TECHNISCHE UNIVERSITÄT DRESDEN

Fakultät Verkehrswissenschaften „Friedrich List“

DISKUSSIONSBEITRÄGE AUS DEM INSTITUT FÜR WIRTSCHAFT UND VERKEHR

NR. 3/2015

JÖRN SCHÖNBERGER

HYBRID SEARCH AND THE DIAL-A-RIDE PROBLEM WITH TRANSFER SCHEDULING CONSTRAINTS

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.

CASPT 2015

Hybrid Search and the Dial-A-Ride Problem with Transfer Scheduling Constraints

Jörn Schönberger

Abstract In a conventional dial-a-ride-system passengers are moved with the same vehicle between their pickup and their drop-off location. In a dial-a-ride-system with transfer, it is possible (or even standard) that passengers change the vehicle once or several times. Transfer Scheduling Constraints (TSC) are imposed in order to ensure that the comfort of the transfer remains on an acceptable level by avoiding too short or too long transfer times but also for limiting the total riding time between the initial pickup location to the final destination. In this contribution, we investigate the dial-a-ride-problem with transfer scheduling constraints (DARP-TSC) as an example for routing scenarios with TSC. We provide initial insights into the consequences of introducing TSCs using computational experiments with a memetic algorithm metaheuristic.

Keywords dial-a-ride · transfer planning · mathematical programming · metaheuristic · memetic algorithm

1 Introduction

The acceptance of public transport systems mainly depends on the service level it provides to passengers. In urban areas regular services following a previously published timetable form an appropriate operation mode to serve individual origin-to-destination demand (OD-demand). In rural areas but also in some suburban areas or during night or other periods with low demand, on-demand services are often used to replace or enrich the regularly executed timetable-based transport services. Beside large vehicles, mini-buses or taxis are deployed in response to a customer pre-booking via online-platforms or via call-ins. These services are called dial-a-ride services and the resulting dispatching problem is referred to as the dial-a-ride problem (DARP).

J. Schönberger
Technische Universität Dresden
Tel.: ++49 (0)351 463-36815
E-mail: joern.schoenberger@tu-dresden.de

In a conventional dial-a-ride-system passengers are moved with the same vehicle between their pickup and their drop-off location. In a dial-a-ride-system with transfer, it is possible (or obligatory) that passengers change the vehicle once or several times. In this paper, we assume that there is one central transfer hub. This hub is used by all passengers that travels from a pickup location situated in region $\mathscr{A}$ to a destination situated in another region $\mathscr{B}$. Inbound passengers going to the hub are collected from their pickup points and are brought to this transfer hub where they change the vehicle. The outbound passengers going out from the hub are then distributed to their indicated individual drop-off points.

The scheduling of inbound and outbound services at the transfer hub come along with several challenges that increases the complexity of the underlying routing problem. Obviously, it is necessary that the inbound passengers arrive at the transfer hub before their associated outbound service leaves the hub. Furthermore and for the reason of offering a quite high service and comfort level to the passengers changing the vehicle there are constraints that have to be considered during the transfer planning. In this article, we are going to focus on the following two service-oriented requirements (i) The total transfer time must not exceed a given threshold. Here, the transfer time includes the riding time from the pickup location to the hub, the waiting/transfer time at the hub as well as the riding time for the distribution on board of the outbound vehicle (maximal OD-transfer time OD^{max}) (ii) A least transfer time TT^{mix} has to be ensured for all passengers in order to enable elder persons or persons with limited mobility a comfortable and relaxed transfer between the inbound and the outbound service providing vehicle.

The here reported research proposes adequate constraints that can enrich mathematical optimization models of the DARP leading to a model for the Dial-A-Ride-Problem with Transfer Scheduling Constraints (DARP-TSC). In addition to previous work on dial-a-ride-problems with transfers we particularly address customer convenience. Therefore, the maximal OD-riding time as well as the least transfer time have to be considered during the route compilation as well as during the schedule construction. The primary research question addressed in this paper is: *What are the quantitative impacts of considering customer-convenience related aspects on travel distances, vehicle operation times and the number of needed vehicles?*

We start with description of the DARP-TSC in Section 2. Section 3 addresses the development of a metaheuristic for the DARP-TSC. We report and analyze results from initial computational experiments with the DARP-TSC in Section 4.

2 Dial-A-Ride-Problem with a Central Transfer Hub

2.1 Related Literature

Crainic (2000) defines three phases of transport planning. Network design (long term), network configuration (mid term) and network deployment (short term). Typical applications of the DARP-TSC are found in the two latter mentioned planning phases. Network configuration comprises the installation of regularly served transport connections between nodes of a network. In this context, DARP-TSCs must be solved in order to achieve an overview about the network performance. Network deployment comprises the decision about the routes of vehicles between pairs of nodes through the network in case that explicit transport demand specified by different customers (requests) is known.

Splitting a request is an option that can be drawn to improve a planning objective function value. Instead of one request two concatenated request must now be handled. It is beneficiary to split a request and assign different parts of a request to different vehicles that commonly deliver the complete request to a customer site e.g. if the capacity utilization degree of a vehicle can be increased. Such a situation is called a vehicle routing problem with split deliveries (Archetti and Speranza (2008)). In a DARP-TSC splitting an OD-request into a collection as well as a distribution request is not part of the problem. An OD-request is split into the collection (or inbound) request and the distribution (or outbound) request before the route construction is initiated. However, the interdependency between the two requests has to be considered during route construction and schedule building. The pair of collection request and distribution request is called a (complex) order (Schönberger, 2015a). The coordination of activities or operations executed commonly by different vehicles at a network node is investigated so far for terminal nodes in a transhipment or cross-docking applications (Bredström and Rönnqvist, 2008).

