European Journal of Operational Research 225 (2013) 106–117

Contents lists available at SciVerse ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Innovative Applications of O.R.

Synchronized dial-a-ride transportation of disabled passengers at airports

Line Blander Reinhardt [a,1], Tommy Clausen [a,b,2], David Pisinger [a,*]

[a] The Technical University of Denmark, Management Engineering, Management Science, Produktionstorvet 424, DK-2800, Kgs. Lyngby, Denmark
[b] SITA WorkBridge A/S, Hauser Plads 18, 1127 Copenhagen K, Denmark

ARTICLE INFO

Article history:
Received 8 September 2011
Accepted 3 September 2012
Available online 24 September 2012

Keywords:
Airport operations
Transportation planning
PRM
Heuristics
Simulated annealing
Dial-a-ride

ABSTRACT

The largest airports have a daily average throughput of more than 500 passengers with reduced mobility. The problem of transporting these passengers is in some cases a multi-modal transportation problem with synchronization constraints. A description of the problem together with a mathematical model is presented. The objective is to schedule as many of the passengers as possible, while ensuring a smooth transport with short waiting times. A simulated annealing based heuristic for solving the problem is presented. The algorithm makes use of an abstract representation of a candidate solution which in each step is transformed to an actual schedule by use of a greedy heuristic. Local search is performed on the abstract representation using advanced neighborhoods which modify large parts of the candidate solution. Computational results show that the algorithm is able to find good solutions within a couple of minutes, making the algorithm applicable for dynamic scheduling. Moreover high-quality solutions can be obtained by running the algorithm for 10 minutes.

1. Introduction

Around 1% of all passengers arriving at an airport need special assistance. Such passengers may be passengers returning from vacation with an injury, elderly or weak passengers, blind and deaf passengers, and passengers with other disabilities. We will refer to passengers needing assistance as passengers with reduced mobility (PRM). At the 31st biggest airport London Gatwick there was a throughput of 32 million passengers in 2009, of which around 900 each day needed assistance [10].

The support provided for the PRMs can be dedicated transport through the airport, and assistance at boarding. When assisting PRMs through an airport the PRM is picked up at the arriving location, e.g. check-in or gate of arrival, and delivered at the destination location, e.g. arrival hall or gate of departure.

Airports often have several terminals. At the studied airport the transport between the terminals is done in special buses solely for PRMs. Such buses will have a specific location for picking up PRMs at each terminal. Moreover, for aircrafts not located at a gate, the PRM will be transported in a special bus between the gate and the aircraft. Therefore, the pickup and delivery of a PRM is represented as a number of pickup and delivery segments. The airport and airlines require that the PRMs are not left alone at any point during their journey through the airport, and the PRMs are required to be in their assigned flight seat at a fixed pre-specified time before departure. However, the PRM may be left alone for a while before boarding at the departing terminal in a supervised area. It may be possible to assist more than one PRM at a time depending on whether they are able to walk and orient themselves or the PRM is in a wheelchair. Each PRM is assigned a weight depending on their disability and the personnel and vehicles are assigned a capacity depending on their type.

Given a fixed set of transporters, the objective is to minimize the number of PRMs not delivered and to minimize the total unnecessary travel time used on the journeys. PRMs which cannot be delivered on time must be scheduled for a later flight.

The studied airport requires to receive a solution within 2 minutes. One reason is that at airports the schedule often changes due to weather, breakdowns and other issues, therefore a new solution is needed several times during the day of operation. Small changes can be updated manually, while changes affecting many flights and passengers often require a complete reschedule.

We view the problem of assisting PRMs as a dial-a-ride problem (DARP), which is a generalization of the pickup and delivery problem (PDP).

The DARP is NP-hard by reduction from the Hamiltonian cycle problem see Baugh et al. [1].

Between each delivery and pickup of a PRM the transporter delivering the PRM must meet the transporter picking up the PRM. This vehicle synchronization is called a temporal dependency, therefore the problem is a dial-a-ride problem with temporal dependencies (DARPTD). The concept of synchronization in

* Corresponding author. Tel.: +45 45254555.
E-mail addresses: lbre@man.dtu.dk (L.B. Reinhardt), tcl@workbridge.com (T. Clausen), pisinger@man.dtu.dk (D. Pisinger).
[1] Tel.: +45 45253389.
[2] Tel.: +45 33447500, +45 45253389.

0377-2217/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ejor.2012.09.008

routing was used by Ioachim et al. [11] for the fleet assignment problem and later extended to the more general temporal dependencies by Dohn et al. [9]. In pickup and delivery problems the similar problem of cross docking has been considered. In cross docking there is a transfer of goods between vehicles at the synchronized points. The pickup and delivery with cross docking is used in supply chain and planning city logistics systems [2,7]. Pickup and delivery with cross docking was studied by Wen et al. [18] and Chen et al. [2]. In the cross docking problems the cross docking is optional for the vehicles. This is not the case in the problem of assisting PRMs at an airport, as the cross-docking points for each PRM are known and fixed. Moreover, even though cross docking problems often include a cost for the time the transshipped items spend at the cross docking facility, there is no requirement of synchronization. In the cross docking example considered by Chen et al. in [2] the demand is not a single pickup and delivery location pair. Instead, the demand is represented by a source and a sink for a given product and therefore the demand can be picked up at several different locations or delivered to several different locations. In the cross docking problems considered in [2,18] there will be at most one cross docking between the pickup and delivery of a resource. This also differs from the problem discussed in this paper as up to four synchronization points can be included in a transit journey through the airport. Other closely related problems are the pickup and delivery problem with transfers solved by Cortes et al. [6] in 2010 and the pickup and delivery with transfers and split delivery solved for liner shipping by Reinhardt and Pisinger [16] in 2011. The PDP with transfers described in [6] contains a limited number of transfer points where the vehicles are synchronized. A model is presented and instances with up to 6 requests, 2 vehicles and 1 transfer point are solved to optimality using a combinatorial Benders decomposition method. The problem considered by Reinhardt and Pisinger in [16] does not consider synchronization at transfer points and is solved by branch and cut for instances a little larger than those of Cortes et al. [6].

The synchronization constraints and the objective also separate the transportation of PRMs in airports from the rich pickup and delivery problem described in [15]. In the survey by Cordeau and Laporte [3] from 2007 a list of some of the methods used for the dial-a-ride problem with multiple vehicles is provided. In this list the only exact methods are, a branch and cut method optimizing on vehicle travel cost by Cordeau [5], and an improvement on this method by Ropke et al. [17]. The exact method has been tested on a maximum 96 requests and 8 vehicles, which was solved in 71 minutes.

The dial-a-ride problems are usually solved by heuristics, as the studied problems often are real-life cases. Real life problems generally contain some additional constraints, which can be complicating and the objective varies. Moreover, in real-life, there can be constraints or desires not defined in the problem, and the size of the problems is often large. Due to this an optimal solution may actually not be the best solution for the users.

Since the problem covered here is a dial-a-ride problem with complicating synchronization constraints and contains a large number of requests, vehicles and transfer locations, it is natural to consider heuristic solution methods. Moreover, there is a solution time requirement of 2 minutes given by the studied service provider. When solving instances with between 900 and 1500 requests within 2 minutes a heuristic method seems to be the only option.

For more details on the definition of DARP see Cordeau and Laporte [3] from 2007 and the more recent paper by Parragh et al. [14] where the variable neighborhood search heuristic is applied to a standard formulation of the dial-a-ride problem. Parragh et al. [14] report competitive solution for problems with up to 144 request based on the test sets delivered by Cordeau and Laporte [4]. Jaw et al. [12] in 1986 report finding good solutions to their dial-a-ride problem on an instance with 2617 requests and 28 vehicles using an insertion sort method. When run on present computers the method would satisfy the solution time requirement. Other heuristic methods, which are able to find good solutions for dial-a-ride problems with a large number of requests, are the regret insertion method by Diana and Dessouky [8] solving problems with 1000 requests, and a local search heuristic by Xiang et al. [19] solving problems with 2000 requests.

