TECHNISCHE
UNIVERSITÄT
DRESDEN

Fakultät Verkehrswissenschaften „Friedrich List“

DISKUSSIONSBEITRÄGE AUS DEM INSTITUT FÜR WIRTSCHAFT UND VERKEHR

NR. 1/2017
JÖRN SCHÖNBERGER

IMPLICIT TIME WINDOWS AND MULTI-COMMODITY MIXED-FLEET VEHICLE ROUTING

HERAUSGEBER: DIE PROFESSOREN DES
INSTITUTS FÜR WIRTSCHAFT UND VERKEHR
ISSN 1433-626X

In den Diskussionsbeiträgen aus dem Institut für Wirtschaft und Verkehr der TU Dresden erscheinen in zeitlich loser Folge verkehrswirtschaftliche Arbeiten von allgemeinem Interesse. Die Diskussionsbeiträge enthalten Vorträge, Auszüge aus Diplomarbeiten, interessante Seminararbeiten, verkehrswirtschaftliche Thesenpapiere, Übersichtsarbeiten, ebenso wie Beiträge, die zur Veröffentlichung in referierten Zeitschriften vorgesehen sind. Allen Beiträgen gemeinsam ist wissenschaftliche Fundierung und wissenschaftlicher Anspruch, jedoch je nach Zweck des jeweiligen Beitrages in unterschiedlichem Maße. Die in diesem Diskussionsbeitrag vertretenen Standpunkte liegen ausschließlich in der Verantwortung der Autoren und decken sich nicht zwingend mit denen der Herausgeber.

Als Herausgeber fungieren die Professoren des Instituts für Wirtschaft und Verkehr der TU Dresden.

manuscript No.
(will be inserted by the editor)

Implicit Time Windows and Multi-Commodity Mixed-Fleet Vehicle Routing

J. Schönberger

November 16, 2017

Abstract We report about a metaheuristic solving approach for a vehicle routing problem with a heterogeneous vehicle fleet. Customer locations require the visit of two different types of vehicles. Operation starting times at the customers are coupled by a specific time window constraint called implicit time window. An implicit time window limits the difference among the starting times of two different operations. At first glance, an implicit time window seems to be less restricting than an explicit time window. However, we exhibit that typical search mechanism originally developed for handling explicit time windows in metaheuristic search fail to meet the requirements of implicit time windows. We therefore propose additional solution manipulation techniques to achieve solution feasibility with respect to the implicit time windows. Within comprehensive computational experiments, we demonstrate the superiority of a memetic algorithm specifically equipped with these new search components over a memetic algorithm that deploys only constraint handling techniques for explicit time windows.

Keywords Time Windows · Memetic Search · Metaheuristic · Constraint Handling

1 Introduction

Transport plays an important role in today's value creation chains. Following the historic development the appreciation of manufacturing activities seems to be higher than the appreciation of supportive activities like warehousing and transportation. As a consequence, the negotiating position of transport service providers has been considered as minor compared to focal manufacturing supply chain manufacturing partners. Planning models for transport operations take this imbalance

---

J. Schönberger
Technische Universität Dresden, Faculty of Transportation and Traffic Sciences ́Friedrich List ̧
01062 Dresden, Germany
Tel.: +123-45-678910
Fax: +123-45-678910
E-mail: joern.schoenberger@tu-dresden.de

of power into account by considering external requirements like time windows in process planning. Famous appearances of this fact can be found in the fleet deployment models for the so-called vehicle routing problem (Golden et al., 2008).

During the last decade focal partners as well as service providing partners discovered that a more cooperative and respectful common planning can result in the realization of additional efficiency potentials. This article reports about a planning problem and its solution in which a focal supply chain partner and a transport service provider commonly look for transportation processes. The basic idea is to ensure that the service quality proposed by the focal company corresponds with the route execution costs of the trucking company.

The here investigated transportation scenario falls into the category of vehicle routing problems. In contrast to the traditional vehicle routing setup each of the spatially distributed customers requires the provision of goods from two different sources that cannot be consolidated into one vehicle. Therefore, each customer must be visited twice. Additionally, the two resulting unloading starting times must not differ by more than a given number of time units (*delivery time synchronization*). Therefore, a cross-vehicle scheduling constraint must be considered.

Section 2 delivers the introduction of the decision problem. The computational evaluation of a memetic algorithm adapted from an application of the vehicle routing problem with delivery time windows (Solomon, 1987) in Section 3 demonstrates a minor performance w.r.t. the synchronization constraint. To overcome this weakness, we propose the definition of search operators that specifically take care about the installation of the synchronization (Section 4). We report results from comprehensive computational experiments in Section 5.

2 The Vehicle Routing Problem with Two Commodities and Synchronization Constraints

2.1 Literature on Related Problems

The 2-CVRP-S combines several extensions of the classic vehicle routing problem variants. Toth and Vigo (2014) as well as Golden et al. (2008) survey contributions and research directions on the vehicle routing problem class. Problems in which individual pickup and/or locations are surveyed and classified by Parragh et al. (2008b) as well as Parragh et al. (2008a). In case that one of these basic decision tasks is enriched by complicated constraints, often found in real world applications, the resulting decision problem falls into the category of rich vehicle routing problems (Caceres-Cruz et al., 2014; Lahyani et al., 2015; Doerner and Schmid, 2010).

Vidal et al. (2012) write about models and algorithmic approaches on a broader class of vehicle routing problems with several depots. Crevier et al. (2007) report on a variant of the vehicle routing problem in which vehicles are replenished at different re-filling depots while being on route.

Cattaruzza et al. (2014) investigates the multi trip vehicle routing problem in which only small vehicles are allowed to enter a downtown area so that a regular return to a depot to refill the vehicle becomes necessary.

Salhi et al. (2014) extend the investigated vehicle routing setup by introducing several depots and assign customer requests to depots taking into account the type of vehicle available in each depot. Sometimes, a depot is selected for each individual request as pickup or delivery location (Schönberger et al., 2013) triggered only by the availability of goods.

In a CVRP with a heterogeneous fleet (Çağrı Koç et al., 2016) requires particular care. In such a setup, the fleet consists of different types of vehicles. On the one hand, some types of vehicles are excluded from serving some types of requests due to disclosure constraints (Ferland and Michelon, 1988) implied by different maximal load capacities or special equipment like cooling units. On the other hand, the contribution of a request-to-vehicle assignment decision to the objective function value depends of the selected vehicle, i.e. different vehicles cause different request fulfillment costs like energy consumption (Kopfer et al., 2014).

Loading constraints restrict the compilation of requests and considers the physical dimension(s) of individual loading units. A corresponding survey is provided by Pollaris et al. (2015). Differences between routing problems with only one and with several commodities are revealed by Archetti et al. (2016). The split of demand associated with a request in the context of vehicle routing is addressed by Archetti and Speranza (2008).

Synchronization refers to the coordination of coupled decisions affecting different requests or resources. Goel and Meisel (2013) propose a decision support tool for a vehicle deployment problem in which operations of different vehicles at different locations are coupled by a scheduling constraint. A general discussion of different types of synchronization constraints is contributed by Drexl (2012).

Time windows are an important mechanism to achieve a temporal coordination of transportation-related operations with upstream and/or downstream processes in the physical material flow. An explicit customer time window is a time period $[t_e, t_l]$ that specifies the part of the time axis in which an operation of a vehicle at a customer location is allowed to start. Basic fleet dispatching tasks incorporating time windows are the vehicle routing problem with time windows (VRPTW) investigated for example in Solomon (1987) and the pickup and delivery problem with time windows, for short PDPTW, (Mitrovic-Minic, 1998). In both fleet dispatching problems VRPTW as well as PDPTW one time window is specified for each loading and/or unloading location. Explicit time windows does not only ensure to comply with the maximal allowed operation starting time difference (*time difference property*). But an explicit customer time window also limits the absolute position of operation starting times along the time axis (*positioning property*).

The concept of using explicit time windows in vehicle routing is often inappropriate if the positioning property is not needed but only the time difference property of a time windows is required. de Jong et al. (1998), Favaretto et al. (2007), Breier (2014) as well as Breier and Gossler (2014) propose to append several explicit customer time windows to each customer site. This time windows are distributed over the time axis and the fleet dispatcher has to select one of these time windows (alternative time windows). However, beside the determination of the operation starting times of the considered operations it is necessary for a fleet planner to select the appropriate time window from the set of available time windows. This is a quite complicated task since a decision problem of enlarged complexity has to be solved. Three different kinds of interrelated decisions (routing as well as time window selection and scheduling decisions) must be made during the fleet dispatching process.

Both explicit time windows as well as multiple-time windows are problem exogenous data that have to be considered during the determination of the vehicle routes. Jabali et al. (2015) reports about self-imposed time windows. They derive time windows after they have determined vehicle routes and want to ensure that these time windows are respected even if travel time variations occur.

Schmid et al. (2009) investigate a fourth type of time windows applied in vehicle routing. They report about the consideration of a maximal difference between the loading and delivery time induced by delivering ready-mixed concrete. Here, only the time difference property has to be considered but the positioning of the time window on the time axis is not limited. Stieber et al. (2015) as well as Clausen (2011) propose the term "implicit time windows“ for a constraint that imposes only the time difference property. Schönberger (2015) has studied impacts of the variation of the length of implicit time windows on key process performance indicators in freight carrier planning.

2.2 Verbal Problem Outline

We consider a distribution scenario with two commodities A and B manufactured by a focal supply chain company. The two commodities must be distributed to spatially spread demanding sites. A trucking company is going to be hired for the fulfillment of the corresponding transport requests using its own fleet. A type-A-request requires the supply of a customer site with a given quantity of commodity A from a warehouse WH^A. Similarly, a type-B-request is defined. We define a request r as the quadruple $(p_r^+; p_r^-; c_r; t_r)$. The first two components determine the pickup location ($p_r^+ \in \{WH^A; WH^B\}$) as well as the delivery location (p_r^-). The third component c_r carries the quantity to be moved. The final component indicates the commodity to be moved, i.e. $t_r \in \{A; B\}$.

The transport of a commodity requires a specially fitted vehicle. These vehicles compose the fleet of the hired freight forwarder that is available to cover the supply demand at customer locations. Each of the n^A type-A-vehicles can be used to forward commodity A exclusively and each of the n^B type-B-vehicles is able to carry commodity B only. Carrying a mixture of commodity A as well as commodity B is impossible. There are no vehicles that can carry both types of commodities. The type-A-vehicles form the fleet $\mathcal{V}^{\mathcal{A}}$ but the type-B-vehicles form the fleet $\mathcal{V}^B$. The set $\mathcal{V} := \mathcal{V}^{\mathcal{A}} \cup \mathcal{V}^{\mathcal{B}}$ represents a heterogeneous fleet. Overall, this fleet comprises $n^{veh} := n^A + n^B$ vehicles. The maximal allowed payload of vehicle v is labeled as $C(v)$.

Transport demand to a customer order is expressed as an order. Formally defined, an order is a pair $(r_1; r_2)$ of two requests. Among them, r_1 is a type-A-request and r_2 is a type-B-request. In addition, the two delivery locations $p_{r_1}^-$ as well as $p_{r_2}^-$ coincide so that the customer site requires the visit of one type-A-vehicle and the visit of one type-B-vehicle. We collect all orders in the set $\mathcal{O}$.

The route of a type-A-vehicle starts at the depot of the fleet $\mathcal{V}$. First, warehouse WH^A is visited in order to load sufficient quantity of commodity A. Next, the loaded quantity is distributed among customers. If necessary, the vehicle returns to WH^A to replenish commodity A and to visit other customer sites. After the vehicle has visited the sites of all assigned customers it returns to the depot. Similarly, type-B-vehicles operate.

The earliest vehicle starting time is 0. All vehicles have to return to their depot not later than time MS^{max}.

If a customer demands quantities of both commodities A and B it is visited by two vehicles. Such a customer wants to coordinate the associated two unloading operations so that the two unloading operation starting times differ not more than Δ time units. There are several reasons to impose this scheduling constraint:

- unloading setup efforts can be avoided/reduced if both vehicles can be unloaded close together with the same equipment and/or staff
- scheduling the unloading operations by the receiver is easier if both unloading operations can be considered as one operation since there is only a maximal pre-determined idle time between these two operations
- downstream processes require the availability of both commodities within the given maximal time difference at the beginning of a process (especially in process industry setups)

Independently from the motivating reason the maximal time gap requirement imposes a scheduling requirement that prevents the decomposition of the routing problem into a type-A-routing problem as well as into a type-B-routing problem. This requirement establishes a logical dependency between the scheduling of type-A-operations and type-B-operations. A vehicle routing problem must be solved for the complete fleet $\mathcal{V}$.

After having received all relevant requests the trucking company determine routes for the vehicles so that the total sum of travel distances across the fleet is minimized. Every request is assigned to a vehicle of the compatible type (C_1). Both operations of a request are assigned to the same vehicle and the pickup operation precedes the associated delivery operation (C_2). The maximal operation starting time difference is kept for each order (C_3). The vehicle capacity is not exceeded at any stage of the route of a vehicle (C_4). Each vehicle route must be completed not later than time MS^{max} (C_5). We call this variant of the CVRP the **two-commodity capacitated vehicle routing problem with synchronization (2-CVRP-S)**.