Drexl (2012) uses the term synchronization to describe the need to consider any type of coordination between different entities in a transport network. He classifies the synchronization requirements that appear in transport logistics into five categories. *Task synchronization* is required in the event that several resources (e.g. vehicles) must fulfill a demand cooperatively (like in the split delivery vehicle routing problem). If loading or unloading operations must be coordinated by time and/or location then this coordination is referred to as *operation synchronization*. *Movement synchronization* means that two or even more resources (vehicles) must use the same path like in truck-trailer applications Drexl (2014). If cargo is interchanged between vehicles then *load synchronization* is needed and *resource synchronization* addresses the situation when two or more vehicles use (”share”) the same (scarce) resource(s). According to this classification, DARP-TSC claim a kind of task and/or operation synchronization.

The dial-a-ride-problem (DARP) belongs to the class of vehicle routing problems (Golden et al, 2008) and it is a special case of the capacitated pickup and delivery problem surveyed by Parragh et al (2008). A survey of DARPs is given by Cordeau and Laporte (2007). Scheduling constraints mainly address time windows for pickup and delivery operations at customer sites Mues and Pickl (2005). The incorporation of transfers into DARP is investigated by Masson et al (2014) as well as Deleplanque and Quilliot (2013).

2.2 Informal Problem Statement

We consider an on-demand public transport service system like a night bus or a service for handicapped persons. This system offers a demand-responsive transport of individuals as well as groups of passengers from a customer specified pickup location to a customer specified drop-off location. An individual customer demand o is described by the order (p_o, d_o, c_o), where p_o (d_o) represents the customer specified pickup (drop-off) location and c_o gives the number of passengers to be transported between p_o and d_o. The considered dial-a-ride system covers an area that is partitioned into two regions $\mathscr{A}$ and $\mathscr{B}$. An order falls into one of the following four categories depending on the position of the pickup as well as of the delivery location: (i) an *AA*-order o starts and terminates in region A, i.e. $p_o, d_o \in \mathscr{A}$. These orders are collected in $\mathscr{O}^{AA}$ (ii) A *BB*-order o starts and terminates in region B, i.e. $p_o, d_o \in \mathscr{B}$. These orders are collected in $\mathscr{O}^{BB}$ (iii) An *AB*-order o starts in region $\mathscr{A}$ and terminates in region $\mathscr{B}$, i.e. $p_o \in \mathscr{A}$ and $d_o \in \mathscr{B}$. These orders are collected in $\mathscr{O}^{AB}$ (iv) A *BA*-order o starts in region $\mathscr{B}$ and terminates in region $\mathscr{A}$, i.e. $p_o \in \mathscr{B}$ and $d_o \in \mathscr{A}$. These orders are collected in $\mathscr{O}^{BA}$.

All available orders are collected in the set $\mathscr{O} := \mathscr{O}^{AA} \cup \mathscr{O}^{BB} \cup \mathscr{O}^{AB} \cup \mathscr{O}^{BA}$. There is a fleet $\mathscr{V}$ of m vehicles (buses, mini vans with passenger seats or even taxis) available to serve the orders. Each vehicle is allowed to operate only into one specific region. Some of these vehicles can only be used to serve orders in region $\mathscr{A}$. These vehicles are collected in the sub-fleet $\mathscr{V}^A$. The remaining vehicles form the sub-fleet $\mathscr{V}^B$. These vehicles can only be assigned to requests within region $\mathscr{B}$. The fixed assignment of a vehicle to a region is caused by the source of funding. Sponsoring or public funding requires the strict assignment of a vehicle to regional demand. Another reason is the fact that different bus operators have contracts for one region only. In order to provide inter-regional transport services starting from region A and terminating in regionB a transfer point H is introduced. Passengers going from region $\mathscr{A}$ to $\mathscr{B}$ have to change the vehicle at H. That is, an *AB*-order is split into an *AA*-collection-request from p_o to H and an *BB*-distribution request originating from H and terminating in d_o. Similarly, a *BA*-order is split up. Each *AB*-order o consists of two requests $r_A^o := (p_o; H; c_o)$ and $r_B^o := (H; d_o; c_o)$. We call all requests starting and terminating in $\mathscr{A} \cup H$ a **type-A-request** and a request starting and terminating in $\mathscr{B} \cup H$ a **type-B-request**.

In a DARP-TSC with a given set of orders $\mathscr{O}$ a trip is generated for each vehicle. This trip of vehicle v is a path that starts at the initial position of v, serves all received requests and finally terminates at the terminus point of vehicle v. The following conditions must be met by a feasible route (C1) the pickup location of a request must be visited before the corresponding drop-off position is visited (C2) the limited capacity of the vehicle must not be exceeded (C3) a vehicle can only be assigned to a request that is situated into its operational area (C4) all vehicles have to return finally to their starting positions after MS^{max} time units (C5) the operation synchronization requirements must be respected, e.g. least and maximal transfer times have to be considered and the total ride span must not be exceeded.