In this paper we present a local search heuristic for the specific problem based on simulated annealing. The algorithm makes use of an abstract representation, which is transformed to an actual schedule by use of a greedy heuristic. Local search is performed on the abstract representation using large neighborhoods. In each iteration, the resulting candidate solution is evaluated and accepted according to the standard criteria in simulated annealing. Computational results are reported showing that the algorithm is able to construct high-quality solutions in 10 minutes.

The model and algorithm can be used for other dial-a-ride problems with synchronization. An example of such a problem is the transport of patients for surgery at hospitals. Here, several steps occurs such as transport to the anesthesia area and then the transport to the surgery room, and from there to the "wake-up" room. During this procedure the patient is not to be left alone at any time. Other examples could be transports involving liner vessels and freight trains. Here, synchronization is needed at the port or station in order to avoid excessive costs due to crane rentals and space limitations in the unload-area.

The main contribution of the paper is to present a highly relevant multi-modal transportation problem. With still more passengers traveling by air, we may expect increasing need for transport planning of passengers with reduced mobility. The problem is a true multi-modal transportation problem with synchronization, which may be used to model several other coupled pickup-and-delivery problems appearing in real-life problems. Finally, the proposed local search based on an abstract representation is generally applicable for other tightly constrained problems, and it shows promising results for the considered instances.

This paper is organized as follows. Section 2 contains a detailed problem description to ensure a thorough understanding of the operational process. In Section 3 a mathematical model of the problem is presented. Section 4 presents the solution method used. Section 5 contains the specifications of the data instances received. In Section 6 the tuning of the parameters for simulated annealing is described. Section 7 contains the results of the solution method applied to the real-life instances received. Finally in Section 8 the results and future work are discussed.

2. Problem description

We will denote the considered problem the *Airport passenger with reduced mobility transport problem* (APRMTP). The APRMTP has been defined in cooperation with a service company providing the assistance for PRMs at a major transit airport. The company has 120 employees assisting between 300 and 500 PRMs through the airport each day. Each employee has a pre-specified working area such as a specific terminal, driving between the terminals bus stop locations or driving between aircrafts and gates. An employee assigned to one area may not move into another area. Therefore, the journey of the PRM is split into a pickup and delivery for each of these areas. We call the pick up and delivery in a specific area for a *segment* and the ordered set of segments of a given PRM for a *journey*. The path of a bus or a foot personnel is referred to as *route*. On average there are three segments per PRM, and hence with 300–500 PRMs each shift we get a total of between 900 and 1500 pick up and delivery segments. This also includes assistance

Table 1
List of constraints which are additional to the usual constraints in the standard dial-a-ride problem. The constraints often originate from a service contract agreement between the service provider, the airport and the airlines.

	Additional constraints
1	An arriving PRM must be picked up exactly upon arrival
2	A terminal transfer is done by bus between the bus stop locations of the terminals
3	The PRM may not be left unsupervised
4	The PRM must not use more than 30 minutes of excess time on each segment
5	Embarkment lasts 20 minutes and cannot start earlier than 60 minutes before departure
6	The PRM must be seated in the aircraft exactly 20 minutes before departure

when boarding, which we have represented as a pickup and delivery request with special conditions. We will refer to the boarding assistance as *embarkment*. The employee assigned to an embarkment cannot go to another location between pickup and delivery of the embarkment segment. However, an employee may assist as many PRMs embarking onto the same flight as their capacity allows.

It should be noted that all of the pickup and delivery locations for every segment of the journey are predetermined.

As mentioned in the previous section, the PRM may not be left alone except at special supervised areas located in the departing terminal. This means that the employee delivering the PRM to a bus must wait with the PRM for the bus to arrive before being able to initiate the next task. The bus also has to wait for the employee to come and pick up the delivered PRM before continuing the route.

The company wants to make sure that they deliver the best service possible with the given number of employees and the current employee working area assignments. The PRMs are split into two categories: Those who are prebooked and those who are immediate. The prebooked PRMs order the service when purchasing the ticket or at least a few days in advance. Immediate PRMs request the service at check-in. If they are departing from the airport, the request is not known before the moment it has to be carried out. Immediate PRMs arriving on flights may be known down to half an hour before flight arrival. It is not always possible for the company to assist all PRMs and in such cases the prebooked PRMs have higher priority. The company also wishes to minimize unnecessary time the PRM spends on the journey segments. This could be time spent waiting to be picked up by the employee of the succeeding segment or extra travel time caused by picking up or delivering other PRMs before being delivered. Note, that the time spent at the supervised area of the departing terminal is not included in the service evaluation. The unnecessary time spent on the segments is called excess time. Clearly, when minimizing the overall traveling time there is a risk of having a few PRMs with very large traveling times. Therefore, it is important to limit the traveling times of the different segments so that the journey never becomes very unsatisfactory.

Many additional constraints concerning the pick-up and delivery times, assistance at embarkment, and transport to and from aircrafts not located at a gate, are imposed by the airport and airlines. Such constraints are listed in Table 1. The last two constraints in the table mean that embarkment cannot start later than 40 minutes before departure. When the aircraft is parked away from gate the embarkment must start even earlier than 40 minutes before departure.

Note, that the person assisting the PRM through embarkment will not be able to leave the PRM until 20 minutes before departure if the aircraft is located at the gate, or before the PRM is picked up by a special vehicle if the aircraft is located away from the gate. In the latter case the special vehicle cannot leave the PRM before 20 minutes before departure.

The different transportation forms such as vehicles and assistance on foot have different capacities. Moreover the PRMs are assigned a volume depending on their disability. For example it is very hard for one person to push two wheelchairs, while two hearing or sight impaired persons easily can be assisted by the same employee.

2.1. Example of a journey of a PRM

The most complex example of a journey is the case where a PRM makes a transit from an arriving aircraft not located at the gate to a departing aircraft in another terminal not located at the gate. Such a case is shown in Fig. 1. We say that such a journey has six segments. All transit journeys are formulated as an ordered subset of these segments. Non-transit PRMs are either picked up or delivered to a public area in the terminal of their flight.

The algorithm presented in this paper is to be used in a dynamic setting, where immediate PRMs arrive continuously and disruptions in the daily plan such as flight delays frequently occur. Therefore, the company desires to receive a solution to the problem within a couple of minutes. We do not consider robustness and break times. However, robustness can be obtained by introducing buffer time in the time to get from one location to another and by altering the set of available employees. Breaks can be included by splitting the shift of an employee into several shifts.

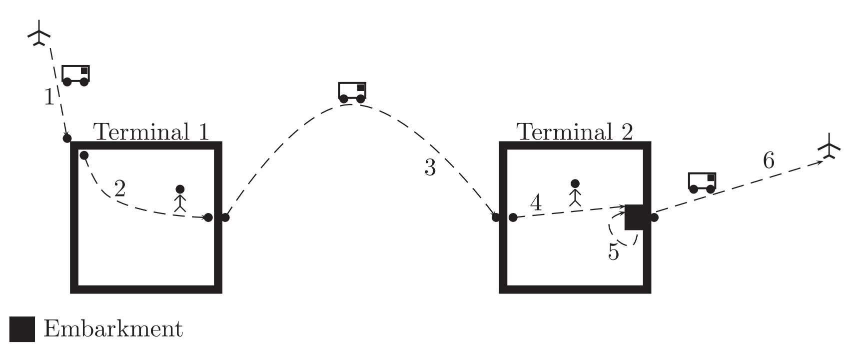


Fig. 1. Illustration of a journey with six segments: (1) the PRM is picked up at arriving aircraft by vehicle at the exact time of arrival and delivered at gate, (2) the PRM is picked up at the gate by an employee. The bus in segment 1 cannot leave the gate before the PRM is picked up. The PRM is delivered at the bus stop for the inter-terminal busses, (3) The PRM is picked up by an inter-terminal bus and delivered to the bus stop at the terminal of the departing flight (the employee of segment 2 cannot leave before the bus arrives for pickup), (4) the PRM is picked up by an employee and delivered to the gate of the departing flight (the bus in segment 3 cannot leave before the employee arrives), (5) the PRM is assisted through embarkment at the gate. This task takes 20 minutes. For the time between segments 4 and 5 the PRM can be left at a special supervised lounge, (6) The PRM is picked up at the gate by a vehicle and delivered to the aircraft exactly 20 minutes before the departure time. The employee of segment 5 cannot leave before the vehicle arrives for pickup.

