These results will be evaluated with an procedure developed in section 4. Finally, we discuss additional extension to measure dynamism in route planning problems and summarize or findings and discuss future work.

2 Problem Formulation

2.1 The Multi Depot Vehicle Routing Problem with Time Windows (MDVRPTW)

The Multi Depot Vehicle Routing Problem with Time Windows (MDVRPTW) is considered for a practical description of transportation planning. The MDVRPTW describes the problem to deliver uniform goods to a set of customers from a set of depots with vehicles with heterogeneous capacities. The delivery has to be done within a customer-specified time window and the vehicles need to return to the same depot where they have started. Each customer has to be delivered once. A customer i is represented as quintuple $(id_i, g_i, ds_i, e_i, l_i)$. Where id_i identifies a customer i, g_i describes the demand, ds_i the service time of customer i which a vehicle has to wait when it arrives at this customer. Finally, e_i (earliest) as well as l_i (latest) define the earliest respectively latest possible delivery time of i. The MDVRPTW is an extension of the fundamental route planning problem the so-called Vehicle Routing Problem (VRP) from Dantzig and Ramser [5]. As the VRP is NP-hard the MDVRPTW with more restrictions also belongs to the set of NP-hard problems [13]. Therefore it is recommended to use a heuristic approach solving the MDVRPTW. The definition of time windows in this case refers to the definition of so-called soft time windows from Chiang and Russell [2]. In comparison to the definition of hard time windows a route is still feasible although the time window restriction is violated. The resulting delay at one or more customers is measured by a penalty P. P will increase if a vehicle arrives at the customer at time t later than l_i. Actually P will increase by the difference of t and l_i. An arriving before e_i will only result in a waiting time at customer i. The quality of the computed routes is measured by a vector R combining penalty P and total traveling distance D to $R = (P, D)$.

In fact there are two optimization criteria: At first the presented approach will optimize a potential penalty, secondly it tries to reduce the covered distance as much as possible. This approach will be evaluated on the MDVRPTW instances from Cordeau et al. [4]. The problem instances contain four to six depots and 48 to 288 customers which are almost uniformly distributed. The depots possess vehicles with limited capacity. The distances between customers and depots correspond to the euclidean distance; thereby the time complies with the distance.

2.2 Regarded Scenarios

The MDVRPTW is investigated in different scenarios: At first a static scenario is regarded where all planning relevant information is known a priori and will remain unchanged. In the second scenario the execution of the planned routes are simulated and disturbing events occur and their consequences have to be incorporated in the current tour plans. The set of possible dynamic events at customer i can be summarized as follows:

– in- and decrease of demand g_i

- shrinking of time windows due to increasing e_i or decreasing l_i
- growing time windows due to decreasing e_i or increasing l_i
- shortening or extension of distances between customers or depots
- new customer requests
- cancellation of customer requests.

The events that occur during the simulation are randomly generated from the aforementioned set of events and are uniformly distributed in respect to the planning horizon. All events appear with the same probability. Thus only valid events are being created: for example the time of appearance of an event like decreasing the demand g_i of customer i would never exceed l_i.

3 Designing the algorithm

The computation of tours for a given MDVRPTW instance comprises different stages: Primarily the customers will be assigned to their nearest depots. The same method was used by Polacek et al. [13] and by Cordeau et al. [4]. Their algorithms Variable Neighborhood Search (VNS) and Unified Tabu Search Algorithm (UTSA) compute the best-known solutions on the MDVRPTW instances from Cordeau et al.[3] [4].
After all customers have been assigned to a depot the initial tours are constructed with the well-known savings algorithm presented by Clarke and Wright [3] which is sketched in figure 1. The savings c_{ij} resulting from combination of two routes arise from the saved distance.