An example of the DARP-TSC with 6 orders is shown in Fig. 1. Here, region $\mathscr{A}$ is formed by the nodes $\{A^+; B^+; C^+; D^-; E^-; F^-\}$. The second region $\mathscr{B}$ is formed by the nodes $\{A^-; B^-; C^-; D^+; E^+; F^+\}$.

There are six orders $o_A := (A^+; A^-; 1)$, $o_B := (B^+; B^-; 1)$, $o_C := (C^+; C^-; 1)$, $o_D := (D^+; D^-; 1)$, $o_E := (E^+; E^-; 1)$, $o_F := (F^+; F^-; 1)$ each comprising one person. These six orders are represented by the narrow dashed arcs. According to the

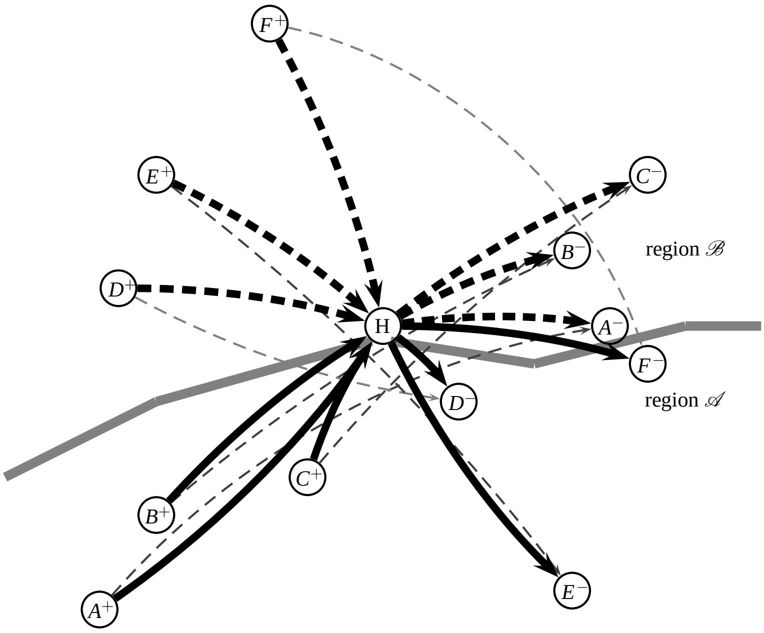

Fig. 1 DARP-TSC example with transfer hub H and 6 orders

previous definitions, the first three orders are defined to be *AB*-orders and the remaining three orders are classified as *BA*-orders.

From the six inter-region orders we derive six type-*A*-requests (printed as continuous thick arcs in Fig. 1) as well as six type-*B*-requests which are represented by the dotted thick arcs in Fig. 1). A least distance set of trips is searched.

We assume that each of the two regions $\mathscr{A}$ and $\mathscr{B}$ provides two vehicles of capacity 3. The coordination of the schedules of vehicle operations at the transfer hub H requires particular attention. First, arrival as well as leaving times of the two vehicles contributing to the fulfillment of an order have to be coordinated in order to ensure that the passenger transfer can be realized. Here, the inbound request operation at H associated with order o must be finished before the associated outbound operation associated with o is initiated. Second, in order to offer transfer passengers a comfortable change of vehicles, a least transfer time TT^{min} is required between the associated inbound as well as outbound service. Third, the waiting time of a passenger at the transfer hub H is limited by TT^{max} time units. Fourth, the total service time of an order (OD-transfer time), which is the time between the pickup up and the drop-off must not exceed a given threshold value OD^{max}. The DARP-TSC searches for a set of trips of the four vehicles with a minimal sum of travel distances so that the aforementioned synchronization requirements are met.

2.3 Mixed-Integer Linear Program: Operation Synchronization Constraints

The DARP-TSC can be modeled as a generalized capacitated vehicle routing problem. Schönberger (2015b) discusses a mixed-integer linear program for a variant of the capacitated vehicle routing problem in which the starting times of the two delivery operations associated with a complex order must not differ by more than a given number of time units. In the following, we replace these synchronization constraints by a set of constraints representing the aforementioned operation (task) synchronization requirements.

Let $o := (r_1; r_2) \in \mathscr{O}$ be a complex order. For the remainder of this paper the first request r_1 is the collection request and r_2 represents the associated distribution request. In case that the triple $r = (p_r^+; p_r^-; c_r)$ represents a request then p_1^- as well as p_2^+ coincide with the central transfer hub H.

The binary decision variable y_{rv} equals 1 if and only if request r is assigned to vehicle v. Four different operation times are stored as continuous non-negative decision variables at_{iv} represents the arrival time of vehicle v at node i. Similarly, we store the operation starting times st_{iv}, the operation completion time ct_{iv} and the leaving time lt_{iv}. Node duplication enables the multiple visit of a location.

$$ct_{p_1^-,v_a} + TT^{min} \leq st_{p_2^+,v_b} + (2 - y_{r_1,v_a} - y_{r_2,v_b}) \cdot \mathscr{M} \ \forall o = (r_1; r_2) \in \mathscr{O}, v_a, v_b \in \mathscr{V} \quad (1)$$
$$st_{p_2^+,v_b} \leq st_{p_1^-,v_a} - TT^{max} + (2 - y_{r_1,v_a} - y_{r_2,v_b}) \cdot \mathscr{M} \ \forall o = (r_1; r_2) \in \mathscr{O}, v_a, v_b \in \mathscr{V} \quad (2)$$
$$ct_{p_2^-,v_b} - st_{p_1^+,v_a} \leq OD^{max} + (2 - y_{r_1,v_a} - y_{r_2,v_b}) \cdot \mathscr{M} \ \forall o = (r_1; r_2) \in \mathscr{O}, v_a, v_b \in \mathscr{V} \quad (3)$$