3. Mathematical formulation

When describing the model it is important to bear in mind that the journey of each PRM is a set of pickup and deliveries called segments. This means that a PRM is picked up and delivered if all the segments of the journey are handled. Therefore, we must for each PRM ensure that all his/her segments are assigned before we consider the PRM delivered. As common for dial-a-ride problems with a heterogeneous fleet, each segment is represented by a pickup vertex and a delivery vertex specific to that segment. Also, there is a location inside each working area where the employees start and end their shift.

3.1. Graph representation

As mentioned earlier there is a vertex for each origin and destination of a segment and the location of all the vertices in a journey are predetermined. Since we already know which transport group is assigned to each vertex we can generate a disjoint graph for each transport group. Each graph has its own depot where the transporter start and end their shift. These graphs are "virtually" connected by the synchronization of the segments of a journey. For each terminal we have a directed graph connecting the vertices that must be serviced by foot personnel working in the given terminal. The busses transporting PRMs between terminals have a directed graph of their pickup and delivery vertices. The vehicles transporting PRMs from gates to aircrafts have a directed graph connecting their pickup and delivery vertices. The journey shown in Fig. 1 has the segments 1, 2, 3, 4 on four disjunct graphs. Both segments 4 and 5 are on the graph representing Terminal 2 and both segments 1 and 6 are on the graph representing vehicles operating between gates and aircrafts. Connections between pickup and delivery vertices, which are infeasible due to their time windows, are removed from the graph.

3.2. Mathematical model

Given the following sets:

K	The set of transporters. Contains all vehicles and persons on foot
R	The set of segments. Contains all the segments of all the journeys
R_p	A subset of segments in R which contains all the segments for PRM p
C	The set of all PRMs
B	The subset of C which contains all the prebooked PRMs
F	The set of departing flights
V^*	The set of pickup and delivery vertices and depots/bases
V	The subset of V^* containing all the pick up and delivery points/vertices
V_f	The subset of V containing all the embarkment vertices for flight $f \in F$
P	The set of pickup vertices, $P \subseteq V$ and P_p is the pickup vertices of PRM p
D	The set of delivery vertices, $D \subseteq V$ and D_p is the delivery vertices of PRM p
E	The set of edges connecting the elements in V^*
λ_p	The set (i,j) for a journey of PRM p where $i \in D_p$ and $j \in P_p$ are the delivery and pickup at a transfer point on the journey of p
δ	The set of vertex pairs that must be synchronized for handover

Each segment $r \in R$ has a start o_r and a destination d_r and the set V is all of the different o_r and d_r vertices. Each work area has a start point v_0 and an end point v_e representing the location, where the transporters start and end the day.

We define the following parameters:

M_b	The penalty for not transporting a prebooked PRM
M_n	The penalty for not transporting an immediate PRM
t_r	The minimum time needed to deliver segment $r \in R$
t_{ij}^k	The minimum time it takes to go from i to j on transport k
l_j'	The change in load at vertex $j \in V$
H	The maximum excess time allowed to be used on a segment
C_k	The capacity of transporter $k \in K$
M	A big constant being at least as large as the shift length
M_s	A big constant larger than the largest number of segments in any PRM
M_l	A big constant at least as large as the biggest capacity plus the largest volume possible for any PRM
a_i	The release time at vertex $i \in V^*$
b_i	The due time at vertex $i \in V^*$

We use the following variables:

s_i^k	The time when transporter k leaves vertex i
ϕ_p	An indicator variable indicating if a PRM $p \in C$ has a segment not assigned. ϕ_p is 0 if all segments of p are assigned and 1 otherwise
x_{ij}^k	An indicator variable indicating if the edge (i,j) is used by transporter k. x_{ij}^k is 1 if the edge is used by transporter k and 0 otherwise
l_i^k	The load on transporter k when leaving vertex i

As objective we have chosen a linear weighted combination of assigning as many PRMs as possible and minimizing the total excess time the PRMs spend on their journey. The model is:

$$\min \sum_{d \in R} \left(\left(\sum_{k \in K} s_{d_r k} - s_{o_r}^k \right) - t_r \right) + M_b \sum_{p \in B} \phi_p + M_n \sum_{p \in C \setminus B} \phi_p \tag{1}$$

$$\text{s.t.}(\text{P equals D}) \quad \sum_{i \in V^*} x_{io_r}^k - \sum_{i \in V} x_{id_r}^k = 0 \quad r \in R,\ k \in K \tag{2}$$

$$(\text{Balance}) \quad \sum_{j \in V^*} x_{ij}^k - \sum_{j \in V*} x_{ji}^k = 0 \quad i \in V,\ k \in K \tag{3}$$

$$(\text{Start}) \quad \sum_{j \in V^*} x_{v_0 j}^k = 1 \quad k \in K \tag{4}$$

$$(\text{End point}) \quad \sum_{j \in V^*} x_{j v_e}^k = 1 \quad k \in K \tag{5}$$

$$(\text{P before D}) \quad s_{d_r}^k - s_{o_r}^k \geqslant 0 \quad k \in K,\ d \in R \tag{6}$$

$$(\text{Complete}) \quad M_s \phi_p - \sum_{d \in R_p} \left(1 - \sum_{k \in K} \sum_{j \in V} x_{o_r j}^k \right) \geqslant 0 \quad p \in C \tag{7}$$

$$(\text{Timelimit}) \quad s_{d_r}^k - s_{o_r}^k - t_r^k \leqslant H \quad k \in K,\ d \in R \tag{8}$$

$$(\text{Connect}) \quad s_i^k + t_{ij}^k + M(x_{ij}^k - 1) \leqslant s_j^k \quad k \in K,\ (i,j) \in E \tag{9}$$

$$(\text{Handover}) \quad \sum_{k \in K} s_i^k = \sum_{k \in K} s_j^k \quad (i,j) \in \delta \tag{10}$$

$$(\text{Journey}) \quad \sum_{k \in K} s_i^k \leqslant \sum_{k \in K} s_j^k \quad (i,j) \in \lambda_p \tag{11}$$

$$(\text{Release}) \quad a_i \leqslant s_i^k + a_i \left(1 - \sum_{j \in V} x_{ij}^k \right) \quad i \in V^*,\ k \in K \tag{12}$$

$$(\text{Due})\ \ b_i \geqslant s_i^k + b_i\left(1 - \sum_{j\in V} x_{ji}^k\right)\ \ i \in V^*,\ k \in K \tag{13}$$

$$(\text{Load})\ \ l_i^k + l_j' - M_l\left(1 - x_{ij}^k\right) \leqslant l_j^k\ \ (i,j) \in E,\ k \in K \tag{14}$$

$$(\text{Capacity})\ \ l_i^k \leqslant C_k\ \ i \in V,\ k \in K \tag{15}$$

$$(\text{Embarkment})\ \ \sum_{k\in K} x_{ij}^k = 0\ \ j \in V \setminus V_f,\ i \in P \cap V_f,\ f \in F \tag{16}$$

$$(\text{Embarkment load})\ \ l_i^k - C_k\left(1 - \sum_{j\in V\setminus V_f} x_{ij}^k\right) \leqslant 0\ \ k \in K,\ i \in D \cap V_f,\ f \in F \tag{17}$$