Solving the 2-CVRP-S enables the trucking company to appropriately calculate the distribution costs if a given *Delta*-value is preserved. Furthermore, they can quantify cost variations induced by a variation of the requested service level demanded by the focal supply chain partner and expressed by Δ. Both partners, the trucking company as well as the focal partner can now find a commonly an accepted trade-off between service level Δ and the corresponding payment to the transport service provider.

2.3 Implicit Time Windows

The 2-CVRP-S exhibits similarities with the pickup and delivery problem (Parragh et al., 2008b,a) since requests may have different pickup locations and individual delivery locations. We re-use the mixed-integer linear program formulation proposed for this problem given in (Schönberger, 2015) to formally represent the 2-CVRP-S. In this subsection we focus on the coordination requirement imposed to keep the two visits at a customer site sufficiently close together.

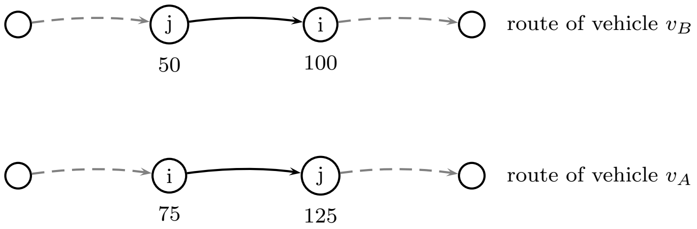

Fig. 1 Two route with unsolvable implicit time window conflicts

At first glance, it seems that the 2-CVRP-S is similar to a vehicle routing problem with explicitly specified customer time windows. However, adequate values for the opening time t_e and the closure time t_l are not given in the 2-CVRP-S as problem parameter. Furthermore, it is not necessary to restrict the position of the operation starting times on the time axis before the vehicle routes are tentatively established. The positioning property of an explicit time window is not needed in the 2-CVRP-S. Therefore, explicit time windows are not an adequate way to integrate the maximal operation starting time requirement into the model of the 2-CVRP-S.

We have decided to formulate the following two constraints (1) and (2) that ensure the consideration of the maximal operation starting time difference Δ. The starting time of an (un)loading operation at node i of vehicle v is represented by the continuous decision variable $st_{i,v}$.

$$st_{r^-_{o,a},v_1} - st_{r^-_{o,a},v_2} \leq \Delta + (2 - y_{r_{o,a}v_1} - y_{r_{o,a}v_1}) \cdot MS^{max} \qquad \forall v_1, v_2 \in \mathcal{V}, o \in \mathcal{O} \tag{1}$$

$$st_{r^-_{o,b},v_1} - st_{r^-_{o,b},v_2} \leq \Delta + (2 - y_{r_{o,b}v_1} - y_{r_{o,b}v_1}) \cdot MS^{max} \qquad \forall v_1, v_2 \in \mathcal{V}, o \in \mathcal{O} \tag{2}$$

We can understand the two constraints (1) and (2) as the realization of a movable or implicit time window of length Δ into which both operations to be performed at a customer site must fall. Implicit time windows provide the difference restricting property, which is required here, but not the positioning property of an explicit time window, which is redundant here and too restrictive. However, since the position of the implicit time window is not part of the fixed problem data it is necessary to determine this position during the solving of the fleet dispatching decision task. Therefore, implicit time windows increase the already quite high complexity of the fleet dispatching task by adding a further decision problem component.

The insertion of waiting or idle periods prior to the start of an (un)loading operation at a customer site is a commonly used approach to minimize time window infeasibility in vehicle routes. In case of an explicit customer time window, waiting heals time windows infeasibility only in case that the corresponding vehicle arrives too early at a customer site. A too late arrival can be healed only by modifying the sequence of previously visited locations.

In the 2-CVRP-S an insertion of waiting periods is not appropriate in some situations to achieve feasibility w.r.t. implicit time windows. Here, the revision of routing decisions becomes necessary to meet the implicit time windows defined at customer sites. We consider the small example shown in Fig. 1. Parts of the route of the type-A-vehicle v_A (lower route) as well as of the type-B-vehicle v_B (upper route) are shown. Both vehicles serve the two customers i as well as j but in different sequences. The numbers below the nodes indicate the earliest possible unloading starting times at the nodes. If we assume that $\Delta = 50$ then the implicit time window constraint is violated for customer i but not for customer j. Letting vehicle v_B wait for 50 time units heals the violation of the implicit time window constraint at node j. Since also all downstream nodes after i in the route of vehicle v_B are also shifted to the right, we get a violation of the implicit time windows constraint now at node j. The only way out of this conflict is to change the visiting sequence in one of the two routes. A route revision becomes necessary. This small example demonstrated the connectivity of the type-A-routing problem with the type-B-routing problem.

2.4 Test Cases

A suite of parameterized test cases is defined for a computational evaluation of the recently introduced 2-CVRP-S. The subsequently outlined general setup applies for all these test cases and comprises a fleet of 10 vehicles. This fleet contains five identical type-A vehicles and five identical type-B vehicles. The fleet operates in the area represented by the square $[-300; 300] \times [-300; 300]$. Initially, all vehicles are positioned at the trucking company's depot at point $(0; 0)$. Each vehicle finally returns to this location after it has fulfilled all assigned operations.

There are two warehouses available. Each warehouse stores one commodity only. Commodity A is stocked only at warehouse WH^A which is located at $(-150; 150)$. Commodity B is exclusively available at warehouse WH^B located at $(200; -50)$.

Overall, 25 different customer locations randomly drawn from the area $[-300; 300] \times [-300; 300]$ must be supplied with commodity A as well as commodity B. The corresponding supply demand is coded into 25 orders. Each order comprises two origin-to-destination requests (od-requests). Each od-request requires the pickup of one of the commodities A or B at the corresponding warehouse and the delivery of this commodity to an individual customer site. The two od-requests contained in an order fulfill the following two properties. First, their delivery locations coincide. Second, the first od-request requires the delivery of the type-A commodity but the second od-request is associated with a type-B commodity. Consequently, each of the 25 customer sites must be visited twice: once by a type-A vehicle and once by a type-B vehicle. We randomly draw five different sets of 25 customer locations using five different random number generator seeding values $\alpha \in \{0; 1; 2; 3; 4\}$.

We assign a demand of 1 quantity unit of each commodity to each customer. But each vehicle can carry up to 100 quantity units so that the vehicle capacity is not scarce. We choose this setting in order to be able to focus on the impacts of the variation of time-related data on route compilation.

The fleet dispatcher has to setup a set of routes with a minimal travel distance sum for the fleet so that all 50 od-requests contained in the 25 orders are served. We are going to analyze the impacts of varying the maximal allowed difference Δ between the two visits to be scheduled by the fleet dispatcher at each customer site. We assume the same implicit time window length for all customer sites. Three different situations are distinguished. If $\Delta = \infty$ then there is no coordination of the two visits necessary. We have found out in preliminary experiments that an arrival time difference larger than 500 time units must be implemented for some delivery locations if the travel distance sum over all vehicle routes is minimal. In a second experiment we therefore limit the maximal allowed unloading starting time difference at each customer site to $\Delta = 500$ time units. This enforces the fleet dispatcher to revise the least distance route set in order to fulfill the requirements of the implicit time windows of length $\Delta = 500$ time units at each customer site. Finally, in a third experiment, we want to analyze the impacts of enforcing the two unloading operations to the same starting time ($\Delta = 0$). In total, we analyze the three implicit time window length settings $\Delta = \{\infty; 500; 0\}$.

With the intention to keep the total travel distances as short as possible a fleet dispatcher would preferentially apply a waiting strategy to achieve the implicit time window feasibility of the generated vehicle operation schedules. The insertion of waiting times at customer sites often implies a prolongation of the makespan, which is the time span between the leaving of the first vehicle from the depot and the return of the last vehicle to the depot. The specification of a maximal allowed makespan MS^{max} let the insertion of excessive waiting periods become impossible. That is, the fleet dispatcher has to revise some routes in order to avoid any exceeding of MS^{max}. In the aforementioned preliminary experiments we have seen that the maximal makespan without any time-related operation starting restriction is larger than 3000 time units. The reduction of MS^{max} from (the referential value) of ∞ time units to 3000 time units let a route set revision become necessary in order to comply with this the maximal allowed makespan. We investigate scenarios with the maximal allowed makespan values $MS^{max} \in \{\infty; 3000; 2000\}$.

In summary, following the aforementioned ideas, we setup $\|\{0; 1; 2; 3; 4\}\| \cdot \|\{\infty; 500; 0\}\| \cdot \|\{\infty; 3000; 2000\}\| = 5 \cdot 3 \cdot 3 = 45$ different fleet dispatching (or: vehicle routing) scenarios. For each of the 5 customer location sets we can have 9 different time-oriented limitation sets resulting from the combination of different maximal makespan values with different values for the length of the implicit time windows at the customer sites.

3 A Memetic Algorithm Approach to the 2-CVRP-S

Vehicle routing problems with explicit or implicit time window require two types of decisions. First, routing decisions are required. Second, scheduling decisions are needed. The first mentioned decisions determine the movement of vehicles in the space while the second mentioned decisions determine the behavior of vehicles over the time. Of course, both types of decisions are interdependent, i.e. routing as well as scheduling decisions cannot be made independently. Typically, tentative routing decisions are made first. In a subsequent step, the scheduling decisions are made, i.e. waiting times are inserted at customer locations in order to meet operation starting time windows. In case that unsolvable time window conflict are detected during the schedule determination, previously made routing decisions are revised leading to an updated set of vehicle routes.

We have seen in the previously reported experiments that even small instances of the proposed analytic decision model require huge computational resources (Schönberger, 2015). Therefore, the setup of a heuristic search algorithm for the identification of high quality approximation solutions of the model is reasonable in order to determine high quality solutions for larger problem instances.

Since we are not aware of any straight forward neighborhood structure that preserves feasibility with respect to the numerous constraints and which takes care of the cross-route schedule coordination implied by the implicit time windows, we propose to apply a structured random sampling of the search space. This sampling is iterated by a genetic algorithm towards high quality model solution approximations. The genetic search addresses the determination of a set of routes (*plan*). We use a direct path representation, e.g. we use a multi-chromosome representation for a plan (Schönberger (2005)). A straightforward scheduling procedure with postponement (waiting time) options is incorporated that explicitly addresses the need for a cross-route operation starting time coordination during the schedule determination. This leads to the following constraint handling concept implemented in the proposed heuristic. The precedence constraint (C_1) as well as the capacity constraint (C_2) and the area consistency constraint (C_3) are syntactically preserved. We propose a construction procedure that takes care about the fulfillment of these three constraints and all subsequently applied operators preserve the solution feasibility with respect to (C_1), (C_2) as well as (C_3). Infeasibility with respect to (C_4) as well as (C_5) is temporarily accepted but penalized during the evolutionary search process. A population management scheme is used that prefers those solution proposals with few or even no violations of (C_4) or (C_5).

3.1 Plan Construction Procedure

Let *op* be an operation. An attribute *attribute* of this operation is labeled by *op.attribute*. We use the following attributes to fully describe operation *op*: *op.at* (arrival time of a vehicle serving this operation), *op.st* (starting time), *op.servicetime* (duration of service), *op.ct* (finishing/completion time of operation *op*), *op.lt* (leaving time of a vehicle assigned to this operation), *op.req* (reference to a request), *op.type* (pickup or delivery operation), *op.prec* (reference to the direct predecessor if *op* is contained in a route) and *op.next* (direct successor). The attribute *op.order* is used to link an operation with the donating order. In case that *op* is associated with the collection request of *op.order* then *op.reqtype* $:= COLLECTION$ and in case that OP is associated with the corresponding distribution request of *op.order* it is *op.reqtype* $:= DISTRIBUTION$.

A route *trip* is defined by two attributes which are both references: to a dummy start operation *trip.start* and to a dummy terminating operation *trip.stop*. These two operations are concatenated by setting $trip.start.next = trip.stop$ and $trip.stop.prev-ious = trip.start$. Since the dummy start operation has no predecessor operation, we set $trip.start.previous = NULL$ and since the dummy terminating operation *trip.stop* has no succeeding operation, we set $trip.stop.next = NULL$. The route of an unused vehicle consists only of these two dummy operati-

```
(a) PROCEDURE PLANCONSTRUCT($P$, $ReqPerm$)
(b)   initialize_plan(P);
(c)   for $i = 1$ to $m$
(d)     initialize_route($P.trip_i$);
(e)   for $r = 1$ to $n$
(f)     $CurReq = ReqPerm_r$;
(g)       select a vehicle $v$ at random that fulfills area compatibility constraint ($C_3$);
(h)       generate operations belonging to $CurReq$ and store them in $A(v)$;
(i)   next $r$;
(j)   for $i = 1$ to $m$
(k)     generate a permutation $p$ of operations in $A(i)$ at random feasible w.r.t ($C_1$);
(l)     insert operations from $A(i)$ into $P.route_i$ in the sequence determined by $p$;
(m)     repeat
(n)       identify request $r^*$ belonging to first pickup operation where
                 a capacity exceeding is detected;
(o)       move both operations belonging to $r^*$ to the begin or
                 to the end of $P.route_i$ (random decision);
(p)     until (feasibility w.r.t. ($C_2$) is achieved);
(q)   next $i$;
(r) end;
```

Fig. 2 Pseudo-code for the plan generation procedure

ons. All routes form a plan. Therefore, a plan p consists of an array $p.route_1, ..., p.route_m$ of references to the routes of the available m vehicles. Let $trip_i$ denote the route of vehicle i then $p.route_i = trip_i$.