Constraint (1) ensures that at least TT^{min} time units are left between the completion of the unloading operation associated with the inbound vehicle $v_a \in \mathscr{V}$ and the start of boarding into the outbound vehicle $v_b \in \mathscr{V}$. The consideration of constraint (2) ensures that at most TT^{max} time units are left between the completion of the unloading operation associated with the inbound vehicle $v_a \in \mathscr{V}$ and the start of the boarding into the outbound vehicle $v_b \in \mathscr{V}$. No more than OD^{max} time units are scheduled between the initial pickup time and the final delivery of the passenger associated with order o (3).

2.4 Decision Model Validation

We have applied the CPLEX-solver to different instances of the DARF-TSC. Fig. 2 contains an optimal trip set for the example introduced above in Fig. 1. Only the least transfer time $TT^{min} := 10$ must be considered as TSC. In the optimal solution, two vehicles are deployed in region $\mathscr{A}$ (continuously drawn arcs as well as dotted arcs) but only one vehicle is deployed in region$\mathscr{B}$ (dashed arcs). First, vehicle 1 (continuous arcs) pickups up passengers from B^+, A^+as well as C^+ and brings them

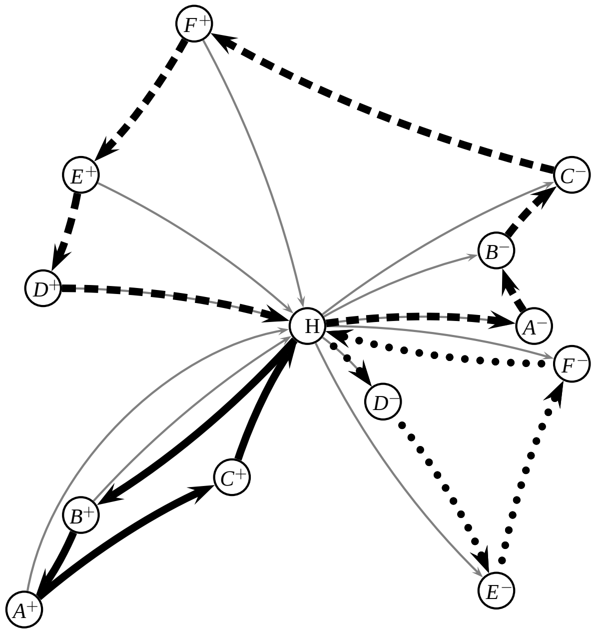

Fig. 2 DARP-TSC example: optimal solution with minimal transfer time $TT^{min} = 10$ time units and $TT^{max} = \infty$ (no maximal waiting time)

to the transfer point H where they arrive at time 245. From here, they left at 255 with vehicle 3 (dashed arcs) which first drops off passengers at A^-, B^- and C^-. After that, it picks up the passengers from F^+, E^+ and D^+. These transfer passengers arrives at the transfer point H at time 668 and after 10 time units vehicle 2 (dotted arcs) brings them to their destinations in region $\mathscr{A}$. The total sum of trip lengths is 845.63 distance units.

Tab. 1 gives an insight into the observed pickup times t_p, drop-off times t_d as well as the resulting OD-transfer times t_{OD} observed in different experiments. The first experiment represents the unlimited case without any control of the OD-transfer length or the trip duration. A minimal transfer time of 10 time units is guaranteed. In the second experiment, we limit the OD-transfer time to 300 time units. Here, this constraint can be met by reversing the route of one of the deployed vehicle 2 in region $\mathscr{A}$. The overall travel distances remains the same but the makespan of the schedule is slightly increased.

Compared to the limitation of the OD-transfer time we observe significant schedule changes if we limit the schedule makespan to 700 time units. Now, a second vehicle is operated into region $\mathscr{B}$ (Fig. 3), whose deployment leads to a significant

Table 1 Pickup times, drop off times and OD-transfer times

TT^{min}		10			10			10	
OD^{max}		∞			300			∞	
MS^{max}		∞			∞			700	
order	t_p	t_d	t_{OD}	t_p	t_d	t_{OD}	t_p	t_d	t_{OD}
A	107	333	226	79	333	254	107	333	226
B	78	351	273	107	351	244	78	379	301
C	162	365	203	162	365	203	163	365	202
D	583	707	124	584	837	253	196	320	124
E	540	758	218	540	786	246	153	371	218
F	482	819	337	481	724	243	95	433	338
obj.		845			845			933	
makespan		819			837			432	