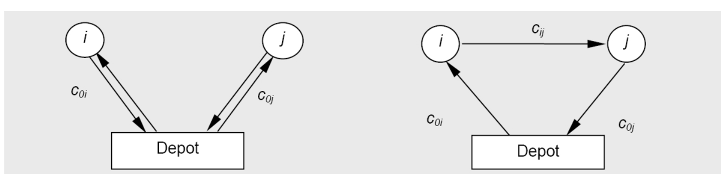

Fig. 1. The savings algorithm [8]

If and only if the result complies with the capacity restriction of the vehicle and P does not increase, the combination of two routes is allowed. This creates an initial solution which can be optimized by the algorithm presented in the following section.

[3] available at http://neumann.hec.ca/chairedistributique/data/mdvrptw/ - last check on 12.06.2008

3.1 The optimization

The optimization of an initial route plan is done according to the Adaptive Large Neighborhood Search (ALNS) from Pisinger and Ropke [12]. ALNS combines a set of different heuristics to explore different regions of the solution space for a good solution. In doing so different strategies for computing the neighborhood are used. Here we use a 2-opt neighborhood presented in [6] as well as a λ-interchange neighborhood with $\lambda = 1$ introduced by Osman [11].
The 2-opt algorithm replaces two edges belonging from the route with two edges not part of the tour so far. The swap operations can be executed within one tour (intra-tour) and between two tours (inter-tour) of a depot, see figure 2. In contrast the λ-interchange algorithm swaps a maximum number of λ vertices (customers) between two tours described best in tuple notation (a,b). a respectively b donates the number of vertices swaped from the first to the second tour and vice versa. In this case the swap possibilities with $\lambda = 1$ are $(1, 0)$, $(0, 1)$ and $(1, 1)$. The combination of 2-opt and λ-Interchange is reasonable as the solutions deriving from the λ-Interchange neighborhood will never be reached with the 2-opt algorithm as the set of customers of a tour in a 2-opt algorithm never changes. Thus the combination of different local search strategies increases the search space, that can be investigated during the optimization phase. The combined neighborhood search optimizes an initial solution while using always the neighborhood solution with the largest improvement (steepest-descent) [1] for further calculation: Either regarding the penalty P or in case of no change of P regarding the lower covered distance. The algorithm chooses only solutions which do not increase P and preserve the vehicles' capacity restrictions. All neighborhood solutions will be tested (best-fit). The applied heuristics are shown in figure 2.