The proposed plan generation procedure does not aim at identifying plans with a quite low objective function value but it is intended to re-use this procedure to generate a collection of quite diverse individuals dispersed over the search space. Feasibility with respect to (C_1)-(C_3) is guaranteed but feasibility with respect to (C_4) as well as (C_5) is not guaranteed. The plan generation is controlled by a permutation of the available requests, e.g. calling the procedure with two different permutations should generate two different plans.

The pseudo-code of the plan generation procedure is shown in Fig. 2. First, an empty plan P is initialized (b) and all needed m routes containing only the dummy start and end operations are added to the plan (c)-(d). Second, all n requests are randomly distributed among the available m vehicles (e)-(i). Each loop execution starts with the identification of the next request to be assigned to a vehicle (f). A vehicle that operates in the region of the currently considered request is randomly selected (g). Third, the visiting sequences are determined for all m routes (j)-(q). A random sequence respecting the precedence constraint is instantiated (k) and the operations assigned to vehicle v are consecutively inserted into the route $P.route_v$ in the sequence determined by the proposed permutation (v). The proposed permutation is modified until no capacity exceeding is detected anymore (m)-(p). In case that such a capacity exceeding is detected, the first operation leading to this exceeding (this must be a pickup operation) is identified and the associated request is fetched (n). In order to heal the capacity exceeding it is arbitrarily decided whether the two operations belonging to this request are positioned at the beginning or at the end of the permutation (o). After the capacity exceeding has been solved, the next route is determined. Finally, the generated solution proposal (plan) is feasible with respect to (C_1), (C_2) as well as (C_3).

```
(a)  PROCEDURE POSTPONEMENT($P$; VEHSEQ);
(b)    for $v = 1$ to $m$;
(c)      $op := P.route[VEHSEQ[v]].start$;
(d)      $OP.at := 0.0$;
(e)      $OP.st := 0.0$;
(f)      $OP.ct := 0.0$;
(g)      $OP.lt := 0.0$;
(h)      $OP := OP.next$;
(i)      while( $OP.next \neq NULL$ )
(j)        $OP.at := OP.previous.lt$
                 $+ TravelTime[OP.previous][OP]$;
(k)        $OP.st := OP.at$;
(l)        if ($OP.type = DELIVERY$ )
(m)          $\bar{r}$ := associated_request(OP.order,OP.req);
(n)          if ($OP.order.\bar{r}.delivery\_operation$ already scheduled)
(o)            if ($OP.order.\bar{r}.delivery\_operation.st - OP.st > \Delta^{max}$)
(p)              $OP.st := OP.order.\bar{r}.delivery\_operation.st - \Delta^{max}$;
(p)            end;
(p)          end;
(p)        end;
(s)        $OP.ct := OP.st + OP.duration$;
(t)        $OP.lt := OP.ct$;
(u)        $OP := OP.next$;
(v)      end while;
(w)    next $v$;
```

Fig. 3 Scheduling procedure with postponement

3.2 Inter-Route Operations Scheduling with Postponement

Operation scheduling refers to the determination of the operation starting times in the routes of a plan P. Typically, operations are scheduled to start as early as possible in order to prevent unproductive idle times of machines and staff (*left-to-right scheduling* or *forward scheduling*). In the context of vehicle routing, a pickup or a delivery operation starts immediately after the corresponding vehicle has arrived from the previously visited location.

The application of the left-to-right scheduling procedure $SCHEDULE$ determines the earliest arrival, starting, finishing as well as leaving times along a given route. Waiting times are not inserted at any position in the route. However, the necessary inter-route coordination of starting and completion times imposed by the implicit time window constraints (1)-(2) requires the consideration of scheduling decisions made for operations contained in previously scheduled routes. Hence, the sequence in which the routes $PLAN.route_1, ..., PLAN.route_m$ are scheduled is important since previously determined operation starting times of operations impose implicit time windows for the associated so far unscheduled operations. In order to prevent operations from starting to early, we propose the following enhancement of the left-to-right schedule determining approach. Let $VEHSEQ$ denote the sequence in which the vehicle routes are determined (permutation of the vehicle fleet members).

The procedure $POSTPONEMENT$ determines the vehicle schedules in the sequence coded in $VEHSEQ$. Initially, the next route to be processed is selected (c). The four time parameters are set to 0 (d)-(g) and the next operation from the currently considered route is selected (h). As long as the selected operation is not the last operation is the currently considered route the loop (i)-(v) is iteratively repeated. The earliest possible arrival time at the customer location is set (j) and the operation starting time is tentatively set to the arrival time (k). In case that the current operation is a delivery operation (l) the second request belonging to the same order as the request to which is current operation belongs is identified (m). If the delivery operation of this request is already determined(n) it is checked if the postponement of the operation starting time is useful in order to avoid violation of the implicit time window at the customer site of this operation (o). In case that the postponement is beneficial then the postponement is established (p). The operation completion time is set (s) and the leaving time of the vehicle is fixed (t). Going to the next operation is the considered vehicle route terminates the iteration (u).

In order to apply the proposed procedure $POSTPONEMENT$ it is necessary to specify a vehicle sequence $VEHSEQ$ to control the schedule building. No operation starting time coordination is necessary among routes of a certain commodity type. But routes of two vehicles of different type might both contain the two delivery operations of an order. Here, the sequence in which routes are scheduled affects the schedule building. Whenever the $POSTPONEMENT$-procedure has fixed an operation starting time in one route then this starting time cannot be revised in this procedure call. In order to analyze those characteristics that are important for achieving implicit time window feasibility, we test the following four different strategies to determine the vehicle scheduling order $VEHSEQ$ in order to prepare the next execution of $POSTPONEMENT$.

- Sort by Number (SBN): This is the default vehicle order sequence. Here, the vehicles are sorted by their id. Since the vehicle id remains unchanged during the MA execution, the same vehicle sequence is used for all calls of the $POSTPONEMENT$-procedure.
- Random Vehicle Sequence (RVS): For each call of the scheduling procedure a random vehicle sequence is drawn. This strategy is used to see if any deviation from the default strategy (SBN) has an impact with regard to the obtained solution quality.
- Appended Vehicle Sequence (AVS): We make the vehicle scheduling sequence a part of the decision problem. A vehicle routing sequence is appended to each individual in the maintained population and this key is used to decode the partial individual comprising the routes to a full individual comprising routes and scheduling decisions. The appended vehicle sequence is a permutation of the fleet and this permutation is varied during the population evolution. Whenever two individuals are recombined the offspring individual contains a recombination of the two parental vehicle sequences. We use the PPX (Bierwirth et al. (1996)) permutation crossover operator. The mutation of an individual also varies the appended vehicle permutation. The idea behind AVS is to identify adequate combinations of route sets and vehicle sequences that together identify high quality vehicles routes and schedules.
- Busiest Vehicle First (BVF): We sort the vehicles in decreasing order by their estimated route duration. The estimated route duration is the duration of the route without any inserted waiting times. The ratio of applying BVF is that the route with the longest estimated duration contains operations that are start very late. If the starting times of these late operations are determined

early during the consecutive scheduling procedure then the associated delivery operation belonging to the second request of an order get a quite high change to fulfill the implicit time window constraint, possibly after the insertion of a waiting period.

3.3 Population Model

In order to make different plans comparable, we evaluate each plan POP_i with respect to (i) the sum of makespan exceeding in all routes $E^{MS}(POP_i)$ (ii) the sum $E^{SYNC}(POP_i)$ of exceeding of the implicit time window constraints and (iii) the objective function value $E^D(POP_i)$, which is the sum of the travel distance of all routes. The exceeding of an implicit time window constraint at a customer site is defined as 0 if the implicit time window is fulfilled by the two determine operation starting times. In case that the operation starting time difference exceeds Δ then the exceeding is defined as the starting time difference minus Δ.

$$POP_{i_1} \succ POP_{i_2} \Leftrightarrow \begin{cases} E^{MS}(POP_{i_1}) \leq E^{MS}(POP_{i_2}) \text{ or} \\ \\ E^{MS}(POP_{i_1}) = E^{MS}(POP_{i_2}) \\ \wedge E^{SYNC}(POP_{i_1}) \leq E^{SYNC}(POP_{i_2}) \text{ or} \\ \\ E^{MS}(POP_{i_1}) = E^{MS}(POP_{i_2}) \\ \wedge E^{SYNC}(POP_{i_1}) = E^{SYNC}(POP_{i_2}) \\ \wedge E^{D}(POP_{i_1}) \leq E^{D}(POP_{i_2}) \end{cases} \tag{3}$$

We define plan POP_{i_1} to be superior to another plan POP_{i_2} if $POP_{i_1} \succ POP_{i_2}$ as defined by (3). A plan POP_{i_1} dominates another plan POP_{i_2} if $E^{MS}(POP_{i_1}) \leq E^{MS}(POP_{i_2})$.

In case that both proposals have a common makespan evaluation then POP_{i_1} dominates POP_{i_2} if and only if $E^{SYNC}(POP_{i_1}) \leq E^{SYNC}(POP_{i_2})$. In $E^{SYNC}(POP)$ all exceeding of the TT^{min}-values, TT^{max}-values as well of OD^{max}-thresholds found in the solution proposal POP are stored. If both individuals exhibit the same $E^{MS}(\dot{})$-value as well as the same $E^{SYNC}(\cdot)$-value then POP_{i_1} dominates POP_{i_2} if and only if the traveled distances in POP_{i_1} are less than the traveled distances in POP_{i_2} which is equal to $E^D(POP_{i_1}) \leq E^D(POP_{i_2})$.

3.4 Hill Climbing and Constraint Violation Repair

Fig. 4 shows the hill-climbing procedure applied to each generated solution proposal in order to reduce the number of constraint violations w.r.t to (C_4) and (C_5) while feasibility w.r.t. (C_1) as well as (C_2) is preserved. Furthermore, this procedure tries to reduce the total travel distance sum (the objective function value). Due to the interdependency between operation sequences, capacity requirements and scheduling decision the hill-climber cannot guarantee to return a feasible plan w.r.t. (C_3)-(C_5). While the genetic search space sampling preferentially explores the search space, we use this procedure to exploit the neighborhood of any solution proposal identified by the genetic search space sampling.

```
(a) Function HILLCLIMBER($PLAN$, $VEHSEQ$)
(b)   POSTPONEMENT($PLAN$,$VEHSEQ$);
(c)   CAP_REPAIR($PLAN$);
(d)   MAKESPAN_REPAIR($PLAN$);
(e)   2-OPT($PLAN$);
(f)   POSTPONEMENT($PLAN$,$VEHSEQ$);
(g)   return $PLAN$;
(h) end;
```

Fig. 4 Hill-Climbing and Repair Procedure

First, a tentative operation scheduling is made (b). Second, (C_4) is addressed and the routes are scanned for a capacity exceeding. As soon as such an exceeding is detected (on an arc originating from a pickup operation) this pickup operation and the associated delivery operation are both shifted to the end of the route (c). Third, with the goal to reduce the number of routes that exceed the maximal allowed route duration (C_5) the procedure $DURATION\_REPAIR(PLAN)$ is called (d). This procedure executes the following steps until no further exceeding of the maximal allowed route duration is possible. (i) all route durations are determined. (ii) routes are sorted by duration in a decreasing order (iii) from the top-ranked route randomly selected requests are shifted into other randomly selected routes until the top-ranked route does not exceed the maximal allowed duration any more. The target route is selected so that no maximal duration exceeding is achieved after the insertion. If such a target route is unavailable, then the procedure $DURATION\_REPAIR(PLAN)$ is stopped. As soon as the shifting of a request solves the maximal duration exceeding the procedure jumps to step (i) again. Fourth, it is tried to reduce the total travel distance by applying the 2-opt heuristic to each individual route contained in the proposal $PLAN$ (e). Finally, the updated route set is processed by the schedule determining procedure $POSTPONEMENT$ (f) and the updated $PLAN$ is returned (g).

3.5 Genetic Search Procedure Overview

A memetic algorithm (MA) incorporates a genetic search framework and a local search procedure. Fig. 5 shows the here used MA. It deploys a $\lambda + \mu$ population model (Grefenstette, 2000) to evolve a population of $PopSize$ plans $POP_1, \ldots, POP_{PopSize}$ over several iterations until the termination criteria $TermCrit$ is fulfilled. In each iteration $PopSize$ offspring plans are generated from the existing $PopSize$ parental plans. The set of parental plans is merged with the set of offspring proposals into a temporary set of $2 \cdot PopSize$ plans and the $PopSize$ highest evaluated plans (according to $\succ$) form the next population of plans.

A set of $2 \cdot PopSize$ plans is created in the population construction phase (b)-(h). The generation of plan POP_i starts with the determination of a random request permutation (c). This request permutation controls the construction of the routes (d). Next, a vehicle permutation is determined (e) that is forwarded to the hill climbing procedure (f). After all $2 \cdot PopSize$ plans have been generated and processed by the hill climbing procedure, the set of plans is sorted according to $\succ$ and re-numbered, e.g. POP_1 is now the highest valued plan (h).