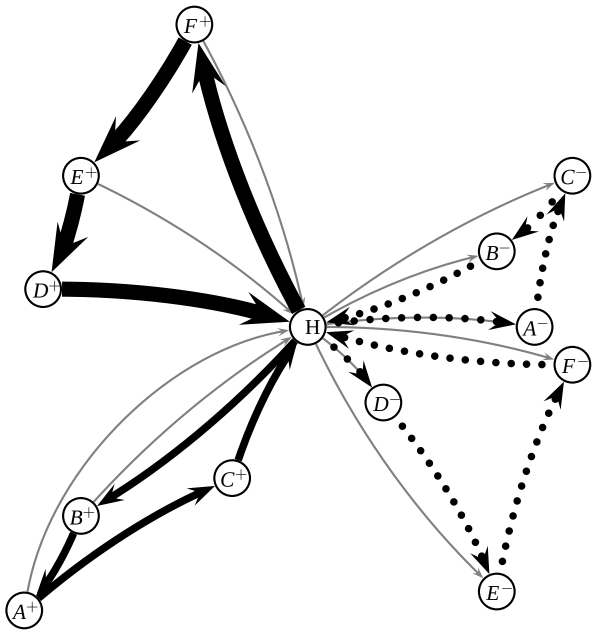

Fig. 3 DARP-TSC example: optimal solution with minimal transfer time $TT^{min} = 10$ time units and $MS^{max} = 700$

increase of the travel distance of the fleet to 933 length units. However, the makespan of the schedule is reduced down to 432 time units.

3 Metaheuristic Approach

We have seen in the previously reported experiments that even small instances of the proposed analytic decision model require huge computational resources. 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, 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 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. The precedence constraint (C1) as well as the capacity constraint (C2) and the area consistency constraint (C3) 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 feasibility with respect to (C1), (C2) as well as (C3). Infeasibility with respect to (C4) as well as (C5) is accepted but penalized during the evolutionary process. A population management scheme is used that prefers those solution proposals with few or even no violations of (C4) or (C5).

3.1 Plan Construction Procedure

Let op be an operation. An attribute $attribute$ of this operation is labeled by $op.attri-bute$. 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 operations. All routes form a plan. There-

```
(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 (C3);
(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 (C1);
(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. (C2) is achieved);
(q)    next i;
(r)  end;
```

Fig. 4 Pseudo-code for the plan generation procedure

fore, a plan p consists of an array $p.route_1, \ldots, p.route_m$ 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 (C1)-(C3) is guaranteed but feasibility with respect to (C4) as well as (C5) is not guaranteed. The plan generation is controlled by a permutation of the available requests, e.g. calling the procedure with two different permutations generate two different plans.

The pseudo-code of the plan generation procedure is shown in Fig. 4. 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 (C1), (C2) as well as (C3).

```
(a)   PROCEDURE SCHEDULE(P);
(b)     for r = 1 to m;
(c)       OP := P.route_r.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 ≠ NULL )
(j)         OP.at := OP.previous.lt
                     + TravelTime[OP.previous][OP];
(k)         OP.st := OP.at;
(l)         OP.ct := OP.st + OP.servicetime;
(m)         OP.lt := OP.ct;
(n)         OP := OP.next;
(o)       wend;
(p)     next r;
```

Fig. 5 Basic forward-scheduling procedure

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 pseudo-code of such a forward-scheduling procedure for the determination of the schedule of a given plan *p* is given in Fig. 5. For each of the given *m* routes the arrival time at_{mi} of vehicle *m* at *i* node is calculated recursively along the proposed route as well as the operation starting time st_{mi}, the operation completion time ct_{mi}, and the leaving time lt_{mi}. The procedure starts with the selection of the next route (b). The following loop (c)-(o) is repeated for all subsequent routes. Each loop starts with setting the operation pointer *OP* to the dummy start operation of the currently considered route (c). All four time attributes are set to 0.0 for this operation (d)-(g). The direct successor operation of the start operation is selected (h). The following loop of the schedule construction steps (j)-(n) is repeated until the scheduling decisions for the dummy end operation in the current route have been made. Each loop starts with the determination of the arrival time which is set to the leaving time from the previously visited operation location plus the travel time to the location of the currently considered operation (j). The associated loading or unloading operation is scheduled to start immediately after the location has been reached (k) and the operation completion time is calculated (l). As soon as the operation is completed the considered vehicle leaves the currently considered operation (m). The loop is completed by selecting the next operation in the considered route (n).

The application of the 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 so that the execution duration of a route is minimized. The order in which the routes are scheduled does not influence the

```
(a)  PROCEDURE POSTPONEMENT(P; VEHSEQ);
(b)    for r = 1 to m;
(c)      op := P.route[VEHSEQ[r]].start;
(d)      OP.at := 0.0;
(e)      OP.st := 0.0;
(f)      OP.ft := 0.0;
(g)      OP.lt := 0.0;
(h)      OP := OP.next;
(i)      while( OP.next ≠ NULL )
(j)        OP.at := OP.previous.lt
                   + TravelTime[OP.previous][OP];
(k)        OP.st := OP.at;
(l)        if(OP.reqtype = DISTRIBUTION )                              ⎫
(m)          if (OP.type = PICKUP )                                    ⎪
(n)            if (OP.order.r_1.delivery_operation already scheduled)  ⎪ check for post-
(o)              OP.st := OP.order.r_1.delivery_operation.ct + TT^min  ⎬ ponement of op
(p)            end;                                                    ⎪
(q)          end;                                                      ⎪
(r)        end;                                                        ⎭
(s)        OP.ct := OP.st + OP.duration;
(t)        OP.lt := OP.ct;
(u)        OP := OP.next;
(v)      wend;
(w)    next r;
```

Fig. 6 Scheduling procedure with postponement

scheduling decisions at all because no coordination of the scheduling decision made in a route with previously made scheduling decisions in other routes are needed. However, the inter-route coordination of starting and completion times imposed by the synchronization constraints (1)-(3) requires the consideration of scheduling decisions made for operations contained in previously scheduled routes. Hence, the sequence in which the routes $PLAN.route_1, \ldots, PLAN.route_m$ are scheduled is important since previously determined operation starting / completion times of operations impose implicit time windows for the so far unscheduled operations. In order to prevent operations from starting to early, we propose the following modification of the basic procedure described above. The scheduling sequence for the vehicles of the operations is determined by a vehicle permutation $VEHSEQ$.