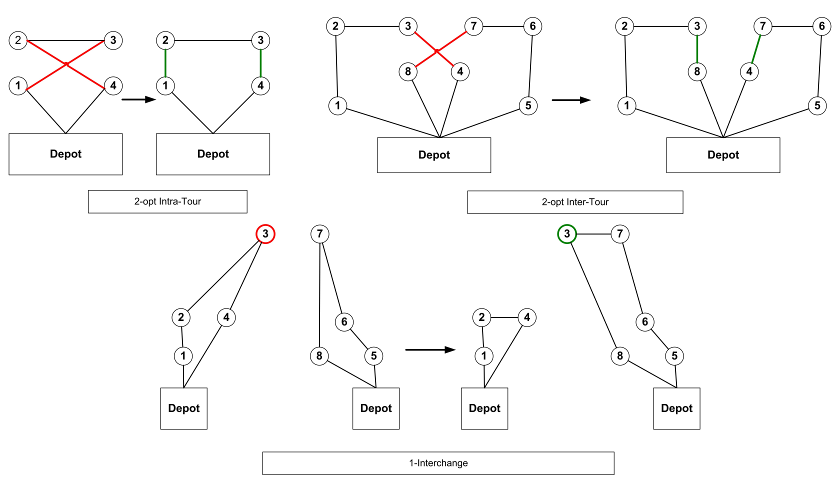

Fig. 2. The Neighborhood Search

The local search strategies can be realized in an efficient and fast way. However, the neighborhood search will probably end in local optima. To escape from this sub-optimal solution the neighborhood search is guided by the metaheuristic tabu search introduced by Glover and Laguna [7]. If the local search runs into a local optimum, this solution is excluded for a given number of search steps in the following, it is set tabu. To do so, it is necessary, that tours are comparable, e.g. see if a solution is currently tabu. To do this in an efficient way only attributes of tours are stored in the tabu list. Due to the definition of the MDVRPTW a customer i is only permitted to be delivered once by a vehicle. Hence i appears only in exactly one route. As a result the best solution (local optimum) distinguishes from the second best solution in exactly one route regarding the 2-opt Intra-Tour neighborhood. This means the route which belongs to the route plan of the local optimum needs to be set tabu for a defined number of iterations; in case of 2-opt Inter-Tour and λ-Interchange the route plan differs in exactly two routes which need to be excluded then. However, the tabooing also prevents the search to find other good solutions previously not known. This is the reason for introducing an aspiration criteria [7] that is able to revoke (aspirate) the tabu of a solution leading the search into a new optimum not known so far (aspiration-by-improved-best). The search terminates if a defined number of iterations is reached. The number of iterations is computed by the multiplication of a constant factor with the instance size (number of customers). Although the search will terminate after a defined number of unsuccessful iterations where no better solution was found.

3.2 Repair mechanism in dynamic scenario

The route plans of the previous section was optimized on the given data of existing customer and depot information. If the tours should be executed, typically changes will occur, as the environment is dynamic. In this section dynamic changes of planning relevant information will be integrated to the existing route plan during its execution. The dynamic events we regard in this work are listed in section 2.2. To simplify the estimation of the vehicles location at each point in time an event occurs, it is assumed that vehicles only revise their routes when they arrive at the next customer. Thus a started trip to the next customer in the route is not interrupted, even if this could be advantageous for the plan repair. If and only if an event refers to a customer already supplied or which is currently in service the updated information cannot be integrated and will be discarded. A customer i gets the status "in service" when a vehicle is already achieving the edge leading to i. Events that change the time window or the demand of an existing customer will be simulated by first removing the customer from the plan and then integrating the customer with the updated information. The insertion will be done using the cheapest-insertion heuristics which checks all possibilities to insert i, choosing the one with the lowest increasing penalty P observing the time flow. If there exists more the one possible insertion point, the distance D is used as second parameter. The repair mechanism triggers a real-time optimization procedure executed after every occurrence of an event;

this procedure is initiated right after the event occurs and necessary repair has been done, if so at all. The real-time optimization is realized with a combined neighborhood search consisting of 2-opt Intra-Tour and 1-Interchange neighborhood already presented in section 3.1. Here the search strategies are combined using the best-fit method and the solutions for further iterations are selected by steepest-descent strategy. In doing so the real-time optimization regards only the customers of status not supplied. At this point meta-heuristics are not applied because of performance reasons. The goal is to keep the reaction time as low as possible to provide a route plan with updated information within a short time frame.

4 Evaluation

The results in the static scenario are compared to the solutions from Cordeau et al. [4]. Thereby it needs to be mentioned that Cordeau et al. respect an additional restriction, the maximum route duration, which is omitted here. A c-competitive analysis is examined for the evaluation of the results in the dynamic scenario as no sources for results in dynamic MDVRPTW instances were found in the literature. The c-competitive analysis indicates by parameter c the factor a dynamic solution is worse in comparison to a static computation where all events were known in advance. Note that the c-competitive analysis addresses the average-case because the events are generated randomly. As described in section 2 the solution quality is defined by vector $R = (P, D)$. To simplify the assignment of c the following formula is used to transform the solution vector R to a simple term:

$$(P, D) = \left(\tfrac{P \cdot 100}{D}\right) \cdot P + D$$

Thus it is possible to integrate the penalty P weighted by the covered distance of the route plan into the term of the formula mentioned above. If (P_{dyn}, D_{dyn}) describes the result of a dynamic computation R_{dyn} and (P_{stat}, D_{stat}) the corresponding static result R_{stat}, c can be easily identified by:

$$c = \tfrac{(P_{dyn}, D_{dyn})}{(P_{stat}, D_{stat})}.$$

4.1 Results

All experiments were run on a Intel Core 2 Duo processor (E6850) with 4096K cache size, 2 GB RAM and a frequency of 3.0 GHz in a Linux-based environment. The results in the static scenario are in average about 4.24 % worse and need four vehicles more per instance than the best-known solutions from Cordeau et al. [4]. The results are shown in table 1. The best parameter configuration was identified by experimental testing. The best solutions were reached with $2 \cdot \#customers$ iterations, 30 unsuccessful iterations, 10 diversification moves and a tabu length of 60 iterations.

instance	UTSA	calculated solution	RPD in %	time in sec.
pr01	1074.12	1089.58	1.42	82
pr02	1762.21	1864.92	5.51	310
pr03	2373.65	2556.38	7.15	1427
pr04	2852.29	2937.15	2.89	6872
pr05	3029.65	3102.74	2.36	10891
pr06	3627.18	3949.66	8.16	12975
pr07	1418.22	1466.98	3.32	229
pr08	2102.61	2296.53	8.44	957
pr09	2737.82	2859.04	4.24	7258
pr10	3505.27	3799.13	7.73	14318
pr11	1005.73	957.48	-5.04	57
pr12	1478.51	1544.18	4.25	509
pr13	2011.24	2131.85	5.66	2227
pr14	2202.08	2336.28	5.74	2661
pr15	2494.57	2737.75	8.88	5643
pr16	2901.02	3048.93	4.85	19844
pr17	1236.24	1201.73	-2.87	402
pr18	1792.61	1922.37	6.75	1806
pr19	2285.1	2401.71	4.86	3020
pr20	3079.16	3091.54	0.40	14111
Avg.			4.24	5279.95

Table 1. Calculated results vs. solutions from Cordeau et al. [4]

In particular on instances with a smaller number of customers the developed approach is better compared to the solution from Cordeau et al. [4] because the restriction of maximum tour durations are not regarded in our approach or because a marginal higher number of used vehicles was used in the calculated tours. The variation of approx. 8 % on a couple of instances can be explained by the fact that the UTSA approach can use unfeasible solutions during the search and thus can investigate a larger search space. Particularly the customers' time window restrictions are responsible for the larger search space that can be explored by the UTSA system compared to our approach.
The consequences of a dynamic environment was tested in different scenarios, with a varying number of events from 12 to 180. All results base on three test runs per scenario. The averaged results are shown interpolated in figure 3. It can be easily seen that the solution quality expressed by factor c decreases with an increasing number of dynamic events.

The response time for an event required a maximum of only 13 seconds on the largest instances. This meets the requirement of short reaction time of dynamic planning systems. As expected events that occur early during the plan execution cause a higher response time, as the valid search space for new solutions is still large. The response times decrease very quickly if the events occur in later phases of plan execution. This issue is shown in figure 4.