$$x_{ij}^k \in \{0,1\}\quad k \in K,\ \ (i,j) \in E \tag{18}$$

$$\phi_p \in \{0,1\}\quad p \in C \tag{19}$$

$$s_i^k \in \mathbb{R}_0^+\quad i \in V \cup \{p_k\},\ \ k \in K \tag{20}$$

$$l_i^k \in \mathbb{Z}_0^+\quad i \in V,\ \ k \in K \tag{21}$$

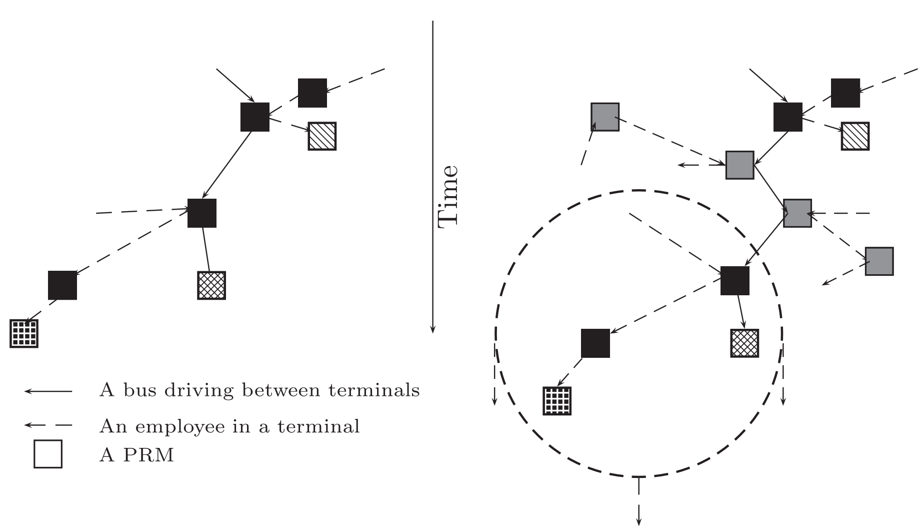
The objective function (1) sums all the excess time used on the segments and adds a penalty if a PRM is not delivered. Different penalties are used depending on whether PRM p is prebooked ($p \in B$) or immediate ($p \in C \setminus B$). Constraints (2) ensure that for each segment any PRM picked up is also delivered. Constraints (3) ensure that transporters leave all pickup or delivery vertices they enter. Constraints (4) ensure that all transporters leave their base. Constraints (5) ensure that all transporters return to their base. Constraints (6) ensure that on each segment a PRM is picked up before it is delivered. Constraints (7) ensure that any PRM with at least one segment not assigned generates exactly one penalty in the objective. Constraints (8) ensure that the excess time used on segment d does not exceed the maximum excess time H. Constraints (9) ensure that if an edge (i,j) is used then the time at which vertex j is visited is not earlier than the time leaving vertex i plus the travel time on edge (i,j). Constraints (10) ensure that the time of delivering transporter and pickup transporter are synchronized at their respective vertex points of $(i,j) \in \lambda$. Constraints (11) ensure that the segments of the journey are completed in the right order. Constraints (12) and (13) ensure that the segments are started and ended within their given time window. Constraints (14) update the load when a PRM is picked up or delivered. Note that since load is increased for any pickup vertex in V, constraints (14) together with constraints (9) ensure a connected route. This is true under the general assumption that the transport from pickup to the delivery point is always greater than zero. Constraints (15) are the capacity constraints. Constraints (16) and (17) enforce the embarkment conditions of only starting embarkment tasks on the same flight before completing an embarkment segment. Constraints (16) only allow edges going from a pickup vertex of an embarkment segment to vertices of embarkment on the same flight. Constraints (17) ensure that when using an edge between an embarkment vertex and any vertex not belonging to an embarkment request for the same flight the load must be zero.

4. Solution method

The solution method we present is a greedy insertion heuristic combined with simulated annealing. The synchronization constraints present in the APRMTP add complexity to the generation of feasible solutions and the calculation of the objective value.

When a segment is inserted in a route of a bus or employee, it may influence not only the travel times of the later segments in the route inserted in, but also other of the PRMs transported by the route and segments of the routes of the personnel serving those segments and so forth. This means that every time the end or start time of a segment is changed it may generate a cascade of changes on related segments. Therefore, when checking for feasibility one may, in worst case, have to evaluate the feasibility of all the segments in the problem. The same is true for calculating the objective as the insertion of a segment may influence the travel time on all remaining segments in all the routes. Therefore, an easy update of the objective when inserting a segment is not evident. However, together with a constraint on the maximum excess time allowed on each segment it may also constrain the problem significantly. An example of this is that the synchronization constraint reduces the number of possible feasible solutions.

We have constructed a greedy insertion heuristic for the initial solution, described in the next section, and later used in a simulated annealing scheme to improve solutions. The greedy insertion heuristic will, given an ordered list of PRMs, lead to a deterministic solution found with a search for best insertion spots, while the simulated annealing broadens the search by randomly selecting a



(a) Before insertion (b) After insertion of gray PRM

Fig. 2. A section of the routes when inserting the gray PRM. The part of the graph inside the dashed circle is moved to a later time because of propagated delays caused by the insertion of the gray PRM. The patterned PRMs are the next PRMs to be handled by the employee or bus.

neighborhood from a large neighborhood space and accepting solutions with a worse objective value. Due to the synchronization constraints the routes in the constructed solution are very interdependent and therefore it would be very difficult for a local search to find better solutions using neighborhood search. Instead of performing a local search on the constructed solution as done in GRASP we perform the local search on the abstract list representation of the previous solution.

Fig. 2 illustrates how PRM segments are moved when inserting a PRM segment into the route represented by the solid line.

4.1. Insertion heuristic

A greedy insertion heuristic (GIH) is created to quickly find a feasible solution. The heuristic takes two lists of PRMs, one containing the prebooked PRMs, and one containing the immediate PRMs. The insertion heuristic inserts first the PRMs from the prebooked list P_1, then the PRMs from the immediate list P_2 by going through the lists in sequential order. The reason for this is, that it is very important for the service provider to serve the prebooked PRMs.

We have sorted the lists by earliest pickup time of the PRM, starting at the earliest arrival time. For each PRM the segments are inserted in the order they appear in the journey and the next PRM is not inserted before all segments of the previous PRM are inserted.

For each segment the insertion place is found by only investigating the set of transporters working in the graph containing the given segment. The segment is inserted in the place and transporter where it creates the least increase in the total excess time. We only allow an insertion to push the time of the other segments forward. Therefore, when checking for feasibility, we only need to go through the segments with larger delivery times.

Usually, when minimizing the route cost, as in pickup and delivery problems, it is possible to calculate the new objective by the difference in the time introduced by the insertion and removal. However, in our case we did not do this as we found it too complicating when the excess time is minimized and the insertion can generate propagating delays induced by the synchronization constraints.

Algorithm 1. GIH(P_1,P_2)

1: **for** each PRM p taken first from list P_1 then from list P_2 **do**
2: **for** each segment s in p **do**
3: **for** each employee e serving s **do**
4: **for** each vertex v_1,v_2 in e in the possible time interval for s **do**
5: **if** load and time feasible insert s_{start} before v_1 and s_{end} before v_2
6: calculate total excess time for all inserted segments;
7: **end if**
8: **end for**
9: Select s_1 and s_2 where the least excess time is generated;
10: **end for**
11: **if** insertion was not possible **then**
12: Delete already inserted segments of p;
13: Register p as not inserted;
14: **else**
15: insert s in the e where the least excess time is generated;
16: **end if**
17: **end if**
18: **end if**

In the pseudo code of **GIH** the lines 1 and 2 are performed $O(n)$ times where $n = |R|$. The combinations that occur in lines 3 and 4 can be $O(n^2)$. Checking for feasibility in line 5 is done by a depth first search which goes through $O(n)$ vertices updating their start/end times and the edge load. The calculation of total excess time of all inserted segments in line 6 is done by adding up excess time of all inserted segments, which is $O(n)$. Therefore, the asymptotic running time of the greedy insertion heuristic is $O(n^4)$.

In the test cases provided by the company the time windows are quite tight and by definition only a subset of employees are available in the area of a given segment. Therefore, the number of feasible insertion points is limited before starting the propagated feasibility test and calculating the excess time.

4.2. Simulated annealing

A simulated annealing algorithm using the two lists of PRMs (prebooked and immediate) as an abstract representation was implemented. The initial solution is generated by the greedy insertion heuristic on the two lists of PRMs, sorted by earliest possible start time. At each iteration of the simulated annealing algorithm, a number of moves takes place to obtain a candidate solution x from the current solution x'. The moves are made in the two PRM lists, which are then converted to a solution by the insertion heuristic. The moves are as follows:

- moving a not assigned PRM a random number of places forward in the respective list
- swapping the place of two PRMs randomly selected within the same list.

Note, that the prebooked PRMs and immediate PRMs will always remain in their respective lists. The lists are, when the moves are completed, converted by the greedy insertion heuristic into a schedule.