With respect to the DARP-TSC the major deficiency of the simple scheduling procedure proposed in Fig. 5 is the missing of features to establish an inter-route operation starting time coordination of operations belonging to the same order. More concretely, let v_1 and v_2 be the two vehicles assigned to the collection request r_1 respectively to the associated distribution request r_2. Let op_1 be the reference to the delivery operation of v_1 at the central transfer hub and let op_2 be the reference to the pickup operation of v_2 associated with r_2 at the transfer hub. If the operation completion time $op_1.ct$ has already been fixed then the procedure $SCHEDULE$ does not coordinate $op_1.ct$ with $op_2.st$ so that the consideration of the least transfer time cannot be guaranteed.

In order to equip the procedure $SCHEDULE$ to take care about the needed coordination of operation starting and completion times, we insert the steps (l) to (r) into the scheduling procedure as shown in Fig. 6. The basic idea is to try to postpone the starting time of the currently considered pickup operation op_2 in the route of vehicle v_2 serving the distribution request if the associated operation in the second vehicle v_1 has already been scheduled (l)-(n). Here, postponing the operation means that the operation starting time $op_2.st$ is set to at least $op_1.ct + TT^{max}$ time units if possible (o). However, such a postponement guarantees a reduction or prevention of missed least transfer times only if the route serving the collection request of an order is scheduled before the route containing the associated distribution request. In this case the pickup starting time of the distribution request can be shifted to the right so that it starts at least TT^{max} time units after the associated delivery operation of the respective collection order (postponement). In all other cases, the consideration of the least transfer time cannot be guaranteed.

3.3 Comparison of Plans and Population Ranking

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 synchronization constraints and (iii) the objective function value $E^D(POP_i)$, which is the sum of the travel distance of all routes.

$$POP_{i_1} \succ POP_{i_2} \Leftrightarrow \begin{cases} E^{MS}(POP_{i_1}) \leq E^{MS}(POP_{i_2}) \text{ or} \\ \\ E^{MS}(POP_{i_1}) = E^{MS}(POP_{i_2}) \\ \wedge E^{SYNC}(POP_{i_1}) \leq E^{SYNC}(POP_{i_2}) \text{ or} \\ \\ E^{MS}(POP_{i_1}) = E^{MS}(POP_{i_2}) \\ \wedge E^{SYNC}(POP_{i_1}) = E^{SYNC}(POP_{i_2}) \\ \wedge E^{D}(POP_{i_1}) \leq E^{D}(POP_{i_2}) \end{cases} \tag{4}$$

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 (4). 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. 7 shows a hill-climbing procedure that is applied to each generated solution proposal in order to reduce the number of constraint violations w.r.t to (C4). Furthermore, this procedure tries to reduce the total travel distance sum (the objective function value). First, a tentative operation scheduling is made (b). Second, with the

(a) **Function** HILLCLIMBER(*PLAN*, *VEHSEQ*)
(b) POSTPONEMENT(*PLAN*,*VEHSEQ*);
(c) DURATION_REPAIR(*PLAN*);
(d) 2-OPT(*PLAN*);
(e) **end**;

Fig. 7 Hill-Climbing and Repair Procedure

goal to reduce the number of routes that exceed the maximal allowed route duration (C4) the routing $DURATION\_REPAIR(PLAN)$ is called (c). 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. Third, it is tried to reduce the total travel distance by applying the 2-opt heuristic to each individual route contained in the proposal *PLAN* (d).

3.5 Genetic Search Procedure Overview

A memetic algorithm (MA) incorporates a genetic search framework and local search procedure. Fig. 8 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 replaced iteratively by *PopSize* new plans (i)-(ae) until the termination criterion is fulfilled. First, the so 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 selec-

(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. 8 Pseudo-code of the Memetic Algorithm

tion (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 that lead to 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. (C1) and (C3). 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 associated pickup and delivery operations). The labeled operations are shifted to another randomly selected route where all labeled operations are inserted between two 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 (C1) feasibility is still preserved.
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 (C1) as well as (C3) 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 routes. Again, (C1) as well as (C3) are respected.
5. A arbitrarily selected sub-route of a randomly chosen non-empty route is inverted but (C1) remains considered
6. A arbitrarily selected sub-route of a randomly chosen non-empty route is shifted inside the donating route (C1 remains considered).

4 Computational Experiments

4.1 Experimental Setup

In order to keep the variety of possible problem instance configurations on a manageable level, we have fixed the time period in which all vehicle operations are to be completed to $[0;10000]$. In addition, we have set the maximal allowed transfer time TT^{max} at the central transfer hub to 10000 time units.