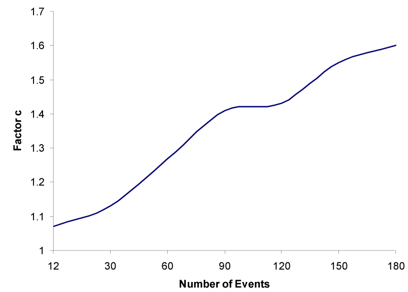

Fig. 3. Results in dynamic scenario

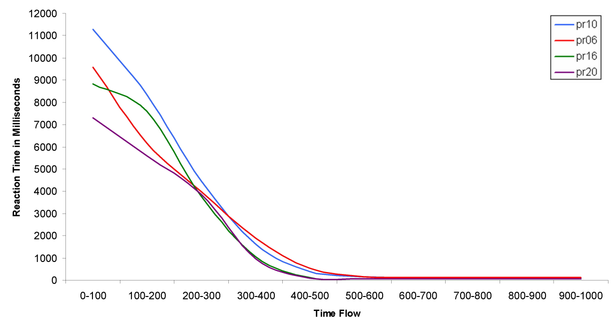

Fig. 4. Reaction time frames on the four largest instances

It can be observed that the results of test runs with the same configuration of the event generator have a high deviation. This leads to the assumption that the solution quality does not only depend on the number of events. Possibly their appearance times have influence on the solution quality as the number of feasible swap operations decreases for late occurring events as already shown in figure 4. To distinguish not only the set of events on the basis of its number a degree of dynamism $\phi_{mdvrptw}$ according to the effective degree of dynamism with time windows (EDODTW) from Larsen [10] is applied. The EDODTW is defined on dynamic vehicle routing problem (DVRP)-instances where only new customers are liable to a time window restriction but not the customers known from the beginning. This makes a slight modification of the EDODTW formula necessary. Thereby defines n_d the number of new customer requests. The following formula makes sure that only the time window restriction of new customers in the MDVRPTW influences $\phi_{mdvrptw}$:

$$\phi_{mdvrptw} = \frac{\sum_{i=1}^{n_d} [T-(l_i-t_i)]/T}{n_d}.$$

To apply this formula here only the event "new customer" is considered. In addition the event *new customer* can be generated in arbitrary time slices and always a feasible event will be created. In contrast the effects of other events, like the change of a customers demand g_i or its time window information e_i and l_i, depends on the occurrence time of the event and the current route plan. The event *new customer* can always be incorporated into an existing route plan.

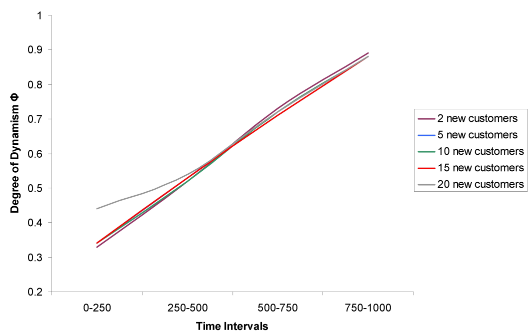

Fig. 5. $\phi_{mdvrptw}$ in different time intervals

As shown interpolated in figure 5 $\phi_{mdvrptw}$ increases almost uniformly independent from the number of events. Figure 6 represents the increase of factor c in accordance to $\phi_{mdvrptw}$ and approves the assumption that both the number of events and their appearance times summarized in $\phi_{mdvrptw}$ are influencing the solution quality. Note that in cases of a minimum of ten new customers the solution quality decreases at $\phi_{mdvrptw} > 0.55$ substantially faster than in dynamic scenarios with a lower $\phi_{mdvrptw}$. However, the factor c has never exceeded 1.8 in the experiments.

4.2 Discussion of further refinements for the degree of dynamism

Unfortunately the presented definition for the degree of dynamism is limited only to one event, namely adding a new customer. As presented above, this event is one out of ten possible events in our model. Here we discuss possible extension for the measurement of dynamism for MDVRPTW problems, which of course can easily be transferred to other tour planning problems. In our discussion here we focus on five events. We look at those events that can possibly lead to infeasible plans or decrease the plan quality. Those events are:

1. Decrease of l_i
2. Increase of e_i
3. Increase of distance d_{ij}
4. Increase of demand g_i of customer i

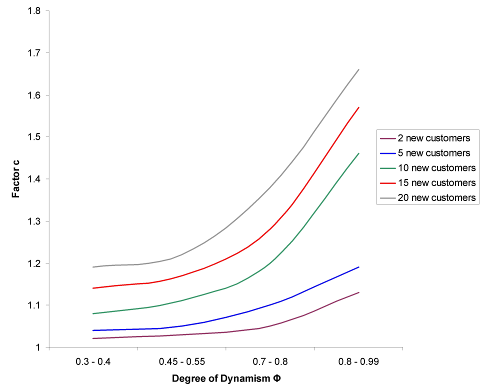

Fig. 6. Factor c according to $\phi_{mdvrptw}$

5. New customer requests.

The other events cannot decrease plan quality. Nevertheless, if such an event occurs, the existing routes might not be optimal anymore, but are still feasible. In the following we do not regard these potentials for plan improvement as reasons for replanning and limit the measurement of the degree of dynamism to the aforementioned five events. The question how the remaining events can be incorporated in the measurement of the degree of dynamism has to be subject of further research.
Let Γ be the set of the regarded events. Then Γ is composed of the following types of γ: It is always necessary to find the latest point in time *pit* when an event has to be integrated into the route plan. *pit* can be calculated using a function h regarding event γ with $h(\gamma)$. The event type of γ is represented as binary quintuple with $\gamma = (c_1, c_2, c_3, c_4, c_5)$: $c_i = 1$ for event type i and $c_j = 0$ $\forall j \neq i$. Thus the formula introduced in section 4.1 will be modified as follows:

$$\phi_{mdvrptw} = \frac{\sum\limits_{\forall \gamma \in \Gamma} [T - (h(\gamma) - t_\gamma)]/T}{|\Gamma|}$$

The function $h(\gamma)$ examines whether or not an event $\gamma \in \Gamma$ and thus should take effect on $\phi_{mdvrptw}$. In case of an event of type 1 (decrease of l_i to l'_i, represented by $\gamma = (1, 0, 0, 0, 0)$), l'_i describes the desired point in time *pit*. In case of an increasing e_i to e'_i causing a delay at one customer j that follows customer i in this route l_j has to be used. If the distance d_{ij} increases between two customers i and j, the *latest-* time restriction l_k of the first following customer k in this route where the time window is missed. If there is an increase of g_i, the value *pit* is given by the departure time to the first following customer x in the considered route whose demand g_x cannot be satisfied by the remaining goods of the vehicle. *pit* in case of event type new customer request i remains l_i as hitherto.

Let us assume a_i is the planned arrival time of the vehicle at customer i, a route $Tour$ represents an ordered list of customers, CAP is the capacity of the vehicle used on the regarded route and Δ describes the modification of the changed information e_i, g_i and d_{ij} to e'_i, g'_i and d'_{ij} with $\Delta_{e_i} = e'_i - e_i$, $\Delta_{g_i} = g'_i - g_i$ and $\Delta_{d_{ij}} = d'_{ij} - d_{ij}$. Then the function $h(\gamma)$ can be defined as follows:

$$h(\gamma) = \begin{cases} l_i, & \text{if } c_1 = 1 \wedge a_i > l_i \\ l_j, & \text{if } c_2 = 1 \wedge \exists\ min\ j \geq i : a_j + \Delta_{e_i} > l_j \\ l_k, & \text{if } c_3 = 1 \wedge \exists\ min\ k > i : a_k + \Delta_{d_{ij}} > l_k \\ a_x - d_{x-1,x}, & \text{if } c_4 = 1 \wedge \Delta_{g_i} + \sum_{\forall j \in Tour} g_j > CAP\text{:} \\ & x \text{ is first customer holds } \sum_{k=n}^{x} g_k > \Delta_{g_i} \\ l_i, & \text{if } c_5 = 1 \\ 0, & \text{else} \end{cases}$$

Of course, the formula holds only under the ceteris paribus assumption, that is here that there exists no event in the future that changes parameters of that formula. A criticism of this extension is the fact that the computation depends on the route planning algorithm used, now. For the calculation of $h(\gamma)$ information about the current route plans are needed, like the sequence of customers in each route and the planned arrival time at each customer. Thus the computation of the degree of dynamism can hardly be done in analytic studies, moreover simulation studies seem more appropriate.
Thus it is debatable if such an extension is reasonable. We are convinced that it is, as it is a first step to broaden the field of dynamic tour planning towards more realistic scenarios. For algorithm selection in dynamic planning environments it is important to quantify the dynamism within different applications, therefore the extension of the degree of dynamism could be a reasonable way.