Let the objective function be defined as $f(x)$ for a solution x. If $f(x) \leqslant f(x')$ then x is accepted. Otherwise we accept the solution with probability:

$$\exp^{(f(x')-f(x))/T} \tag{22}$$

Details on the selected values for the temperature T will be covered in Section 6 on tunings. The temperature is decreased by a selected factor at each iteration. This decreasing factor is also called the cooling rate. The described large neighborhood and the acceptance probability allow the algorithm to escape local minima.

5. Data instance and other parameter values

We have received 12 test cases from the service company covering dates September 20 to October 1, 2009. These test cases contain a total of 5000 PRM requests with information about the type and the position of the origin and destination. Travel time between the locations for the different transport forms were calculated from this information. Each test case represents a day and contains between 374 and 555 PRMs. The company delivers service in the majority of the airport, but also other service providers are present in some parts of the airport. Therefore, the test cases do not cover all PRM requests at the airport.

Some of the PRMs in the data set were removed before running the tests due to corrupted data for the PRM or due to insufficient time available for the PRMs journey so that a solution transporting the PRM cannot exist satisfying the requirements given. This resulted in sets of between 353 and 495 PRMs. Each data set had a set of employees for the given day with a assigned terminal or vehicle. The number of employees assigned on a day was around

120. The employees were assigned to six different terminals and two different bus types. The two bus types each has a specific area of operation.

The capacity of the transporters has been settled with the ground handling company as:

- An employee assisting inside a terminal has capacity 2
- A bus between terminals has capacity 12
- A bus between gate and aircraft has capacity 9

For each PRM there is given a start time, an end time, a start location, an end location and a PRM type. There are six different type of PRMs in the data sets. For each type of PRM we have assigned a volume as follows:

WCHC Cannot walk or stand, needs wheel chair. Volume 1.5
WCHS Cannot walk up or down stairs. Volume 1
WCHR Cannot walk long distances. Volume 1
BLND Passenger is blind. Volume 1
DEAF Passenger is deaf. Volume 1
ASS Passenger cannot orient themselves. Volume 1

From the airport structure described to us by the company we have generated the segments for each PRM given the arrival and departure location of the PRM. For each shift between 900 and 1500 segments were generated. The excess time allowed for each segment is $H = 30$ minutes as this generally matches the requirement at airports. We have chosen the weights in the objective function as follows:

$$M_b = 1000, \quad M_n = 400$$

This is to ensure a strong priority to the prebooked PRMs. Moreover, since a PRM journey has at most six segments, all PRMs must have an excess time of at most 180 minutes. Therefore, delivering any PRM will always be prioritized over a faster delivery and even several faster deliveries.

6. Tuning

The Simulated Annealing algorithm is tuned using three test cases selected from the received data sets. Theses instances represent the 20th of September (2009), 26th of September (2009) and the 1st of October (2009).

Since our insertion algorithm has a complexity of $O(n^4)$, the number of iterations completed in simulated annealing during the 2 minutes running time is limited to a few hundreds. This means that for most instances the problem contains more PRMs than iterations performed during simulated annealing. Therefore, we consider the possibility for making several moves in each iteration. The moves are relocations in the list and do not influence the running time significantly. However, the neighborhood becomes much larger and the previous solutions may be ruined by a large number of moves.

We have included the initial objective value in the generation of the initial temperature to make the acceptance rate robust to variances in the size of the different problems. Thus the initial temperature is adjusted so that the probability of selecting a solution, which is exactly t percent greater than the initial solution is 50%. To find the best combination of the cooling rate and the number of moves, we have fixed the initial temperature so that a solution 5% worse than the initial solution must initially be accepted with 50% probability. This means that given an initial solution x and the temperature parameter of t, the initial temperature T is calculated as follows:

$$T = -tx/\log(0.5) \tag{23}$$

In Section 7 the algorithm is tested with the required solution time of 2 minutes, a slightly larger solution time of 5 minutes and a large solution time of 1 hour. To accommodate these tests the simulated annealing parameters are tuned both for a solution time of 2 minutes and a solution time of 1 hour. It is presumed that the values of the 5 minutes running time are close to the ones for 2 minutes and therefore we have not tuned for 5 minutes.

The algorithm is run 10 times and the average solution of the 10 runs is found. The average solution is then used to calculate the percent wise gap between the initial solution and the average solution found by simulated annealing. This is calculated as follows:

$$impr = \frac{initsolution - average\ solution}{initsolution} \cdot 100$$

Note, that by using the average solution the impr represents the expected improvement for a single run of the simulated annealing algorithm with the same solution time requirement.

We test a combination of different number of moves (4,12,20) and different cooling rates (0.5,0.8,0.9,0.95,0.99) given a fixed initial temperature T (using $t = 0.05$ in Eq. (23)). These tests are run with a time limit of respectively 2 minutes and for 1 hour for the three selected instances. The tuning results can be found in Appendix A.

For runs with time limit of 2 minutes, analyzing the gap of the test for combinations of cooling rate and number of moves, we have selected the value 0.9 for the cooling rate and 12 moves at each iteration. For runs with time limit of 1 hour, applying the same method, we have chosen the value 0.99 for the cooling rate and four moves at each iteration.

The selected values are used in the tuning tests on the initial temperature for 2 minutes and 1 hour time limit.

The initial temperature T is tested with the values of $t = \{0.01, 0.05, 0.1, 0.15\}$ in Eq. (23). For 2 minutes runs the best solution is obtained when choosing the initial temperature so that a solution 5% greater than the initial value is accepted with 50% probability. These values will be used in Section 7 for tests with solution time requirement of 2 and 5 minutes. For 1 hour runs the best initial solution is obtained by choosing the initial temperature so that a solution 1% greater than the initial value is accepted with 50% probability. These values will be used for the tests in Section 7 with solution time requirement of 1 hour.

The values found in this section for the simulated annealing determines the intensification and diversification of the heuristic. The high number of moves at each iteration and the acceptance probability ensures diversification while the cooling rate gives rise to intensification over time. The start temperature determines the start diversification generated by the acceptance probability.

7. Test results

The 12 test cases received from the service company covers the dates September 20 to October 1, 2009. Unfortunately, the company does not have records on how the PRMs actually were scheduled for the test cases. Hence, we cannot directly compare our schedules with the historic data as the service company does not currently make a plan of the day. Moreover from Cortes et al. [6] it can be seen that for a similar problem only tiny problems are so far solvable with exact method. In [6] problem sizes with up to 6 requests 1 transfer point and 2 vehicles is solved using both CPLEX and combinatorial Benders cut method.

It should be noted that in the test cases a little more than half of the PRMs are prebooked. The test cases are evaluated individually and therefore we include the three test cases used for tuning in this test section.

Table 2
The results of the insertion heuristic run on the test cases. Note, that this is also the initial solution used by the simulated annealing heuristic.

Case	PRMs	PRMs deleted	NAP	NAI	Sol.
20090920	374	21	0	4	3566
20090921	426	35	7	5	11,430
20090922	451	23	0	5	5162
20090923	403	42	0	0	2001
20090924	465	33	0	3	4062
20090925	484	46	0	2	3500
20090926	401	27	0	0	1782
20090927	429	32	0	2	3095
20090928	401	23	0	0	2208
20090929	456	37	0	1	3259
20090930	555	60	7	9	13,443
20091001	519	45	1	1	4401

The tests were run on a computer with a 64 bit Intel Xeon 2.67 gigahertz CPU. The simulated annealing tests reported in this section with time requirement of 120 seconds and 300 seconds have been run with:

- *Temperature*: we use the factor $t = 0.05$ to adjust the temperature T according to Eq. (23)
- *Cooling rate*: 0.9
- *Moves at each iteration*: 12

The simulated annealing tests reported in this section with time requirement of 1 hour have been run with:

- *Temperature*: we use the factor $t = 0.01$ to adjust the temperature T according to (23).
- *Cooling rate*: 0.99
- *Moves at each iteration*: 4

Note, that the temperature is calculated given an initial solution x and the temperature parameter of t as described in Eq. (23).

In Table 2, we report the results of running the greedy insertion heuristic on the test case where the PRMs are sorted by earliest arrival time for all 12 data sets.