The plans $POP_{PopSize+1}, \ldots, POP_{2 \cdot PopSize}$ are iteratively updated and replaced by $PopSize$ new plans (i)-(ae) until $TermCrit$ is fulfilled. First, the so

```
(a)  Function MEMETIC_ALGORITHM($PopSize$, $p_{xo}$, $p_{mut}$, $TermCrit$)
(b)    for $i$=1 to $2 \cdot PopSize$;
(c)      $ReqPerm$ := generate random permutation of requests;
(d)      PLANCONSTRUCT($POP_i$, $ReqPerm$);
(e)      $VehSeq$ := determine_vehicle_sequence();
(f)      HILLCLIMBER($POP_i$, $VehSeq$);
(g)    next $i$;
(h)    evaluate and sort array $POP$ according to $\succ$, renumber array elements;
(i)    repeat
(j)      $POP_{PopSize+1}$ := copy($POP_1$);
(k)      for $i$=2 to $PopSize$;
(l)        $p$ := random value from interval $[0; 1]$;
(m)        if $p \leq p_{xo}$ then
(n)          $i_1^* $ = roulette_wheel_selection from set $\{1, 2, \ldots, PopSize\}$;
(o)          $i_2^* $ = roulette_wheel_selection from set $\{1, 2, \ldots, PopSize\}$;
(p)          $POP_{PopSize+i}$ := cross_over($POP_{i_1^*}$,$POP_{i_2^*}$);
(q)        else
(r)          $i^*$ = random value from set $\{1, 2, \ldots, PopSize\}$;
(s)          $POP_{PopSize+1}$ := copy($POP_{i^*}$);
(t)        end if;
(u)        $p$ := random value from interval $[0; 1]$;
(v)        if $p \leq p_{mut}$ then
(w)          mutate($POP_{PopSize+i}$);
(x)        end if;
(y)      next $i$;
(z)      for $i$=$PopSize$ to $2 \cdot PopSize$;
(aa)       $VehSeq$ := determine_vehicle_sequence();
(ab)       HILLCLIMBER($POP_i$, $VehSeq$);
(ac)     next $i$;
(ad)     evaluate and sort array $POP$ according to $\succ$, renumber array elements;
(ae)   until $TermCrit$ is fulfilled;
(af)   return $POP_1$;
(ag) end;
```

Fig. 5 Pseudo-code of the Memetic Algorithm

far best found plan is saved and put in the temporary population (j). Second, the remaining $PopSize - 1$ offspring plans are created using recombination, mutation and duplication (k)-(y). With probability p_{xo} the next offspring is created by recombination (n)-(p) two parental plans are draw from the parental population using proportional roulette wheel selection (n)-(o) and the offspring plan is generated by a recombination operator (p). In all other cases a parental plan is selected by proportional roulette wheel selection (r) and copied into the temporary population (s). Each generated offspring plan is randomly varied with probability p_{mut} (u)-(y). Third, all generated offspring plans are evaluated and the required scheduling decisions are made in the hill climbing procedure (z)-(ac). An iteration terminates with the sorting of the temporary population according to $\succ$ (ad). The MA returns the best found plan as solution as soon as the termination criterion has been fulfilled (af).

The incorporated rank-based selection scheme ensures that individuals with the smallest constraint violation sum are preferentially transferred into the next population. Applying this reproduction scheme first eliminates the individuals with makespan exceeding, next the synchronization constraint violations are remedied and finally, the travel distance is minimized.

3.6 Search Operators

The search trajectories are evolved by interchanging information among two search trajectories (cross-over) and by randomly varying individual search trajectories (mutation). Both operators vary assignments of requests to vehicle(s) as well as operation sequences in a route. However, the offspring are feasible w.r.t. (C_1) and (C_3). The parental routes are consecutively recombined using the mppx-operator (Schönberger, 2005) for cross-over operations. If an offspring is mutated then one of the following plan modification steps is selected randomly and applied to the plan:

1. A non-empty route is selected at random. A sub-route of this selected route is arbitrarily labeled (including both the associated pickup and delivery operations of the affected requests). The labeled operations are shifted to another randomly selected route where all labeled operations are inserted between two randomly selected existing operations without varying their sequence.
2. A non-empty route is selected at random. In this route an operation is selected at random. This selected operation is arbitrarily re-positioned into the selected route but the precedence constraint feasibility (C_1) remains.
3. A non-empty route is selected at random. In this route both operations associated with a request served in this route are labeled, moved to another randomly selected route where they are inserted randomly so that (C_1) as well as (C_3) are respected.
4. The longest route in the plan is selected. All requests are shifted away from this route and are inserted at different randomly selected other routes. Again, (C_1) as well as (C_3) are respected.
5. An arbitrarily selected sub-route of a randomly chosen non-empty route is inverted but (C_1) remains considered
6. An arbitrarily selected sub-route of a randomly chosen non-empty route is shifted inside the donating route (C_1 remains considered).

3.7 Initial Computational Experiments

The proposed MA is applied to the 45 instances proposed in Subsection 2.4. Each instance is processed by four different configurations of the memetic algorithm resulting from the four different $VEHSEQ$-policies leading to $45 \cdot 4 = 180$ individual experiments. Since the memetic algorithm is a randomized search procedure each experiment is repeated with 5 different random number seeding values so that $180 \cdot 5 = 900$ individual experiments are executed.

The MA maintains a population of 250 individuals. An elite strategy is incorporated and the highest ranked individual is copied into the intermediate population without any modification. From preliminary experiments we learned that best results are observed if the crossover probability as well as the mutation probability are set to 100% each (for the non-elitist population members). The genetic search trajectory evolution terminates as soon as the average objective function value in the population is not improved for 20 consecutive iterations.

Tab. 1 summarizes the observed infeasibility associated with the implicit time windows. Feasibility w.r.t. the implicit time windows is achieved in all cases even

Table 1 Sum of implicit time window exceeding (number of exceeded time windows)

		$\mathit{\Delta}$	
	MS^{max}	500	0
SBN	∞	0 (-)	20.9 (0.2)
	3000	0 (-)	98.9 (0.6)
	2000	0 (-)	282.6 (2.1)
RVS	∞	0 (-)	0.5 (0.1)
	3000	0 (-)	78.9 (0.4)
	2000	0 (-)	229.3 (1.7)
AVS	∞	0 (-)	0 (-)
	3000	0 (-)	107.9 (0.6)
	2000	0 (-)	237.3 (1.8)
BVF	∞	0 (-)	3.6 (0.1)
	3000	0 (-)	37.4 (0.2)
	2000	0 (-)	29.4 (0.3)

if the length of the implicit time window is limited to $\mathit{\Delta} = 500$. If the implicit time window length is reduced to $\mathit{\Delta}^{max} = 0$ and if the route duration is limited then infeasibility w.r.t. implicit time windows cannot be prevented in average independently from the vehicle sequencing policy. However, if the BVS-policy determines the vehicle scheduling sequence then this remaining sum of exceeding of the implicit time windows is minimal. Also the average number of violated implicit time windows is significantly less compared to SBN, RVS as well as AVS.

The upper plot in Fig. 6 shows the best identified solution for the instance $\alpha = 4$, $MS^{max} = 2000$ without implicit time windows ($\mathit{\Delta} = \infty$). Different arc styles represent the paths of the deployed vehicles. The red routes represent the travel paths of the type-A-vehicles. Blue routes represent travel paths of the deployed type-B-vehicles. A total travel distance of 6142.27 length units is necessary but no makespan exceeding is observed. After the reduction of $\mathit{\Delta}$ to 0 the best found route set (lower plot in Fig. 6) looks completely different from the best solution identified in the situation without implicit time windows although still 4 vehicles (2 of each type) are deployed. However, the sum of travel distances is prolonged by 7.3% to 6591.61 length units.

Fig. 7 presents an instance in which the implicit time window feasibility cannot be reached after the reduction of the implicit time window length from ∞ (upper plot) to 0 (lower plot). Although the number of deployed vehicle is increased from 4 (2 type-A-vehicles & 2 type-B-vehicles) to 9 (5 type-A-vehicles % 4 type-B-vehicles) one implicit time window is violated and the operation starting time difference is 85.98 time units too large. While the best solution identified for the situation without implicit time windows requires to travel 6816.4 distance units this length is nearly doubled to 13111.18 distance units if $\mathit{\Delta} = 0$. The search operated and hill climbing procedure incorporated by the MA fail to eliminate all implicit time window constraint violations. In the remainder of this article we report about the development of additional hill climbers that directly address this specific type of constraints.

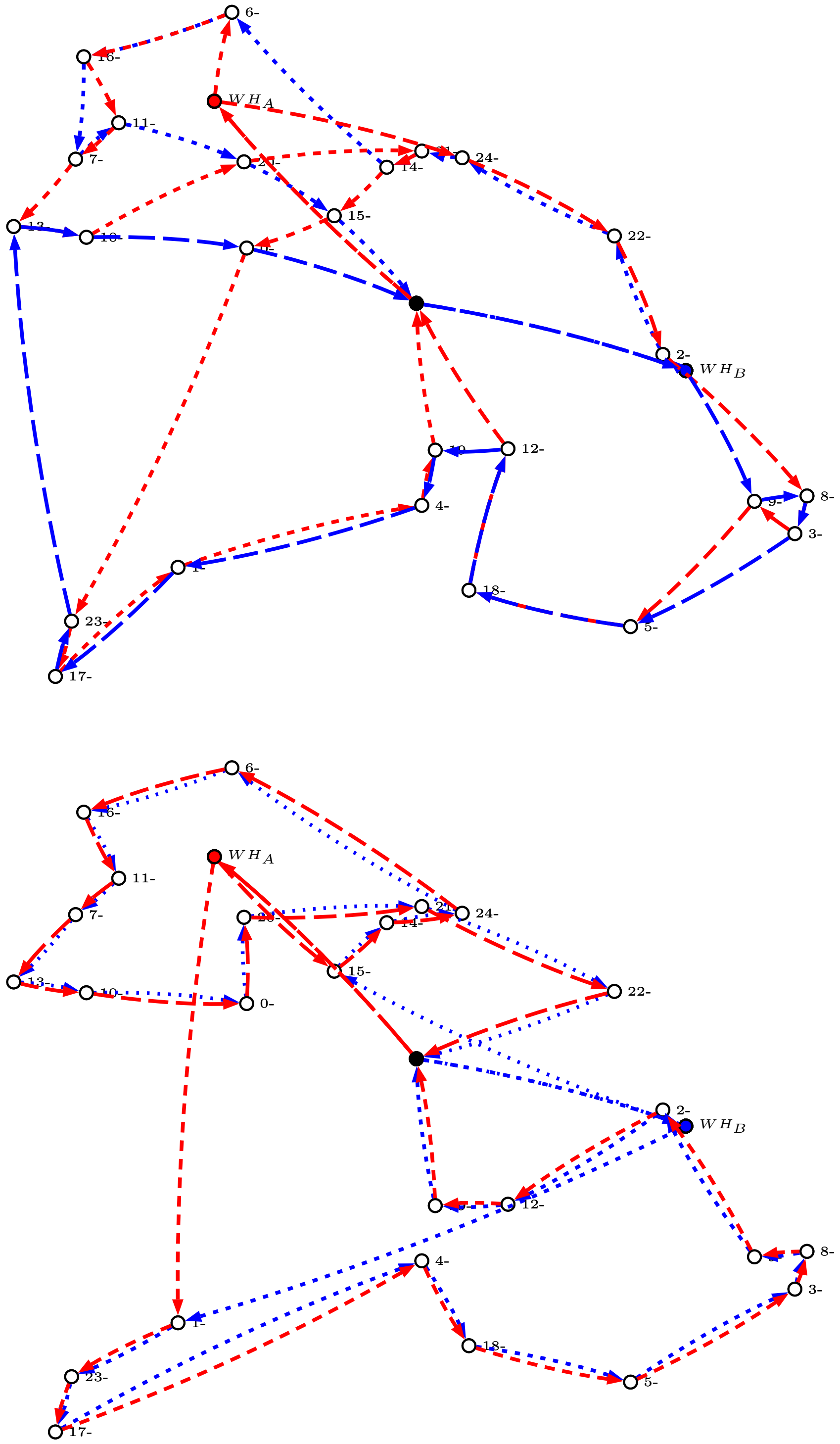

Fig. 6 Instance $\alpha = 4$, $MS^{max} = 2000$: best identified route sets for $\Delta^{max} = \infty$ (upper plot) and $\Delta^{max} = 0$ (lower plot)