We have generated several test instances seeded by the values $\{1;2;3;4;5\}$. The covered service area is the square $[-300;300]\times[-300;300]$. The central transfer hub is located at the origin $(0;0)$. Again, two separate non-overlapping service regions are defined. Region $\mathscr{A}$ comprises all locations in which the first component of the (x;y)-coordinates is <0 but Region $\mathscr{B}$ comprises all points in the aforementioned square which have an x-components >0.

The general set up comprises 25 orders, so that 50 requests are contained in a test case. 100 operations must be scheduled. We consider only $\mathscr{A}\mathscr{B}$- and $\mathscr{B}\mathscr{A}$-orders. The parameter $\rho\in\{50\%;75\%;100\%\}$ determinates the percentage of $\mathscr{A}\mathscr{B}$-orders and therefore expresses the balance of passenger flows between the two regions. Each order requires the movement of one person.

Table 2 Total Travel Distances

OD^{max}	TT^{min}	ρ		
		0.5	0.75	1.00
10000	0	6313	6023	3954
	100	6869	6222	3910
2000	0	8996	8466	6428
	100	9575	8939	6517

The available fleets in both regions contains 3 vehicles each. Both fleets have their depot at the central transfer hub so that all vehicle routes originates from there and terminates there. Since we want to focus on temporary aspects in the here reported experiments we do not consider scarce vehicle resources. For this reason, we assume that each vehicle has sufficient capacity.

Overall, we generate $\mid \{1;2;3;4;5\} \mid \cdot \mid \{50\%;75\%;100\%\} \mid = 5 \cdot 3 = 15$ test instances. We apply the proposed MA to each instance using different parameter settings for the least transfer time TT^{min} as well as the maximal OD-transfer time OD^{max}. We use the values $TT^{min} \in \{0;100\}$ as well as $OD^{max} \in \{2000;10000\}$. In total, we have now $15 \cdot 2 \cdot 2 = 60$ problem instances. Since the MA is a randomized procedure we repeat each problem instance processing with 3 different seedings so that 180 individual optimization experiments are conducted.

The MA maintains a population of 200 individuals, the crossover probability as well as the mutation probability are set to 100%. The genetic search trajectory evolution is terminated as soon as the average objective function of a population is not improved for 20 consecutive iterations. For the scheduling procedure, we use the generic vehicle sequence $0, 1, \ldots, 5$.

4.2 Report and Discussion of Results

Table 2 contains the averagely observed traveled distances. These values initiates the first analysis with the goal to find out which problem parameters have the most severe impact(s) on the objective function value. First, the increase of the minimal transfer time TT^{min} seems to influence the travel distance only slightly. At most 9% increase are observed. Second, the increase of the travel distance after a reduction of the maximal OD-transfer time OD^{max} is more severe. Here we observe an increase varying between 40% up to 67% if OD^{max} is reduced from 10000 down to 2000. Third, also the impact of moving the system from an imbalanced flow situation ($\rho = 1.00$) towards a balanced flow situation ($\rho = 0.50$) is remarkable. We observe an increase of the traveled distance between 30% and 76%. We conclude that the degree of imbalance as well as the allowed OD-transfer time have the most significant impact of the objective function value.

The analysis of the number of used vehicles (Table 3) reveals the following. First, sharping the synchronization constraints implies an increase in the number of needed vehicles. Second, in an imbalanced flow situation the number of needed vehicles tends to be higher compared to a situation with balanced OD-demand although the total traveled distances are less. Third, the difference between deployed type-$\mathscr{A}$-

Table 3 Number of Used Vehicles (type-$\mathscr{A}$;type-$\mathscr{B}$)

OD^{max}	TT^{min}	ρ		
		0.5	0.75	1.00
10000	0	2.07 (1.00;1.07)	2.14 (1.07;1.07)	2.07 (1.07;1.00)
	100	2.13 (1.00;1.13)	2.47 (1.07;1,40)	2.07 (1.07;1.00)
2000	0	4.00 (1.87;2.13)	4.00 (1.80;2.20)	4.20 (2.00;2.20)
	100	4.14 (2.07;2.07)	4.20 (1.87;2.33)	4.33 (1.93;2.40)

Table 4 Contribution of Waiting Time at Central Transfer Hub to Route Duration

OD^{max}	TT^{min}	ρ		
		0.5	0.75	1.00
10000	0	9%	19%	34%
	100	13%	24%	35%
2000	0	29%	30%	35%
	100	31%	33%	39%

Table 5 Average Transfer Time at the Central Transfer Hub (Squared Deviation)

OD^{max}	TT^{min}	ρ		
		0.5	0.75	1.00
10000	0	42 (36)	112 (138)	46 (66)
	100	270 (92)	261 (76)	141 (59)
2000	0	154 (78)	164 (90)	133 (66)
	100	250 (90)	252 (62)	259 (76)

Table 6 Average OD-Transfer Time (Squared Deviation)

OD^{max}	TT^{min}	ρ		
		0.5	0.75	1.00
10000	0	1845 (443)	2134 (428)	1954 (324)
	100	2146 (422)	2326 (470)	2053 (284)
2000	0	1339 (146)	1375 (164)	1281 (158)
	100	1387 (135)	1399 (162)	1392 (184)

vehicles and type-$\mathscr{B}$-vehicles increases as soon as an imbalanced flow situation turns into an imbalanced flow situation or if the maximal OD-transfer time is reduced.