In column one the name of the data set is described by the date of the shift. We report how many PRMs in the given data set we had to remove due to corrupted data. The number of prebooked passengers not assigned (NAP) and not assigned immediate passengers (NAI) in the initial solution are reported in respectively column four and five. Finally, the initial solution retrieved from the greedy insertion heuristic on the PRM lists is reported in column six.

From Table 2 it can be seen that no prebooked PRM were rejected for 9 of the 12 instances. The number of rejected immediate PRMs is also quite low compared to the number of PRMs when solving the problem using just the greedy insertion heuristic.

The results in Table 2 are used for evaluating the test results of the simulated annealing algorithm presented in Table 3.

In Table 3 the simulated annealing is tested for 120 seconds (2 minutes), 300 seconds (5 minutes) and 3600 seconds (1 hour) on each test case as given in column two. Each test is repeated

Table 3
The results of simulated annealing running for 2 minutes, 5 minutes and 1 hour.

Case	Time (seconds)	Av. it	Std. D	Av. NAP	Av. NAI	Best sol.	Av sol.	Impr (%)
20090920	120	279.5	61	0	2.0	2423	2504.6	29.8
20090920	300	767.6	52	0	2.0	2361	2426.9	31.9
20090920	3600	8615.2	28	0	2.0	2110	2173.1	39.1
20090921	120	209.0	369	5.6	3.6	8227	9026.0	21.0
20090921	300	499.4	430	5.6	3.0	8112	8658.3	24.2
20090921	3600	4813.4	390	5.1	2.4	7181	7475.3	34.6
20090922	120	131.1	315	0	1.3	2922	3233.5	37.4
20090922	300	317.8	85	0	1.0	2736	2838.9	45.0
20090922	3600	3376.0	48	0	1.0	2222	2280.8	55.8
20090923	120	244.0	50	0	0	1589	1659.4	17.1
20090923	300	592.1	45	0	0	1464	1550.1	22.5
20090923	3600	6567.3	15	0	0	1388	1409.9	29.5
20090924	120	141.5	142	0	1.1	2670	2822.5	30.6
20090924	300	309.7	85	0	1.0	2415	2578.8	36.5
20090924	3600	3662.7	30	0	1.0	2037	2077.3	48.9
20090925	120	150.4	60	0	0	1935	2051.8	41.4
20090925	300	367.2	38	0	0	1910	1974.0	43.6
20090925	3600	3999.3	17	0	0	1680	1714.0	51.0
20090926	120	246.0	39	0	0	1422	1481.0	16.9
20090926	300	597.3	53	0	0	1303	1351.9	21.1
20090926	3600	6347.8	31	0	0	1072	1110.1	37.7
20090927	120	212.7	34	0	0	1852	1899.4	38.6
20090927	300	516.7	70	0	0	1672	1770.9	42.8
20090927	3600	5987.0	56	0	0	1303	1396.4	54.9
20090928	120	224.2	49	0	0	1533	1623.8	26.5
20090928	300	544.8	46	0	0	1445	1512.9	31.5
20090928	3600	5660.8	17	0	0	1243	1263.9	42.8
20090929	120	123.8	85	0	0	2043	2221.3	31.8
20090929	300	302.1	48	0	0	2003	2102.3	35.5
20090929	3600	2937.0	73	0	0	1658	1752.3	46.2
20090930	120	103.4	1478	2.8	5.0	4051	7351.9	45.3
20090930	300	250.1	1264	1.3	3.3	3992	5142.8	61.7
20090930	3600	2382.1	223	1.0	0.3	3130	3288.4	75.5
20091001	120	112.1	168	0	0.3	2554	2777.1	36.9
20091001	300	270.0	162	0	0.1	2230	2470.0	43.9
20091001	3600	2537.0	57	0	0	1746	1848.1	58.0

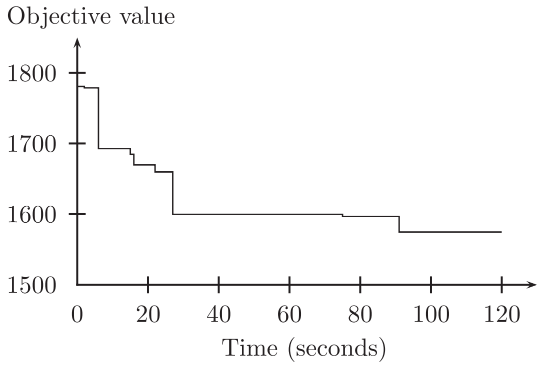


Fig. 3. Development of solution values over time for test case 20090926.

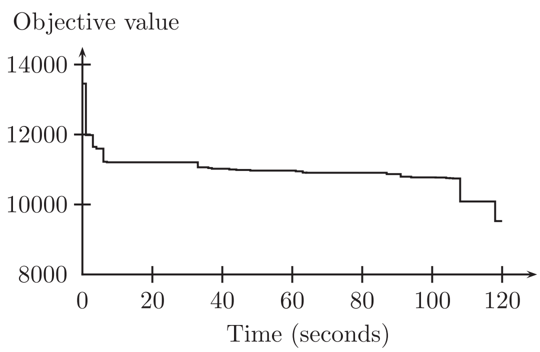


Fig. 4. Development of solution values over time for test case 20090930.

10 times and the average value is reported. The temperature, cooling rate and number of moves are set to the selected values and the excess time allowed on each segment is set to $H = 30$ (minutes).

The average number of iterations and the standard deviation are reported in columns three and four. In columns five and six respectively, the average number of unassigned prebooked PRMs and unassigned immediate PRMs are reported. The best solutions of all the runs are reported in column seven and column eight contains the average solution. The gap between the average solution and the initial solution from Table 2 is reported in column nine. The results in Table 3 show that after running simulated annealing for 2 minutes the initial greedy heuristic solution is significantly improved. Increasing the solution time improves the solution quality in all cases and in some cases this improvement is significant. However, the improvements achieved from the initial solution in the first 5 minutes is in all cases greater than the improvements achieved in the following 55 minutes. The improvement achieved in the first 2 minutes from the initial solution is (in all cases except run 20090926) greater than the improvement achieved in the following 58 minutes. This indicates that the algorithm is good at finding improvements early in the algorithm and therefore works well with the requirement that a solution must be found within minutes.

Figs. 3 and 4 show the improvement of solution values for a 2 minutes run. It is seen that the algorithm is able to steadily improve the solution value.

The graphs in Figs. 5 and 6 show the development in solution values for a single 1 hour run of respectively test instances 20090926 and 20090930. The figures show that after running simulated annealing for 15 minutes the improvements to the solutions become seldom and insignificant. This means that the significant reductions occur early in the simulated annealing and good solutions can be found after 10 minutes. In our case the solution time required is quite limited. In cases with not as tight running time

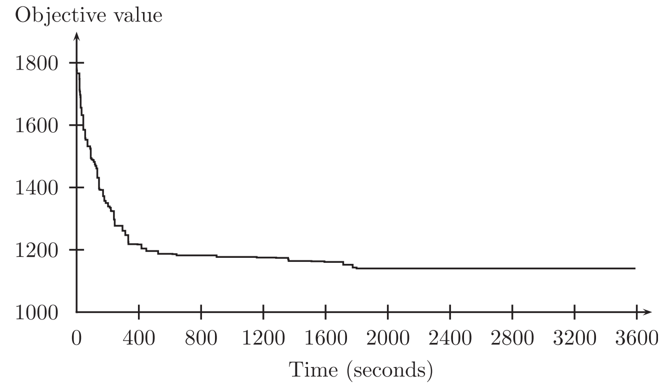


Fig. 5. Development of solution values over time for test case 20090926.

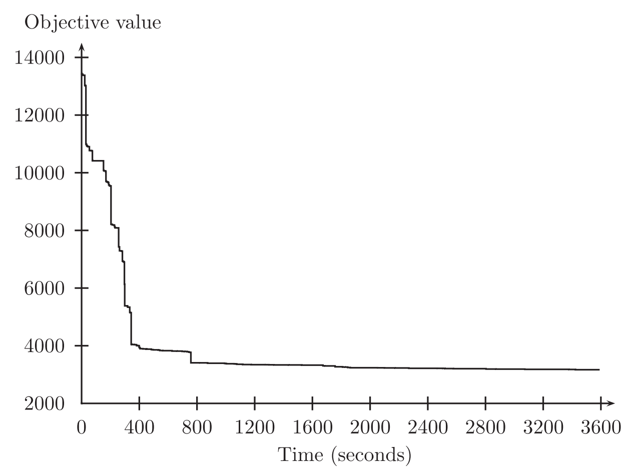


Fig. 6. Development of solution values over time for test case 20090930.

requirements it would be relevant to test if multiple random restarts of the algorithm with a new ordering of the lists would allow for searches in other areas of the search space and possibly improve the solutions. In the current algorithm the neighborhood used is very large and therefore it is possible for the algorithm to investigate a larger area of the search space.