5 Summary and future work

In this article we have investigated the Multi Depot Vehicle Routing Problem with Time Windows (MDVRPTW) in static and dynamic scenarios. Different existing adaptive neighborhood search strategies were combined and guided by a tabu search to escape from local optima. As the presented approach only works on feasible solutions the solution quality of the best-know solutions computed by UTSA from Cordeau et al. [4] could not be reached. The results show an averaged deviation of 4.24 % from the solutions on the MDVRPTW-instances from Cordeau et al.

Our approach implements a repair mechanism and a real-time optimization, thus we can incorporate dynamic changes of existing customer information, new customer requests or cancellations. The results of these dynamic scenarios were evaluated by a c-competitive analysis. The number of events in the experiments varies from 12 to 180. This approach computes dynamic solutions that differ to a static computation in a maximum factor of $c < 1.8$. Thereby the response time is limited to only a few seconds. It was assumed that not only the number of events but also their appearance time influences the solution quality. According to Larsen [10] a degree of dynamism $\phi_{mdvrptw}$ was defined and factor c was opposed to $\phi_{mdvrptw}$. Further refinements for the definition of the degree of dynamism was presented.

To improve the solutions in a static scenario the neighborhood search could also explore the neighborhoods of unfeasible solutions to probably find better solutions. This extension is promising particularly for instances with small time windows where only a few permutations of customers form a feasible solution. The results for the dynamic scenarios could probably be improved if the real-time optimization also uses meta-heuristics. However, it should be regarded that the response time should be limited, this might be done by parallelization procedures.

References

1. O. Bräysy and M. Gendreau. Tabu search heuristics for the vehicle routing problem with time windows. *TOP*, 10(2):211–237, 2002.
2. W.-C. Chiang and R. A. Russell. A metaheuristic for the vehicle routing problem with soft time windows. *Journal of the Operational Research*, 55(12):1298–1310, December 2004.
3. G. Clarke and J. W. Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12(4):568–581, July-August 1964.
4. J.-F. Cordeau, G. Laporte, and A. Mercier. Improved tabu search algorithm for the handling of route duration constraints in vehicle routing problems with time windows. *Journal of the Operational Research*, 55(5):542–546, May 2004.
5. G. Dantzig and J. Ramser. The truck dispatching problem. *Management Science*, 6:80–91, 1959.
6. Wolfgang Domschke. *Rundreisen und Touren*, volume 2. Oldenbourg, München, 4 edition, 1997.
7. Fred Glover and Manuel Laguna. *Tabu Search.* Kluwer, Boston, 1997.
8. Tore Grünert and Stefan Irnich. *Optimierung im Transport - Wege und Touren*, volume 2. Shaker, Aachen, 2005.
9. Geir Hasle. Industrial vehicle routing. In *Modelling of Large Logistics Networks*, 2007.
10. A. Larsen. *The Dynamic Vehicle Routing Problem.* Phd. thesis, Technical University of Denmark, 2001.
11. I.H. Osman. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research*, 41:421–451, 1993.
12. D. Pisinger and S. Ropke. A general heuristic for vehicle routing problems. *Computers and Operations Research*, 34(8):2403–2435, 2007.
13. M. Polacek, R.F. Hartl, K. Doerner, and M. Reimann. A variable neighborhood search for the multi depot vehicle routing problem with time windows. *Journal of Heuristics*, 10:613–627, 2004.
14. M. Zachcial, A. Classen, and Thomas Selz. Gleitende Mittelfristprognose für den Personen- und Güterverkehr. Technical Report 96.0809/2004, Bundesministerium für Verkehr, Bau und Stadtentwicklung, Januar 2007.