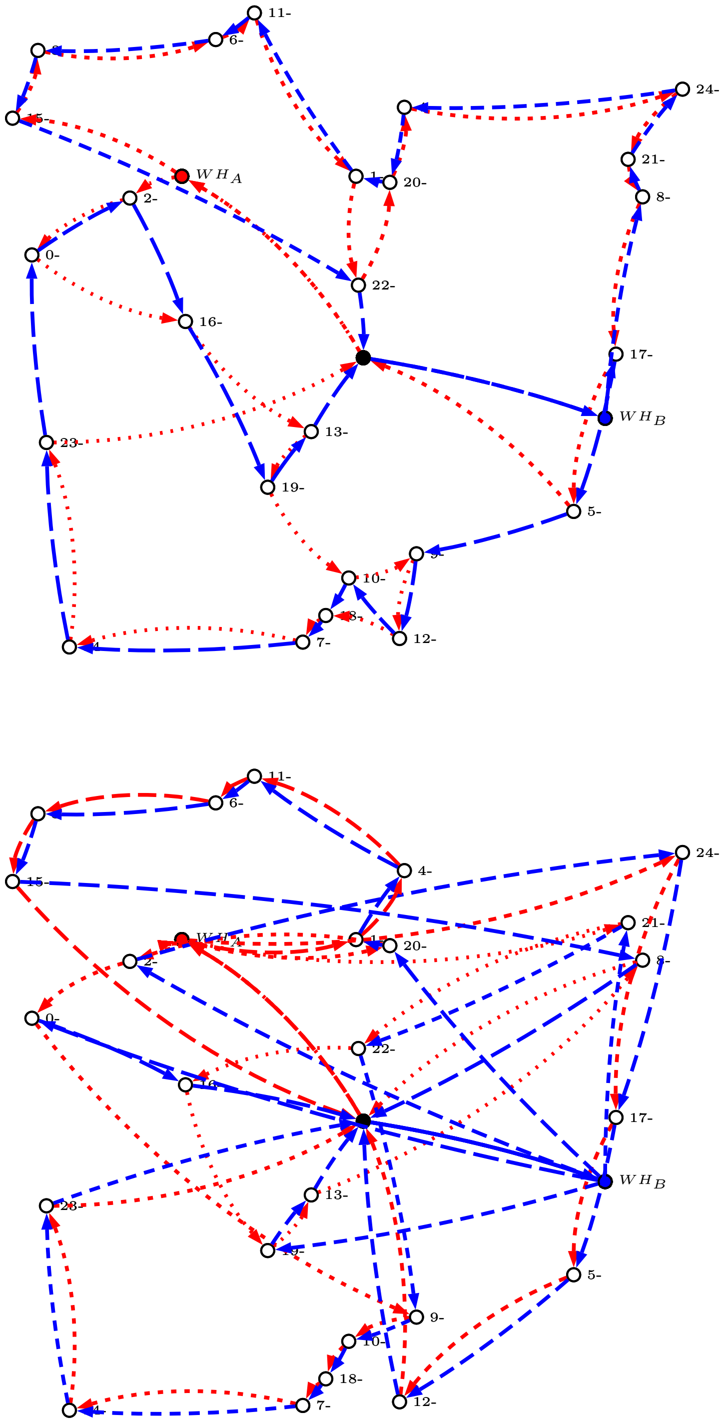

Fig. 7 Instance $\alpha = 1$, $MS^{max} = 2000$: best identified route set for $\Delta^{max} = \infty$ (upper plot) and route set for $\Delta^{max} = 0$ (lower plot)

4 Hill-Climbers Addressing Implicit Time Window Constraints

We have discussed two directions to address feasibility w.r.t. implicit time windows (waiting time insertion as well as deviation from least distance routes) in the problem statement. So far, we have emphasized the insertion of waiting times to synchronize the operation starting times at customer sites. This option is implemented in the memetic search algorithm via insertion of waiting times by the procedure $POSTPONEMENT$. However, if this schedule building procedure is executed it is impossible to vary the routes in order to meet the implicit time window constraints. Reversely formulated, the requirement to achieve feasibility w.r.t. implicit time windows is ignored during the route construction steps. We suppose that the consideration of the implicit time windows during the route construction phase contributes to achieve feasibility w.r.t. the implicit time windows also if $\Delta = 0$. To test this, we develop procedures that explicitly vary the visiting sequences in the route proposals taking into account the needs of implicit time windows.

A comparison of the feasible solutions shown in Fig. 6 as well as Fig. 7 reveals structural differences among the two route sets in the two lower plots. In the feasible solution shown in the lower plot in Fig. 6 one type-A-vehicle as well as one type-B-vehicle follow the same route for a long sequence of visits. Such a congruence between a type-A-vehicle and a type-B-vehicle is not observed in the infeasible solution shown in the lower plot in Fig. 7. We suppose that the assignment of more or less identical visiting sequences contributes to fulfill the requirements of implicit time windows. At the first commonly visited node the earlier arriving vehicle has to wait but at the next commonly visited customer site sequence both vehicles can operate in parallel or with only a small time shift. If the route of one of the two vehicles, say the route of the type-A-vehicle, has been identified to be appropriate (feasible and of an acceptable length) then these properties can also be achieved for the second involved route, say the route of the type-B-vehicle, if the (sub)sequence of operations from the route of the type-A-vehicle is copied to the route of the considered type-B-vehicle. We call this *route alignment* and exploit this idea to equip the previously presented hill climbing procedure with route alignment capabilities. First, we propose two simple and straightforward procedures to modify a visiting sequence alignment without any shifting of requests between routes (Subsection 4.1 and Subsection 4.2). Afterwards, we also propose to revise the request portfolio clustering and exploit the shifting of requests among routes in order to increase the percentage of aligned operation sequences (Subsection 4.3). A revised hill climbing procedure incorporating alignment steps is specified (Subsection 4.4) and an indicator quantifying the alignment degree is proposed in Subsection 4.5.

4.1 Operation Shifting

One approach to solve violations of the implicit time windows is to reposition those operations in the corresponding vehicle routes that cause these violations. Repositioning them reduces the time gap between the starting times of the two delivery operations associated with an order. Fig. 8 shows the pseudo-code of a repositioning procedure. After a tentative schedule determination (b) it is tried to solve implicit time window violation one after another by repeating the loop

```
(a) PROCEDURE SHIFTING($P$, VEHSEQ);
(b)   POSTPONEMENT($P$, VEHSEQ);
(c)   while(implicit time window violation exists)
(d)     $ord$ := randomly selected order with an implicit time window violation;
(e)     ($r_A$,$r_B$) := requests associated with order $ord$;
(f)     ($op_A^-$,$op_B^-$) := delivery operations associated with $r_A$ and $r_B$;
(g)     ($v_A$,$v_B$) := vehicles serving $r_A$ and $r_B$;
(h)     ($dt_A$,$dt_B$) := starting times of $op_A^-$ in route of $v_A$ and of $op_B^-$ in route of $v_B$;
(i)     if( $dt_A > dt_B$)
(j)       shift $op_A^-$ before previous operations in route of $v_A$
            until the implicit time window is fulfilled
            or $op_A^-$ is first delivery operation in the route;
(k)     else
(l)       shift $op_B^-$ before previous operations in route of $v_B$
            until the implicit time window is fulfilled
            or $op_B^-$ is first delivery operation in the route;
(m)     POSTPONEMENT($P$, VEHSEQ);
(n)   wend;
(o) end;
```

Fig. 8 Pseudo-code of the procedure to shift operations with a route

```
(a) PROCEDURE ALIGNPLAN($P$);
(b)   for $i = 1$ to $n^{veh}$;
(c)     $n(v_i)$ := number of operations contained in route of vehicle $v_i$;
(d)   next $i$;
(e)   sort vehicle list be decreasing $n(v_i)$-values;
(f)   for $i = 1$ to $n^{veh} - 1$;
(g)     for $j = 1$ to $n^{veh} - 1$;
(h)       if ($v_i$ and $v_j$ are of different types) then ALIGNROUTES($v_i$,$v_j$);
(i)     next $j$;
(j)   next $i$;
(k) end;
```

Fig. 9 Pseudo-code of the procedure to achieve alignment without re-clustering of the request portfolio

(c)-(m). Each iteration starts with the random selection of an order that shows a time window violation (d). Afterwards, the two associated requests (e) as well as the two associated delivery operations (f) are fetched. The two vehicles that contribute to the fulfillment of the selected order are identified (g). Next, the two delivery operation starting times are identified (h) and compared. The later processed operation is repositioned to an upstream position in its route in order to reduce the gap towards the second delivery operation (i)-(l). After the update of the route the schedule is updated (m) and the repositioning steps are repeated if implicit time window violations are still detected.

4.2 Route Alignment

Another idea to heal violations of implicit time windows is to reposition several operations within a route in order to align operation processing (sub-)sequences to be executed by different vehicles. This idea is realized by the procedure $ALIGNPLAN()$ shown in pseudo-code in Fig. 9. First, the number of serving requests $n(v)$ is determined for each vehicle (b)-(d). Second, the vehicle list is ordered by decreasing

```
(a) PROCEDURE ALIGNROUTES(v, w);
(b)   $ORD(v,w)$ := orders in which one request is served by $v$ and the second request
        is served by $w$;
(c)   label all operations in $v$ and $w$ associated with orders from $ORD(v,w)$;
(d)   $A(v,w)$ := set of labeled operations in $w$;
(e)   if( $A(v,w) \neq \emptyset$ )then
(f)     $SUB\_ROUTE$:=sequence of operation from $A(v,w)$ as determined by $v$;
(g)     $INSERT\_POS$:=operation before first executed operation from $A(v,w)$ in $w$;
(h)     remove all operations found in $A(v,w)$ from $w$;
(i)     insert $SUB\_ROUTE$ after $INSERT\_POS$ in $w$;
(j)   end if;
(k) end;
```

Fig. 10 Pseudo-code of the procedure to align operations contained in two different routes

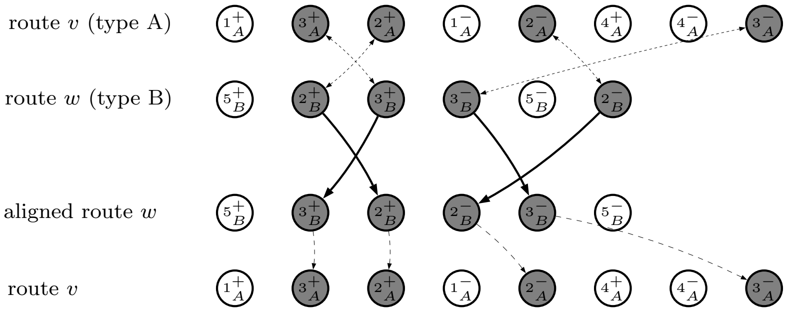

Fig. 11 Alignment of two routes without re-clustering

$n(v)$-values, i.e. v_1 refers to the vehicle that serves the highest number of requests among all vehicles (e). Third, for all $i,j \in \{1,\ldots,n^{veh-1}\}$, the route of vehicle vi is aligned to the route of vehicle v_j if both vehicles are of different types by executing the procedure call $ALIGNROUTES(v_i,v_j)$ (f)-(j).

The pseudo-code of the procedure $ALIGNROUTES(\cdot,\cdot)$ is shown in Fig. 10. After the start of this procedure those orders are labeled that contain requests served by v or w (or both) (b). Next, all operations contained in v or w that are associated with a labeled order are also labeled (c). All labeled operations in route w are collected in the set $A(v,w)$ (d). In case that this set is empty the procedure terminates but if this set is non-empty (e) then the operations in $A(v,w)$ are put in a sequence SUB_ROUTE (f). This sequence is determined by the route v: let o^v_1, o^v_2 be two operations found in v and let o^w_1 and o^w_2 be two operations found in w. The latter two operations are contained in $A(v,w)$. Furthermore, let $o^v_j.req = o^w_j.req$ and $o^v_j.type = o^w_j.type$ ($j \in \{1;2\}$. In the determined SUB_ROUTE o^w_1 precedes o^w_2 if and only if o^v_1 precedes o^v_2 in route v. The direct predecessor operation of the first served operation in w contained in $A(v,w)$ is identified as insertion position (g). Now, all labeled operations are removed from w (h) but the SUB_ROUTE is inserted into w after the insertion position $INSERT\_POS$ (i).

An example application of the procedure $ALIGNMENT(v,w)$ is presented in Fig. 11. The two orders 2 (operations 2^+_A, 2^-_A, 2^+_B and 2^-_B) as well as 3 (operations 3^+_A, 3^-_A, 3^+_B and 3^-_B) are served by the two vehicle routes. The precedence

```
(a)  PROCEDURE RECLUSTER($P$, VEHSEQ);
(b)    POSTPONEMENT($P$, VEHSEQ);
(c)    $\mathbf{v} := (v_1, v_2, \ldots, v_n^{veh})$ = vehicles sorted by increasing route duration;
(d)    while(implicit time window violation exists)
(e)      $ord$ := randomly selected order with an implicit time window violation;
(f)      ($r_A$,$r_B$) := requests associated with order $ord$;
(g)      ($op_A^-$,$op_B^-$) := delivery operations associated with $r_A$ and $r_B$;
(h)      ($v_A$,$v_B$) := vehicles serving $r_A$ and $r_B$;
(i)      ($dt_A$,$dt_B$) := starting times of $op_A^-$ in route of $v_A$ and of $op_B^-$ in route of $v_B$;
(j)      if( $dt_A > dt_B$)
(k)        $v$:=select the first vehicle from $\mathbf{v}$ with same type as $v_A$ but $v_A \neq v$;
(l)        move request $r_A$ from $v_A$ to $v$ and insert $op_A^-$ at randomly selected feasible
             position in route of $v$;
(m)      else
(n)        $v$:=select the first vehicle from $\mathbf{v}$ with same type as $v_B$ but $v_B \neq v$;
(o)        move request $r_B$ from $v_B$ to $v$ and insert $op_B^-$ at randomly selected feasible
             position in route of $v$;
(p)      end if;
(q)      POSTPONEMENT($P$, VEHSEQ);
(r)      $\mathbf{v} := (v_1, v_2, \ldots, v_n^{veh})$ = vehicles sorted by increasing route duration;
(s)    wend;
(t)  end;
```

Fig. 12 Pseudo-code of the procedure to shift operations among routes of different vehicles

relations between the operations contained in route v and the precedence relations among the associated operation observed in route w are different. This can be seen by the crossing dotted arcs. Therefore, both routes are not aligned. Now, the set $A(v, w)$ comprises the four aforementioned operations $2^+_B, 3^+_B, 3^-_B, 2^- B$. It is $SUB\_ROUTE := (3^+_B, 2^+_B, 2^-_B, 3^-_B)$ and $INSERT\_POS := 5^+_B$. After the insertion of SUB_ROUTE immediately after $INSERT\_POS$ both routes are aligned (none of the dashed arcs crosses another dashed arc).