8. Conclusion

We have presented a model for the airport PRM transport problem and developed a heuristic with promising results, even when restricted to a running time of 2 min as required by the service company. Moreover, the number of rejected PRMs is very low also in the initial greedy insertion heuristic solutions.

Although the problem has been defined in cooperation with a specific handling company, we believe that the model is sufficiently general to cover most airports in the world. Moreover the model and solution method can easily be adapted to other multi-modal problems with many synchronization constraints such as patient transport within a hospital and multi-modal transports involving liner vessels and trains.

Due to the fast solution time of 2 minutes, the developed algorithm can easily be run several times during a day in an environment where rescheduling is needed due to unexpected changes in the schedule. However, the formulation of the problem is static and to truly work in a dynamic environment, a different problem formulation will be needed.

The developed heuristic works well with the short solution time constraint, which can be seen by the fact that the big improvements are obtained within the first few minutes. The tests show that increasing the solution time slightly could in some cases give significant improvements. Such improvements could also be achieved by increasing computational power or algorithmic improvements such as parallelization of the program.

The next step in the academic area would be to test different solution methods on the problem to compare the solution quality and to reduce the computation time for the greedy insertion heuristic. One such paradigm could be to apply the combined MIP and ALNS heuristic presented in [13] which has shown good results for tightly constrained problems where even finding a feasible solution is an issue. In the application area the next step would be to incorporate this method at the users to see if the developed plans can improve their daily service.

We have assumed that the personnel and the location of the personnel is fixed, but the short solution times make it possible to use local search to test whether relocation of some personnel may lead to a higher service level. Since the number of passengers arriving at the airport is increasing the heuristic can also be used to investigate when an increase in personnel is needed to deliver acceptable service to the increasing volumes of PRMs arriving.

Acknowledgments

The authors thank Torben Barth, Berit Løfstedt, Christian Edinger Munk Plum, Charlotte Vilhelmsen and Mette Gamst for valuable comments. The authors also wish to thank Jacob Colding for presenting and clarifying the problem to us.

Appendix A. Appendix: Tuning tests

Table 4 describes the three test cases used for tuning. Using a time limit of 2 minutes, Table 5 reports results for tuning the

Table 4
The test cases used for tuning, with the results from the greedy insertion heuristic. NAP stands for the number of not assigned prebooked PRMs and NAI stands for the number of not assigned immediate PRMs. The column PRMs is the total number of PRMs provided in the instance of which the number given in 'deleted' is deleted.

Case	PRMs	Deleted	Init. NAP	Init. NAI	Init. sol.
20090920	374	21	0	4	3566
20091001	519	45	1	1	4401
20090926	401	27	0	0	1782

Table 5
Tuning cooling rate and number of moves for solution time of 120 seconds.

Testcase	Coolrate	Moves	Av. ite	Std D	Av. NAP	Av. NAI	Best sol.	Avg. sol.	Impr (%)
20090920	0.5	4	326.5	179	0	2.3	2491	2655.2	25.5
	0.8	4	325.9	187	0	2.5	2457	2732.2	23.4
	0.9	4	326.7	242	0	2.4	2426	2720.2	23.7
	0.95	4	324.5	201	0	2.5	2511	2753.5	22.8
	0.99	4	322.1	239	0	2.5	2605	2912.2	18.3
	0.5	12	315.8	57	0	2.0	2396	2495.8	30.0
	0.8	12	316.7	58	0	2.0	2396	2496.3	30.0
	0.9	12	315.5	34	0	2.0	2461	2510.4	29.6
	0.95	12	310.1	55	0	2.0	2431	2500.0	29.9
	0.99	12	306.3	52	0	2.0	2541	2609.3	26.8
	0.5	20	306.1	79	0	2.0	2351	2521.5	29.3
	0.8	20	309.9	47	0	2.0	2418	2505.6	29.7
	0.9	20	308.8	33	0	2.0	2450	2515.2	29.5
	0.95	20	301.2	53	0	2.0	2437	2495.0	30.0
	0.99	20	298.7	46	0	2.0	2464	2564.3	28.1
20091001	0.5	4	125.3	198	0	0.4	2461	2691.7	38.8
	0.8	4	125.4	130	0	0.8	2521	2796.7	36.5
	0.9	4	123.2	120	0	0.5	2543	2794.8	36.5
	0.95	4	120.0	187	0	0.2	2319	2737.5	37.8
	0.99	4	117.7	205	0	0.7	2869	3149.8	28.4
	0.5	12	120.8	111	0	0.3	2736	2838.9	35.5
	0.8	12	115.1	161	0	0.1	2510	2742.0	37.7
	0.9	12	114.4	190	0	0.3	2472	2723.3	38.1
	0.95	12	107.7	126	0	0.3	2598	2749.2	37.5
	0.99	12	104.1	130	0	0.2	2677	2893.6	34.3
	0.5	20	113.2	131	0	0.1	2789	2891.6	34.3
	0.8	20	110.4	141	0	0.2	2635	2834.0	35.6
	0.9	20	106.1	184	0	0.3	2459	2820.9	35.9
	0.95	20	103.3	136	0	0.4	2644	2831.5	35.7
	0.99	20	100.0	129	0	0.6	2842	3048.1	28.1
20090926	0.5	4	255.1	40	0	0	1378	1442.6	19.0
	0.8	4	256.5	50	0	0	1368	1427.9	19.9
	0.9	4	254.8	41	0	0	1352	1428.5	19.8
	0.95	4	254.8	48	0	0	1395	1480.8	16.9
	0.99	4	250.1	80	0	0	1437	1563.0	12.3
	0.5	12	248.0	50	0	0	1425	1482.0	16.8
	0.8	12	245.1	42	0	0	1420	1484.3	16.7
	0.9	12	245.2	52	0	0	1322	1431.5	19.7
	0.95	12	241.6	70	0	0	1380	1491.7	16.3
	0.99	12	236.1	78	0	0	1399	1508.8	15.3
	0.5	20	243.0	50	0	0	1400	1508.4	15.4
	0.8	20	240.2	57	0	0	1424	1496.7	16.0
	0.9	20	237.1	73	0	0	1370	1490.5	16.4
	0.95	20	234.5	55	0	0	1347	1471.1	17.4
	0.99	20	228.9	55	0	0	1422	1520.9	14.7

cooling rate and the number of moves per iteration, while Table 6 reports results for tuning the initial temperature parameter t. Using a time limit of 1 hour, Table 7 reports results for tuning the coolrate and the number of moves per iteration, while Table 8 reports results for tuning the initial temperature parameter t.

Table 6
Tuning initial temperature parameter t given required solution time of 120 seconds, 12 moves and cooling rate of 0.9.

Case	Temp. par. t (%)	Av. it.	Std D	Av. NAP	Av. NAI	Best sol.	Av. best sol.	Impr (%)
20090920	1	317.0	53	0	2.0	2379	2490.3	30.2
	5	315.5	34	0	2.0	2461	2510.4	29.6
	10	313.4	146	0	2.2	2454	2585.3	27.5
	15	312.6	36	0	2.0	2462	2542.9	28.7
20091001	1	107.0	175	0	0.4	2470	2812.2	36.1
	5	114.4	190	0	0.3	2472	2723.3	38.1
	10	109.2	130	0	0.2	2538	2717.9	38.2
	15	108.8	234	0	0.3	2428	2774.5	37.0
20090926	1	248.3	64	0	0	1387	1470.3	17.5
	5	245.2	52	0	0	1322	1431.5	19.7
	10	241.2	54	0	0	1370	1473.8	17.3
	15	242.0	44	0	0	1384	1486.4	16.6

Table 7
Tuning cooling rate and number of moves for solution time of 1 hour.