The contribution of waiting times to the total route duration is worse in the situation of imbalanced passenger flow (more than 30%) as shown in Table 4.

In case of a balanced ($\rho = 0.50$) or semi-balanced flow situation ($\rho = 0.75$) the averagely observed transfer time is higher than the transfer time observed in average in the one-direction flow situation ($\rho = 1.00$) which is concluded from the results presented in Table 5. Also the variation of the individual waiting time is higher then. We suppose that in the first two mentioned situations vehicle trips last longer if collection requests are commonly fulfilled with distribution requests by the same vehicle. However, this aspect requires further experiments and analysis. The resulting research will be reported in another manuscript currently in preparation. The average OD-transfer time seems to be maximal in case that the flow situation is undecided ($\rho = 0.75$) as shown in Table 6.

Table 7 Total Travel Distances observed with RV-(left) and with LR-vehicle scheduling sequences (right)

OD^{max}	TT^{min}		ρ			ρ	
		0.5	0.75	1.00	0.5	0.75	1.00
10000	0	6298	6054	4040	6285	6053	3996
	100	6800	6069	3945	6727	6204	4019
2000	0	9713	8922	6674	8843	8542	6717
	100	9580	9120	6398	9492	9221	7757

4.3 Impact of the Route Evaluation Sequence

With the goal to investigate the impact of the vehicle scheduling sequence during the scheduling determination, we have compared the generic vehicle sequence $0, 1, \ldots, 5$ with two additionally defined sequences. The first one, RV (short for *random vehicle sequence*), selects the next vehicle to be scheduled at random. The second one, LR (short for longest routes first), starts with the schedule determination of the route that terminates at the latest, followed by the route that terminates second latest, and so on.

Table 7 contains the travel distances observed in the two repeated computational experiments. We see that RV-sequences do not succeed in improving the objective function value. If the maximal transfer time OD^{max} is reduced ($OD^{max} = 2000$) and if inbound as well as outbound requests are balanced ($\rho = 50\%$) then LR-sequences outperform the generic sequence. We observe travel distance reductions in these experiments. One reason might be that the number of actually deployed vehicles is significantly larger than 1 for each region so that a sequence variation lead to another schedule. If $OD^{max} = 10000$ then the final number of deployed vehicles is nearly one per region so that a variation of the scheduling sequence is useless. However, for $\rho = 1.00$ LR-sequence-based search fails to outperform the generic scheduling sequences. We do not know the reason for this result and will investigate this issue in upcoming experiments.

5 Conclusions

We have presented the DARP-TSC that represents the typical dispatching scenario in an on-demand public transport system if temporal customer convenience restrictions are to be considered. These restrictions let the decision models become so complex that only quite small instances (6 orders with 6 vehicles) can be solved to optimality using modern solver technology in an acceptable time less than one hour. In order to enable the investigation of larger scenarios we have developed a genetic search-based metaheuristic approach. Within computational experiments we have demonstrated its general appropriateness to generate high quality solutions of larger DARP-TSC instances with 25 orders in less than 5 minutes.

The major observation is that the impact of the extend of least guaranteed transfer time TT^{min} is rather small. Both the maximal acceptable OD-transfer time OD^{max} as well as the transport flow equilibrium degree in a region are more important for the actually achieved travel distance sum.

Future investigations will address issues related to the understanding of the impacts of different vehicle scheduling sequences. Here, instances with a larger number of orders as well as vehicles will be investigated. In addition, the impacts of limiting the maximal transfer time at the central hub is subject of already initiated computational investigations.

References

Archetti C, Speranza MG (2008) The Vehicle Routing Problem, Springer New York, chap The Split Delivery Vehicle Routing Problem: A Survey

Bredström D, Rönnqvist M (2008) Combined vehicle routing and scheduling with temporal precedence and synchronization constraints. European Journal of Operational Research 191:19–31

Cordeau JF, Laporte G (2007) The dial-a-ride problem: models and algorithms. Annals of Operations Research 153:29–46

Crainic T (2000) Service network design in freight transportation. European Journal of Operational Research 122:272–288

Deleplanque S, Quilliot A (2013) Transfers in the on-demand transportation: the darpt dial-a-ride problem with transfers allowed. In: Proceedings of the 6^{th} Multidisciplinary International Conference on Scheduling: Theory and Applications

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

Drexl M (2014) Branch-and-cut algorithms for the vehicle routing problem with trailers and transshipments. Networks 63:119–133

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

Grefenstette J (2000) Rank-based selection. In: Evolutionary Computation 1, IoP Publishing Ltd., Bristol

Masson R, Lehuédé F, Péton O (2014) The dial-a-ride problem with transfers. Computers and Operations Research 41:12–23

Mues C, Pickl S (2005) Transhipment and time windows in vehicle routing. In: Parallel Architectures, Algorithms and Networks, IEEE

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

Schönberger J (2005) Operational Freight Carrier Planning. Springer, Berlin Heidelberg

Schönberger J (2015a) Synchronisation in transportsystemen - fahrzeugübergreifende ablaufplanung fr die erfüllung komplexer serviceanforderungen im straengüterverkehr. Industrie Management 2:15–18

Schönberger J (2015b) The two-commodity capacitated vehicle routing problem with synchronization. In: Proceedings of 2015 IFAC Symposium on Information Control in Manufacturing, Elsevier, Amsterdam, to appear

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