4.3 Alignment with Re-Clustering

The third approach addresses the solving of implicit time window infeasibility by shifting selected requests into the route of another vehicle, i.e. manipulating the clustering decisions. Instead of moving an operation within the route as executed by the procedure $SHIFTING$, the complete request is released from the original route into the route of another vehicle of the same type as proposed for the steps (j)-(p) as part of the procedure $RECLUSTER$ (Fig. 12. A request causing an infeasibility w.r.t. an implicit time window is preferentially shifted into a route of short duration in order to increase the chance to start the shifted delivery operation earlier compared to the original situation (steps (k) as well as (n)).

4.4 Revised Hill-Climbing Procedure

We are now prepared to enrich the hill-climbing procedure with the alignment procedures. Fig. 13 shows the revised hill-climbing procedure that incorporates the three procedures $SHIFTING$, $ALIGNPLAN$ as well as $RECLUSTER$.

```
Function HILLCLIMBER($PLAN$, $VEHSEQ$)
  POSTPONEMENT($PLAN$,$VEHSEQ$);
  if ($PLAN$ is infeas. w.r.t implicit time windows) then ALIGNPLAN($PLAN$);
  CAP_REPAIR($PLAN$);
  MAKESPAN_REPAIR($PLAN$);
  POSTPONEMENT($PLAN$,$VEHSEQ$);
  if ($PLAN$ is infeas. w.r.t implicit time windows) then SHIFTING($PLAN$);
  POSTPONEMENT($PLAN$,$VEHSEQ$);
  if ($PLAN$ is infeas. w.r.t implicit time windows) then RECLUSTER($PLAN$);
  2-OPT($PLAN$);
  POSTPONEMENT($PLAN$,$VEHSEQ$);
  return $PLAN$;
end;
```

Fig. 13 Revised Hill-Climbing Procedure

This sequence of plan modification procedure calls has demonstrated the best performance among different static call sequences.

4.5 Measuring the Extent of Route Alignment

$$L(P) := \frac{N^{pairs}_{used}}{min(N^{pairs}_{poss}, N^{orders})} \tag{4}$$

To check if the application of the proposed alignment procedure(s) actually leads to the installation of a higher number of (partially) aligned routes we define the *linkage degree* $L(P)$ of a solution P. It compares the number N^{pairs}_{used} of pairs of vehicles that actually fulfill at least one of the complex orders with the number of possible combinations N^{pairs}_{poss} of two deployed vehicles of different types (4).

Obviously, the values of $L(P)$ fall into the interval $[\frac{1}{min(N^{pairs}_{poss}, N^{orders})}; 1]$. To demonstrate the meaning of reduced as well as increased $L(P) - values$ we consult the example shown in Fig. 14. Three type-A-vehicles v^A_1, v^A_2, v^A_3 as well as three type-B-vehicles v^B_1, v^B_2, v^B_3 are deployed to visit the six delivery locations 1^-, 2^-, 3^-, 4^-, 5^- and 6^-. The left route set represents an aligned set of routes. Overall, we have $N^{pairs}_{poss} = min(9, 6) = 6$ and $N^{pairs}_{used} = 3$, so that $L(P) = \frac{3}{6} = 0.5$. In the right plot the six customer locations are served by six different combinations of vehicles so that $L(P) = 1$.

In general, we see that $L(P)$ increases if the routes of the vehicles are more interlaced in the sense that more pairs of vehicles are used to commonly serve the complex customer orders. We use $L(P)$ to compare the extent of cross-linking (congruence) found in route sets of two different solutions.

5 Computational Experiments

5.1 Setup of Experiments and Performance Indicators

In order to analyze the performance of the proposed procedures we have tested different extensions of the BVF-configurations of the MA. These extended MAs are implemented by activating selected combinations of the alignment procedures

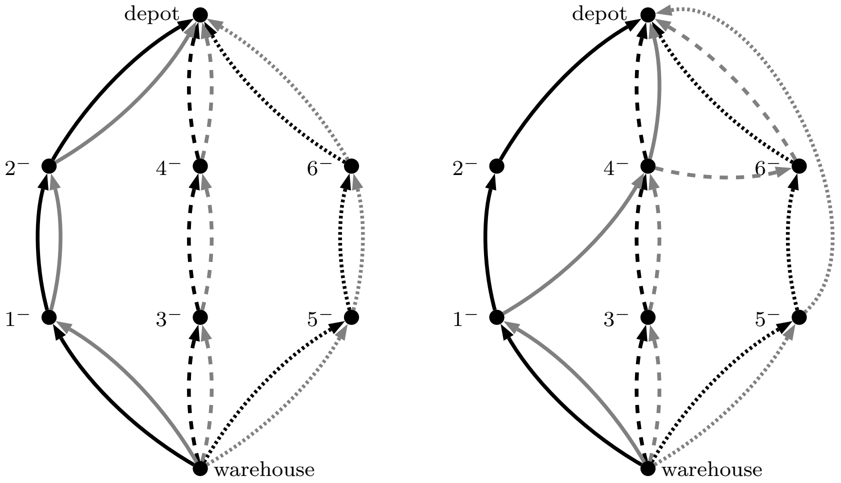

Fig. 14 Examples of route sets with linkage degree $L(P) = 0.5$ (left) as well as $L(P) = 1$ (right)

in the revised hill-climber shown in Fig. 13. First, each procedure is individually incorporated into the hill-climber leading to the configurations BVF + SHIFTING (BVF+S), BVF + ALIGNPLAN (BVF+A) and BVF + RECLUSTER (BVF+R). Second, we report about experimental results observed for the combination of the BVF-MA with the procedures ALIGNPLAN and RECLUSTER (BVF+AR) and about the hybridization of BVF-MA with SHIFTING and RECLUSTER (BVF+SR). The incorporation of ALIGNPLAN and SHIFTING into the BVR-MA results in comparable poor results which are not shown here. Finally, all three procedures ALIGNPLAN, SHIFTING and RECLUSTERING are activated (BVF+ASR). Let $\Phi := \{BVF, BVF + S, BVF + A, BVF + R, BVF + AR, BVF{+}SR, BVF{+}ASR\}$ be the set of tested algorithm configurations. All 900 individual experiments are repeated for the new algorithm configurations BVF+S, BVF+A, BVF+R, BVF+AR, BVF+SR as well as BVF+ASR so that (including BVF) $7 \cdot 900 = 6300$ individual experiments are executed. Results are recorded for each individual experiment and averaged for each combination of a configuration $X \in \Phi$ and (MS^{max}, Δ).

$D_X(MS^{max}, \Delta)$ contains the average of the travel distances observed in all individual experiments with the maximal allowed route duration $MS^{max} \in \{\infty; 3000; 2000\}$ and the common implicit time window length $\Delta\{\infty; 500; 0\}$ if the algorithm $X \in \Phi$ is deployed. Similarly, we define $E_X(MS^{max}, \Delta)$ to store the average number of observed violated implicit time windows. The average number of deployed vehicles is stored in $V_X(MS^{max}, \Delta)$. Furthermore, we calculate and save the average contribution of waiting (idle) time of the deployed vehicles to the total route duration $(W_X(MS^{max}, \Delta))$ as well as the average linkage degree $(L_X(MS^{max}, \Delta))$. In order to quantify the induced travel distances increase resulting from the reduction of the implicit time window length from ∞ down to Δ we calculated the relative travel distance increase $D_X^+(MS^{max}, \Delta) := \frac{D_X(MS^{max},\Delta)}{D_X(MS^{max},\infty)} - 1$.

Cost variations result from (necessary) modifications of the routing plan after a variation of Δ. Here, routing plan modifications comprise the shifting of requests between vehicles, the change of visiting sequences as well as modifications of waiting periods. The waiting period length variation $W_X(MS, \Delta)$ can be calculated immediately. In contrast, the extent of re-assignment as well as re-sequencing decisions requires the setup of a special different measurement. We use the H^2-indicator (Schönberger, 2017) to quantify structure changes between the best plan found without any implicit time windows ($\Delta = \infty$) and the situation with active implicit time window of length $\Delta = 500$ or $\Delta = 0$ time units. The H^2-indicator maps the observed modifications of routing decisions into the interval from 0 (no modification) to 1 (all routing decisions are have updated). Let $H_X^{clust}(MS^{max}, \Delta)$ denote the H^2-distance measuring the variation of assignment decisions and let $H_X^{seq}(MS^{max}, \Delta)$ denote the H^2-distance measuring the variation of sequencing decisions. $H_X(MS^{max}, \Delta)$ is defined as the average of $H_X^{clust}(MS^{max}, \Delta)$ and $H_X^{seq}(MS^{max}, \Delta)$.

5.2 Presentation and Discussion of Offline Results

Tab. 2 contains the observed results for the best identified route sets grouped according to the applied MA-configuration. First, feasibility w.r.t. implicit time windows is guaranteed only if the re-clustering procedure is incorporated (BVF+R, BVF+AR, BVF+SR, BVF+ARS). In all other configurations missed implicit time windows are detected in severely constrained scenarios if $\Delta = 0$. Second, comparing BVF+R, BVF+S as well as BVF+A from the perspective of route length, we see that the alignment of routes of vehicles of different types performs best w.r.t. the compensation of the implicit time window length reduction. In addition, the relative travel distance increase D_X^+ is minimal among the three configurations. Third, if one of the two non-reclustering procedures *ALIGN* or *SHIFTING* is incorporated to improve the performance of the BVF+R configuration then again, *ALIGN* seems to be more suitable (BVF+AR) compared to the configuration BVR+SR. BVR+AR identifies feasible solutions with a significantly reduced travel distance value D_X compared to BVR+SR. Especially, for the tightest implicit time window length ($\Delta = 0$) BVF+AR clearly outperforms BVF+SR. Fourth, for the scenarios with $MS^{max} = 2000$ BVF+AR is outperformed by BVF+ARS. This configuration is able to identify a feasible solution for the scenario with $\Delta = 0$ that shows a travel distance increase $D_{BVF+ARS}^+(2000, 0)$ of only 20% while $D_{BVF+AR}^+(2000, 0)$ equals 31%.

We inspect the properties of the best identified feasible solutions with the goal to explain the different performance of the seven MA-configurations (Tab. 3). The smallest number of vehicles is observed for the configurations that incorporate the *ALIGN*-procedure and the *RECLUSTER*-procedure (BVF+AR as well as BVF+ARS). All other configurations identify solutions that deploy significantly more vehicles in average. Resulting from the reduced number of routed vehicles the contribution of waiting time W_X to the total sum of route durations is also minimal for the two aforementioned configurations BVF+AR as well as BVF+ARS. For these two configurations the observed linkage degrees L_X are slightly below 50%.

Table 2 Average travel distances, makespan-specific relative travel distance increase and average number of missed implicit time windows

		length Δ of implicit time window						
	MS^{max}	∞	500			0		
X		$D_X(MS^{max},\Delta)$	$D_X(MS^{max},\Delta)$	$D^+_X(MS^{max},\Delta)$	$E_X(MS^{max},\Delta)$	$D_X(MS^{max},\Delta)$	$D^+_X(MS^{max},\Delta)$	$E_X(MS^{max},\Delta)$
BVF	∞	5688	7526	32%	-	8852	56%	0.04
	3000	6231	7362	18%	-	9735	56%	0.16
	2000	6549	7460	14%	-	10230	56%	0.32
BVF+S	∞	5688	7291	28%	-	9307	64%	-
	3000	6231	7196	15%	-	9833	58%	0.2
	2000	6549	7641	17%	-	9564	46%	0.24
BVF+A	∞	5688	6040	6%	-	6122	8%	-
	3000	6231	6787	9%	-	7679	23%	-
	2000	6549	7448	14%	-	8310	27%	0.04
BVF+R	∞	5688	7172	26%	-	9248	63%	-
	3000	6231	7164	15%	-	9095	45%	-
	2000	6549	7512	15%	-	9130	39%	-
BVF+SR	∞	5688	7198	27%	-	9395	65%	-
	3000	6231	7077	14%	-	8944	44%	-
	2000	6549	7582	16%	-	9232	41%	-
BVF+AR	∞	5688	5996	5%	-	6074	7%	-
	3000	6231	6747	8%	-	7529	21%	-
	2000	6549	7269	11%	-	8578	31%	-
BVF+ARS	∞	5688	6020	6%	-	6067	7%	-
	3000	6231	6764	9%	-	7674	23%	-
	2000	6549	7299	11%	-	7827	20%	-

This seems to be a compromise between the insertion of waiting times and detours to meet the implicit time windows. In the first situation pairs consisting of a type-A-vehicle as well as a type-B-vehicle follow the same route and one of the vehicles has to wait for the second vehicle at the first commonly visited customer location (low linkage degree). In the second situation, where pairs of vehicles consisting of a type-A-vehicle as well as a type-B-vehicle follow independent routes. In order to delay the arrival at a customer location with the goal to meet the implicit time window, detours are installed (high linkage degree). BVF+ARS leads to least percentages of idle times in most of the situations. Furthermore, in the $\Delta = 0$-cases the least number of vehicles is deployed typically saving also travel distance savings.