Testcase	Coolrate	Moves	Av. ite.	Std. D	Av. NAP	Av. NAI	Best sol.	Avg. sol.	Impr (%)
20090920	0.5	4	7706.9	36	0	2.0	2170	2240.7	37.2
	0.8	4	8129.1	29	0	2.0	2146	2200.0	38.3
	0.9	4	8985.3	44	0	2.0	2089	2164.4	39.3
	0.95	4	8968.7	27	0	2.0	2146	2194.1	38.5
	0.99	4	8957.7	27	0	2.0	2121	2177.2	38.9
	0.5	12	7854.3	19	0	2.0	2197	2242.2	37.1
	0.8	12	8793.4	30	0	2.0	2166	2207.8	38.1
	0.9	12	8762.2	24	0	2.0	2165	2202.5	38.2
	0.95	12	7270.6	26	0	2.0	2165	2200.8	38.3
	0.99	12	7309.2	28	0	2.0	2160	2204.4	38.2
	0.5	20	8863.0	29	0	2.0	2212	2261.2	36.6
	0.8	20	6721.1	33	0	2.0	2250	2297.9	35.6
	0.9	20	5958.8	28	0	2.0	2221	2261.5	36.6
	0.95	20	5939.4	38	0	2.0	2221	2280.7	36.0
	0.99	20	8281.6	35	0	2.0	2200	2275.3	36.2
20091001	0.5	4	3284.7	61	0	0	1723	1848.4	58.0
	0.8	4	3218.4	97	0	0	1689	1844.8	58.1
	0.9	4	3234.7	57	0	0	1723	1823.7	58.6
	0.95	4	3150.8	104	0	0.1	1653	1849.6	58.0
	0.99	4	3010.5	64	0	0	1685	1782.5	59.5
	0.5	12	3166.4	170	0	0	1794	2083.4	52.7
	0.8	12	3028.0	139	0	0.1	1800	2037.6	53.7
	0.9	12	2219.3	77	0	0	1972	2118.8	51.9
	0.95	12	2228.1	58	0	0	1949	2051.5	53.4
	0.99	12	2434.8	92	0	0	1943	2043.6	53.6
	0.5	20	3043.4	93	0	0	2035	2211.1	49.8
	0.8	20	2582.9	102	0	0	2055	2223.2	49.5
	0.9	20	1909.7	138	0	0	2115	2284.2	48.0
	0.95	20	2089.4	66	0	0	2136	2219.3	49.6
	0.99	20	1778.5	66	0	0	2068	2158.8	50.9
20090926	0.5	4	5982.2	22	0	0	1093	1117.8	37.3
	0.8	4	3916.9	16	0	0	1050	1083.8	39.2
	0.9	4	4712.9	35	0	0	1035	1092.8	38.7
	0.95	4	6418.6	17	0	0	1043	1066.9	40.1
	0.99	4	4122.6	38	0	0	1032	1077.9	39.5
	0.5	12	6815.5	35	0	0	1094	1139.4	36.1
	0.8	12	4903.2	32	0	0	1086	1147.0	35.6
	0.9	12	4479.0	53	0	0	1078	1154.6	35.2
	0.95	12	5900.2	25	0	0	1095	1138.9	36.1
	0.99	12	6127.5	39	0	0	1128	1162.3	34.8
	0.5	20	5755.6	28	0	0	1171	1203.7	32.5
	0.8	20	5968.6	26	0	0	1177	1221.8	31.4
	0.9	20	6252.7	44	0	0	1145	1222.3	31.4
	0.95	20	6511.7	44	0	0	1167	1211.5	32.0
	0.99	20	5926.8	39	0	0	1149	1187.5	33.4

Table 8
Tuning initial temperature given required solution time of 1 hour, four moves and cooling rate of 0.99.

Case	Temp. par. *t*	Av. it.	Std. D	Av. NAP	Av. NAI	Best sol.	Av. best sol.	Impr (%)
20090920	1	8821.1	30	0	2.0	2123	2188.4	38.6
	5	8769.1	23	0	2.0	2166	2190.7	38.6
	10	8694.1	32	0	2.0	2113	2181.9	38.8
	15	8287.5	40	0	2.0	2148	2206.5	38.1
20091001	1	3097.5	72	0	0	1632	1759.6	60.0
	5	2916.4	128	0	0	1578	1790.0	59.3
	10	2470.2	87	0	0	1697	1828.2	58.5
	15	2758.6	71	0	0	1750	1856.9	57.8
20090926	1	6256.2	23	0	0	1048	1082.5	39.3
	5	6237.9	21	0	0	1045	1077.4	39.5
	10	6378.6	27	0	0	1069	1108.6	37.8
	15	6617.1	33	0	0	1037	1089.4	38.9

References

[1] J.W. Baugh, G.K.R. Kakivaya, J.R. Stone, Intractability of the dial-a-ride problem and a multiobjective solution using simulated annealing, Engineering Optimization 30 (1998) 91–123.
[2] P. Chen, Y. Guo, A. Lim, B. Rodrigues, Multiple crossdocks with inventory and time windows, Computers & Operations Research 33 (2006) 43–63.
[3] J. Cordeau, G. Laporte, The dial-a-ride problem: models and algorithms, Annals of Operations Research 153 (2007) 29–46.
[4] J.-F. Cordeau, G. Laporte, A tabu search heuristic for the static multi-vehicle dial-a-ride problem, Transportation Research Part B: Methodological 37 (6) (2007) 579–594.
[5] J.-F. Cordeau, G. Laporte, The dial-a-ride problem: models and algorithms, Annals of Operations Research 153 (2007) 29–46.
[6] C.E. Cortes, M. Matamala, C. Contardo, The pickup and delivery problem with transfers: formulation and a branch-and-cut solution method, European Journal of Operational Research 200 (3) (2010) 711–724.
[7] T.G. Crainic, N. Ricciardi, G. Storchi, Models for evaluating and planning city logistics systems, Transportation Science 43 (2009) 432–454.
[8] M. Diana, M.M. Dessouky, A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows, Transportation Research Part B 38 (2004) 539–557.
[9] A. Dohn, M.S. Rasmussen, J. Larsen, The vehicle routing problem with time windows and temporal dependencies, Networks 58 (2011) 273–289.
[10] A. Flower, Gatwick Managing Directors Report. <http://www2.westsussex.gov.uk/ds/cttee/gat/gat230709i7.pdf>, 2009. Publisher: <www.gatwickairport.com>.
[11] I. Ioachim, J. Desrosiers, F. Soumis, N. Belanger, Fleet assignment and routing with schedule synchronization constraints, European Journal of Operational Research 119 (1999) 75–90.
[12] J. Jaw, A. Odoni, N. Psaraftis, H.M. Nigel, A heuristic algorithm for the multi-vehicle advance request deal-a-ride problem with time windows, Transportation Research part B 20 (1986) 243–257.
[13] L.F. Muller, S. Spoorendonk, D. Pisinger, A hybrid adaptive large neighborhood search heuristic for lot-sizing with setup times, European Journal of Oprational Research 218 (2012) 614–623.
[14] S. Parragh, K.F. Doerner, R.F. Hartl, A variable neighborhood search for the dial-a-ride problem, Computers & Operations Research 37 (2010) 1129–1138.
[15] D. Pisinger, S. Ropke, A general heuristic for vehicle routing problems, Computers & Operations Research 34 (2007) 2403–2435.
[16] L. Reinhardt, D. Pisinger, A branch and cut algorithm for the container shipping network design problem, Flexible Services and Manufacturing Journal 24 (2012) 349–374.
[17] S. Ropke, J.-F. Cordeau, G. Laporte, Models and branch-and-cut algorithms for pickup and delivery problems with time windows, Networks 49 (2007) 258–272.
[18] M. Wen, J. Larsen, J.-F. Cordeau, G. Laporte, Vehicle routing with cross-docking, Journal of the Operational Research Society 60 (2009) 1708–1718.
[19] Z. Xiang, C. Chu, H. Chen, A fast heuristic for solving a large-scale dial-a-ride problem under complex constraints, European Journal of Operational Research 174 (2006) 1117–1139.