The best performing configurations BVF+AR as well as BVR+ARS implement least schedule variations after the reduction of the length of the implicit time window (Tab. 4). In case that the maximal route duration in unlimited ($MS^{max} = \infty$) only sequence modifications are installed in order meet the implicit time window requirement (route inversions). If the maximal allowed duration of a route is $MS^{max} = 3000$ time units then around 40% of the route defining decisions are revised if Δ is reduced down to 500 time units or even 0 time units. Even more than 50% of the routing decisions are modified if the maximal allowed route duration is set to 2000 time units.

Table 3 Deployed vehicles, idle time contribution and linkage degree

		Δ								
	MS^{max}	∞			500			0		
X		$V_X(MS^{max},\Delta)$	$W_X(MS^{max},\Delta)$	$L_X(MS^{max},\Delta)$	$V_X(MS^{max},\Delta)$	$W_X(MS^{max},\Delta)$	$L_X(MS^{max},\Delta)$	$V_X(MS^{max},\Delta)$	$W_X(MS^{max},\Delta)$	$L_X(MS^{max},\Delta)$
BVF	∞	2.0	0%	100%	2.8	14%	78%	3	21%	75%
	3000	3.4	0%	68%	4.0	11%	61%	5.1	19%	52%
	2000	4.0	0%	64%	5.2	7%	50%	7.4	18%	32%
BVF+S	∞	2.0	0%	100%	2.6	11%	100%	3.3	24%	94%
	3000	3.4	0%	91%	3.9	15%	92%	5.3	20%	75%
	2000	4.0	0%	96%	5.4	7%	83%	7.0	18%	55%
BVF+A	∞	2.0	0%	100%	2.0	0%	100%	2.0	3%	100%
	3000	3.4	0%	91%	3.7	14%	85%	4.0	12%	78%
	2000	4.0	0%	96%	5.0	6%	78%	5.6	10%	44%
BVF+R	∞	2.0	0%	100%	2.2	4%	99%	3.4	26%	95%
	3000	3.4	0%	91%	4.0	16%	84%	5.0	22%	74%
	2000	4.0	0%	96%	5.0	6%	84%	6.5	16%	57%
BVF+SR	∞	2.0	0%	100%	2.2	3%	100%	3.2	26%	95%
	3000	3.4	0%	91%	4.0	16%	89%	5.0	23%	78%
	2000	4.0	0%	96%	5.2	7%	85%	6.8	19%	59%
BVF+AR	∞	2.0	0%	100%	2.0	0%	100%	2.0	4%	100%
	3000	3.4	0%	91%	3.7	18%	85%	4.0	12%	61%
	2000	4.0	0%	96%	4.8	6%	82%	5.9	11%	42%
BVF+ARS	∞	2.0	0%	100%	2.0	0%	100%	2.0	3%	100%
	3000	3.4	0%	91%	3.7	12%	83%	4.0	7%	56%
	2000	4.0	0%	96%	5.0	7%	80%	5.2	10%	46%

In summary, the integration of the $ALIGN$ procedure leads to the most significant improvement of the MA w.r.t. to the observed travel distance sum. However, its application has to be supported by the incorporation of the re-clustering procedure in order to achieve feasibility w.r.t. to the implicit time windows. Furthermore, the integration of the procedure $SEQUENCE$ contributes to a further reduction of the objective function value in the most severely constrained scenarios with $\Delta = 0$ and $MS^{max} = 2000$.

The hill climbers specifically designed to achieve implicit time window feasibility have been proposed in order to realize the alignment of the routes of a pair of vehicles of different types. Without these new procedures violations of implicit time windows cannot be prevented (for example see Fig. 7). To conclude the presentation and discussion of the offline results the problem instance from Fig. 7 is revisited. The solution shown in the lower plot of Fig. 7 is infeasible (one implicit time window is missed), requires to travel 13111.18 distance units, deploys 9 vehicles and exhibits a linkage degree of 0.28. In contrast, the approximation of the optimal solution computed by the MA in the BVF+ARS-configuration (shown in Fig. 15) is feasible and requires to travel 8600.30 distance units. It deploys 6 vehicles in total (3 of type A and 3 of type B). The linkage degree is 0.44 compared to 0.28 achieved by the MA without any alignment procedures. This indicates that the BVF+ARS is able to carefully balance the duplication of routes (equivalent with the insertion of waiting times) and detours.

Table 4 Variation of routing decisions

X	MS^{max}	Δ					
		500			0		
		$H_X(MS^{max}, \Delta)$	$H_X^{seq}(MS^{max}, \Delta)$	$H_X^{clust}(MS^{max}, \Delta)$	$H_X(MS^{max}, \Delta)$	$H_X^{seq}(MS^{max}, \Delta)$	$H_X^{clust}(MS^{max}, \Delta)$
BVF	∞	27%	36%	18%	24%	32%	17%
	3000	49%	51%	47%	55%	55%	56%
	2000	55%	51%	60%	59%	54%	64%
BVF+S	∞	22%	31%	14%	31%	39%	23%
	3000	46%	49%	43%	54%	54%	54%
	2000	55%	53%	57%	60%	55%	64%
BVF+A	∞	11%	22%	0%	11%	21%	0%
	3000	45%	51%	40%	49%	49%	48%
	2000	55%	52%	58%	55%	51%	58%
BVF+R	∞	13%	23%	3%	31%	38%	25%
	3000	47%	50%	45%	55%	57%	54%
	2000	55%	52%	58%	58%	54%	63%
BVF+SR	∞	12%	22%	3%	34%	43%	24%
	3000	45%	47%	44%	54%	53%	54%
	2000	53%	51%	54%	56%	52%	59%
BVF+AR	∞	10%	21%	0%	11%	22%	0%
	3000	43%	50%	37%	45%	47%	44%
	2000	51%	51%	51%	57%	54%	61%
BVF+ARS	∞	11%	21%	0%	11%	22%	0%
	3000	41%	45%	37%	44%	44%	44%
	2000	56%	54%	58%	55%	51%	58%

5.3 Online-Analysis of the Algorithm Configurations

In order to acquire information about the reason for the superiority of the configuration BVF+ARS compared to BVF we have conducted additional experiments in which we observe the evolution of some key performance indicators describing the best found solution during the execution of the MAs. For each iteration we save the observed sum of time units of makespan exceeding, the sum of time units of implicit time window exceeding, the number of deployed vehicles as well as the resulting travel distance length. For the two MA-configurations $X \in \{BVF, BVF + ARS\}$ and the highly constraint scenario defined by $MS^{max} = 2000$ and $\Delta = 0$ the observed values are scaled into the interval $[0; 1]$. Since the number of executed iterations varies among the individual experiments, the scaled indicator values have been further aggregated. Therefore, the iteration numbers have been scaled into the interval $[0; 1]$ and the average $d_X^t(MS^{max}, \Delta)$ of the value observed within the equidistant interval $[t, t + 0.01]$ have been calculated for $t = 0, 0.01, \ldots, 0.99$. Similarly, the average makespan exceeding $ms_X^t(MS^{max}, \Delta)$, the average number of time units of implicit time window exceeding $s_X^t(MS^{max}, \Delta)$ as well as the average number of deployed vehicles $d_X^t(MS^{max}, \Delta)$ are determined.

Fig. 16 contains the observed values for the four recently introduced online-performance indicators. First, we see that both MA-configurations (with and wit-

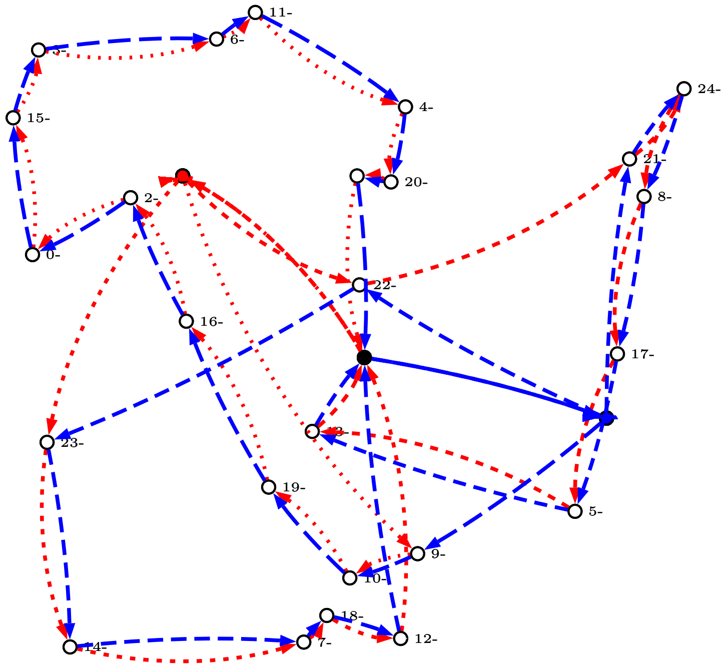

Fig. 15 Re-visited problem instance from Fig. 7: best feasible optimum approximation computed by BVF+ARS

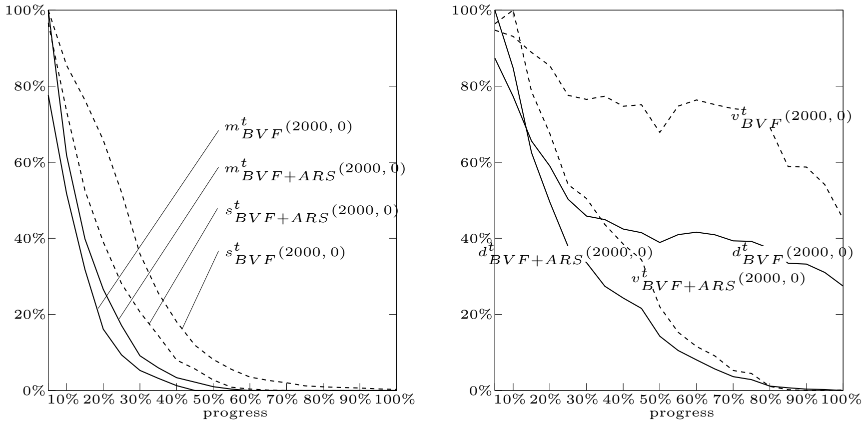

Fig. 16 Online-Analysis: Average development of $s^t_X(2000,0)$, $ms^t_X(2000,0)$ (left picture) as well as $v^t_X(2000,0)$ and $d^t_X(2000,0)$ (right picture) for the MA-configurations $X \in \{BVF; BVF+ARS\}$

hout hill-climber) show a similar comparison regarding the elimination of allowed route duration exceeding (left picture, continuous lines). After approx. 50% of the total processing time feasibility w.r.t. the maximal route duration is achieved. Also the elimination of implicit time window exceeding develops more or less identically in both configurations (dotted lines in the left picture). The quite different performance of BVF and BVF+ARS can be explained by the ability of the MA with specific hill-climber (BVF+ARS) to reduce the number of deployed vehicles significantly quicker compared to the MA without specific hill-climber (BVF). It seems that these hill climbers intensify the propagation of reasonable visiting sequences among the different vehicles. BVF+ARS continuously reduces the number of deployed vehicles during the evolutionary process while BVF let the number of deployed vehicle be stable around 80% during the period $[0.3; 0.8]$ (dotted lines). Similar observations are made for the travel distance development (right picture, continuous lines). In the configuration without specific hill-climbers (BVF) the indicator $d^t_{BVF}(2000; 0)$ stays around 40% during the period $[0.4; 0.8]$. At the same time $d^t_{BVF+ARS}(2000; 0)$ is continuously reduced. BVF achieves feasibility w.r.t. the implicit time windows by incorporating a larger number of vehicles than BVF+ARS at the price of additional travel distances. The specific hill-climbing procedures proposed in Section 4 equip the MA with additional abilities the achieve and sustain implicit time window feasibility without the need of a large number of vehicles.

6 Conclusion

Implicit (or movable) time windows are important in order to coordinate execution times among different operations. Although an implicit time window imposes a less constraining restriction compared to an explicit time window, it is necessary to add new search features to a metaheuristic framework. Without these constraint-specific features there is an enlarged chance that the best found solutions fail to meet implicit time window requirements. In this article, we have discussed a specific implicit time windows constraint in the context of vehicle routing. It has been demonstrated in comprehensive computational experiments that a metaheuristic framework can benefit by incorporating solution manipulating procedures that explicitly exploit knowledge about the impacts of implicit time windows on vehicle route sets. The here reported results contribute to a better understanding of the abilities of metaheuristics to cope adequately with specific constraint types. These results indicate that it is necessary to equip a metaheuristic with constraint-specific search components in order to ensure feasibility w.r.t. to a complicated constraint even if the heuristic mimics a “meta”-behavior.

References

Çağrı Koç, Bektaş, T., Jabali, O., Laporte, G., 2016. Thirty years of heterogeneous vehicle routing. European Journal of Operational Research 249 (1), 1 – 21.

Archetti, C., Campbell, A. M., Speranza, M. G., 2016. Multicommodity vs. single-commodity routing. Transportation Science 50 (2), 461–472.

Archetti, C., Speranza, M. G., 2008. The Split Delivery Vehicle Routing Problem: A Survey. Springer US, Boston, MA, pp. 103–122.

Bierwirth, C., Mattfeld, D. C., Kopfer, H., 1996. On permutation representations for scheduling problems. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 310–318.

Breier, H., 2014. Tourenplanung mit alternativen Lieferperioden und Teillieferungen. KIT Scientific Publishing, Karlsruhe.

Breier, H., Gossler, T., 2014. Branch-and-Price on the Split Delivery Vehicle Routing Problem with Time Windows and Alternative Delivery Periods. Springer International Publishing, Cham, pp. 57–65.

Caceres-Cruz, J., Arias, P., Guimarans, D., Riera, D., Juan, A. A., Dec. 2014. Rich vehicle routing problem: Survey. ACM Comput. Surv. 47 (2), 32:1–32:28.

Cattaruzza, D., Absi, N., Feillet, D., Vidal, T., 2014. A memetic algorithm for the multi trip vehicle routing problem. European Journal of Operational Research 236 (3), 833 – 848, vehicle Routing and Distribution Logistics.

Clausen, T., 2011. Airport ground staff scheduling. Ph.D. thesis.

Crevier, B., Cordeau, J.-F., Laporte, G., 2007. The multi-depot vehicle routing problem with inter-depot routes. European Journal of Operational Research 176 (2), 756 – 773.

de Jong, C., Kant, G., van Vliet, A., 1998. On finding minimal route duration in the vehicle routing problem with multiple time windows. project alcom it, Econometric Institute, Erasmus University Rotterdam, Rotterdam.

Doerner, K. F., Schmid, V., 2010. Survey: Matheuristics for Rich Vehicle Routing Problems. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 206–221.

Drexl, M., 2012. Synchronization in vehicle routing—a survey of vrps with multiple synchronization constraints. Transportation Science 46 (3), 297–316.

Favaretto, D., Moretti, E., Pellegrini, P., 2007. Ant colony system for a vrp with multiple time windows and multiple visits. Journal of Interdisciplinary Mathematics 10 (2), 263–284.

Ferland, J. A., Michelon, P., 1988. The vehicle scheduling problem with multiple vehicle types. The Journal of the Operational Research Society 39 (6), 577–583.

Goel, A., Meisel, F., 2013. Workforce routing and scheduling for electricity network maintenance with downtime minimization. European Journal of Operational Research 231 (1), 210 – 228.

Golden, B., Raghavan, S., Wasil, E. (Eds.), 2008. The Vehicle Routing Problem: Latest Advances and New Challenges. Springer, New York.

Grefenstette, J., 2000. Evolutionary computation 1 - basic algorithms and operators. In: Bäck, T., Fogel, D., Michalewicz, T. (Eds.), Evolutionary Computation 1 - Basic Algorithms and Operators. Institute of Physics Publishing, Ch. Rank-based Selection, pp. 187–194.

Jabali, O., Leus, R., Van Woensel, T., de Kok, T., Mar 2015. Self-imposed time windows in vehicle routing problems. OR Spectrum 37 (2), 331–352.

Kopfer, H. W., Schönberger, J., Kopfer, H., Jun 2014. Reducing greenhouse gas emissions of a heterogeneous vehicle fleet. Flexible Services and Manufacturing Journal 26 (1), 221–248.

Lahyani, R., Khemakhem, M., Semet, F., 2015. Rich vehicle routing problems: From a taxonomy to a definition. European Journal of Operational Research 241 (1), 1 – 14.

Mitrovic-Minic, S., 1998. Pickup and delivery problem with time windows: A survey. SFU CMPT TR 12 (1-43), 38.

Parragh, S. N., Doerner, K. F., Hartl, R. F., 01 2008a. Part ii: Transportation between pickup and delivery locations 58, 81–117.

Parragh, S. N., Doerner, K. F., Hartl, R. F., Apr 2008b. A survey on pickup and delivery problems. Journal für Betriebswirtschaft 58 (1), 21–51.

Pollaris, H., Braekers, K., Caris, A., Janssens, G. K., Limbourg, S., Mar 2015. Vehicle routing problems with loading constraints: state-of-the-art and future directions. OR Spectrum 37 (2), 297–330.

Salhi, S., Imran, A., Wassan, N. A., 2014. The multi-depot vehicle routing problem with heterogeneous vehicle fleet: Formulation and a variable neighborhood search implementation. Computers & Operations Research 52, 315 – 325, recent advances in Variable neighborhood search.

Schmid, V., Doerner, K. F., Hartl, R. F., Savelsbergh, M. W. P., Stoecher, W., 2009. A hybrid solution approach for ready-mixed concrete delivery. Transportation Science 43 (1), 70–85.

Schönberger, J., 2015. The two-commodity capacitated vehicle routing problem with synchronization. IFAC-PapersOnLine 48 (3), 168 – 173, 15th IFAC Symposium onInformation Control Problems inManufacturing.

Schönberger, J., 2005. Operational Freight Carrier Planning. Springer.

Schönberger, J., 2015. Delivery time coordination at customer sites with implicit time windows – a computational evaluation. Zeitschrift für Verkehrswirtschaft 86 (2), 155 – 178.

Schönberger, J., 2017. Computing Indicators for Differences and Similarities Among Sets of Vehicle Routes. Springer International Publishing, Cham, pp. 195–200.

Schönberger, J., Buer, T., Kopfer, H., 2013. Springer Berlin Heidelberg, Berlin, Heidelberg, Ch. A Model for the Coordination of 20-foot and 40-foot Container Movements in the Hinterland of a Container Terminal, pp. 113–127.

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

Stieber, A., Yuan, Z. E., Fügenschuh, A., 2015. School taxi routing for children with special needs. Amos#19, Helmut-Schmidt Universität, Hamburg, Hamburg.

Toth, P., Vigo, D., 2014. Vehicle Routing: Problems, Methods, and Applications, Second Edition. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

Vidal, T., Crainic, T. G., Gendreau, M., Lahrichi, N., Rei, W., 2012. A hybrid genetic algorithm for multidepot and periodic vehicle routing problems. Operations Research 60 (3), 611–624.

SEIT 2000 SIND FOLGENDE DISKUSSIONSBEITRÄGE ERSCHIENEN:

1/2000 Röhl, Klaus-Heiner: Die Eignung der sächsischen Agglomerationsräume als Innovations- und Wachstumspole für die wirtschaftliche Entwicklung des Landes

2/2000 Röhl, Klaus-Heiner: Der Aufbau der ostdeutschen Infrastruktur und sein Beitrag zur wirtschaftlichen Entwicklung in Sachsen

3/2000 Kummer, Sebastian; Mating, Anette; Käsbauer, Markus; Einbock, Marcus: Franchising bei Verkehrsbetrieben

4/2000 Westphal, Jan R.: Komplexitätsmanagement in der Produktionslogistik

5/2000 Röhl, Klaus-Heiner: Saxony's Capital Dresden – on the Way to become Eastern Germany's first "Innovative Milieu"?

6/2000 Schramm, Hans-Joachim: Electronic Commerce im Lebensmitteleinzelhandel - Auswertung einer Konsumentenbefragung im Großraum Dresden

1/2001 Schramm, Hans-Joachim; Veith, Elisabeth: Schwerlasttransport auf deutschen Straßen, Ergebnisse einer Befragung deutscher Schwerlasttransportunternehmen

2/2001 Schramm, Hans-Joachim; Eberl, Katharina: Privatisierung und Going Public von staatlichen Eisenbahnunternehmen - Versuch eines adaptiven Vergleichs zwischen Japan und Deutschland

1/2002 Kummer, Sebastian; Schmidt, Silvia: Methodik der Generierung und Anwendung wertorientierter Performance-Kennzahlen zur Beurteilung der Entwicklung des Unternehmenswertes von Flughafenunternehmen

2/2002 Wieland, Bernhard: Economic and Ecological Sustainability - The Identity of Opposites?

1/2003 Freyer, Walter; Groß, Sven: Tourismus und Verkehr - Die Wechselwirkungen von mobilitätsrelevanten Ansprüchen von touristisch Reisenden und Angeboten (touristischer) Transportunternehmen

2/2003 Stopka, Ulrike; Urban, Thomas: Implikationen neuer Vertriebs- und Distributionsformen auf das Customer Relationship Management und die Gestaltung von virtuellen Marktplätzen im BtoC-Bereich

1/2004 Hoppe, Mirko; Schramm, Hans-Joachim: Use of Interorganisational Systems - An Empirical Analysis

2/2004 Wieland, Bernhard; Seidel, Tina; Matthes, Andreas; Schlag, Bernhard: Transport Policy, Acceptance and the Media

1/2005 Brunow, Stephan; Hirte, Georg: Age Structure and Regional Income Growth

2/2005 Stopka, Ulrike; Urban, Thomas: Erklärungsmodell zur Beurteilung der betriebswirtschaftlichen Vorteilhaftigkeit des Kundenbeziehungsmanagements sowie Untersuchung zur Usability von Online-Angeboten im elektronischen Retailbanking

3/2005 Urban, Thomas: Medienökonomie

4/2005 Urban, Thomas: eMerging-Media: Entwicklung der zukünftigen Kommunikations- und Medienlandschaft

1/2006 Wieland, Bernhard: Special Interest Groups and 4th Best Transport Pricing

2/2006 Ammoser, Hendrik; Hoppe, Mirko: Glossar Verkehrswesen und Verkehrswissenschaften

1/2007 Wieland, Bernhard: Laudatio zur Verleihung der Ehrendoktorwürde an Herrn Prof. Dr. rer. pol. habil. Gerd Aberle

2/2007 Müller, Sven; Kless, Sascha: Veränderung der leistungsabhängigen Schwerverkehrsabgabe in Abhängigkeit der Streckenbelastung

1/2008 Vetter, Thomas; Haase, Knut: Alternative Bedienformen im ÖPNV – Akzeptanzstudie im Landkreis Saalkreis

2/2008 Haase, Knut; Hoppe, Mirko: Standortplanung unter Wettbewerb – Teil 1: Grundlagen

3/2008 Haase, Knut; Hoppe, Mirko: Standortplanung unter Wettbewerb –
Teil 2: Integration diskreter Wahlentscheidungen

1/2009 Günthel, Dennis; Sturm, Lars; Gärtner, Christoph: Anwendung der Choice-Based-Conjoint-Analyse zur Prognose von Kaufentscheidungen im ÖPNV

2/2009 Müller, Sven: A Spatial Choice Model Based on Random Utility

1/2010 Lämmer, Stefan: Stabilitätsprobleme voll-verkehrsabhängiger Lichtsignal-steuerungen

2/2010 Evangelinos, Christos; Stangl, Jacqueline: Das Preissetzungsverhalten von Fluggesellschaften auf Kurzstrecken mit Duopolcharakter

3/2010 Evangelinos, Christos; Matthes, Andreas; Lösch, Stefanie; Hofmann, Maria: Parking Cash Out – Ein innovativer Ansatz zur betrieblichen Effizienzsteigerung und Verkehrslenkung

1/2011 Evangelinos, Christos; Püschel, Ronny; Goldhahn Susan: Inverting the Regulatory Rules? Optimizing Airport Regulation to Account for Commercial Revenues

2/2011 Evangelinos, Christos; Obermeyer, Andy; Püschel, Ronny: Preisdispersion und Wettbewerb im Luftverkehr – Ein theoretischer und empirischer Überblick

1/2012 Geller, Kathleen; Evangelinos, Christos; Hesse, Claudia; Püschel, Ronny; Obermeyer, Andy: Potentiale und Wirkungen des EuroCombi in Deutschland

2/2012 Deweß, Sigrun; Klier, Michael: Verfahren zur Beschränkung von Schwerpunktmodulplätzen am Institut für Wirtschaft und Verkehr

1/2013 Evangelinos, Christos: Infrastrukturpreise - Eine normativ-theoretische Analyse

2/2013 Evangelinos, Christos: Interessengruppen und Preissetzung im Verkehr

1/2014 Hermann, Christina: Die kombinierte Touren- und Personaleinsatzplanung von Pflegediensten – Teil 1: Literatur und Modell

2/2014 Hirte, Georg; Stephan, Andreas: Regionale Beschäftigungswirkungen von öffentlichen Investitionen in Straßen- und Schieneninfrastruktur

1/2015 Schönberger, Jörn: Vehicle Routing with Source Selection - Integrating Sourcing in Fleet Deployment

2/2015 Schönberger, Jörn: Scheduling of Sport League Systems with Inter-League Constraints

3/2015 Schönberger, Jörn: Hybrid Search and the Dial-A-Ride Problem with Transfer Scheduling Constraints

1/2016 Hermann, Christina: Die kombinierte Touren- und Personaleinsatzplanung von Pflegediensten – Teil 2: Ergebnisse

2/2016 Evangelinos, Christos; Wittkowski, Antje; Püschel, Ronny: A Note on "Price Cap Regulation of Congested Airports"

1/2017 Schönberger, Jörn: Implicit Time Windows and Multi-Commodity Mixed-Fleet Vehicle Routing