Synchronizing transportation of people with reduced mobility through airport terminals

René van Twist · Marjan van den Akker · Han Hoogeveen

Keywords Passengers with Reduced Mobility · airport planning · synchronization · simulation · local search · simulated annealing

1 Introduction

Moving through the airport is simple for most passengers, but when you are restricted in movement, visually impaired or have some other kind of disablement, this is not. Airports offer a service to assist Passengers with Reduced Mobility (PRMs). There must always be an employee guiding the PRM unless the PRM stays in a special, supervised lounge. Since an employee is only allowed to work in his own terminal, the PRM must be handed over when moving to another terminal; since the terminals are not connected, PRMs are required to take a special bus to go to another terminal.

In this paper we consider the problem of scheduling the PRMs and the employees who are guiding them such that as many as possible PRMs are served, and such that the served PRMs get the best possible service. This problem was first considered by Reinhardt et al. (2013) [4], who use a greedy heuristic based on Simulated Annealing. They very kindly made available the anonymized data for a major European airport with 300 to 500 PRMs per day.

The main objective is to give the best service possible to the PRMs, and avoid as much as possible that PRMs cannot be scheduled and need to take a later flight.

Moreover, we want to improve service further by providing smooth connections to the PRMs. This is achieved by minimizing unnecessary travel time

Department for Information and Computing Sciences
Utrecht University
P.O. Box 80089, 3508 TB Utrecht, The Netherlands
E-mail: rene@vantwist.net, J.M.vandenAkker@uu.nl, J.A.Hoogeveen@uu.nl

Proceedings of the 11th International Confenference on Practice and Theory of Automated Timetabling (PATAT-2016) – Udine, Italy, August 23–26, 2016

for PRMs, which for instance occurs if an employee assisting a PRM takes a detour or is waiting for someone else to pick up the PRM.

Our addition to the work by Reinhardt et al. [4] is threefold:

1. We decrease unnecessary waiting time for the PRMs.
2. We make a robust schedule.
3. Our approach can be applied in a dynamic environment.

1.1 Journeys of the PRMs

We consider three categories of PRMs: Arrival, Departure, and Transfer. Figure 1 depicts a schematic overview of a journey in the category Transfer. If we see the route as a path in a graph, then each edge must be covered by one employee. We call an edge that can be served by an employee a *segment*.

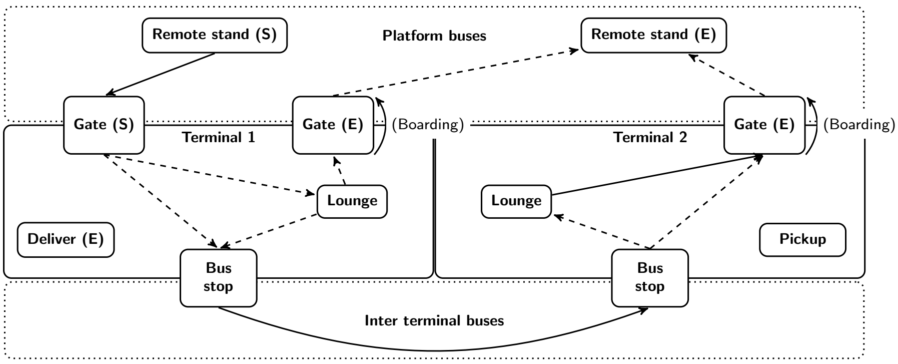

Fig. 1 Examples of all possible transfer journeys. Dashed edges are either optional segments or choices between 2 routes.

2 Our approach: decomposition

We use a decomposition approach. The first step in the decomposition model is to set the start times of the segments. The next step is to use a matching algorithm to find a feasible assignment of employees to the segments. We then alter the start times of the segments using simulated annealing, and update the assignment, etc.

If a PRM must be handed over, then we prefer to start the next segment as soon as possible, unless the PRM has arrived in a lounge. Therefore, if we know the start time of a segment, then we know when the next segment must start to avoid a penalty. Consequently, we split the journey in segment groups (or sub-journeys), where the segments within a segment group must be served contiguously. If we stick to using these segment groups, we fully eliminate the unnecessary waiting times.

As mentioned, the local search algorithm tries to find good starting times for the segments. Observe that synchronizing journeys can allow PRMs to share an employee or bus and are hence good for freeing up employees for other tasks. Therefore, we try to merge segment groups. If such a merger is possible, then we try to synchronize as many segments of the corresponding PRMs as possible. The mutations in our local search algorithm are: Plan PRM, Decline PRM, Move Segment Group, Merge Segment Group and Split Merged Group.

To make the decomposition approach run smoothly, it is essential that after a change of the start times by the local search the assignment is updated immediately. Therefore, we run the matching algorithm after each move of the local search algorithm.

In this matching problem we only have to assign which segment is served by which employee. Since it is fixed which PRMs are to be served and all unnecessary idle time has been eliminated, the only goal is to optimize robustness. Robustness is measured by a specific function of the slack times in the schedule.

If all employees work the same shift, then this problem can be solved as a Single Depot Vehicle Scheduling Problem [3], which can be solved in polynomial time. On the other hand, if the shifts that the employees work are not all equal, then the problem boils down to a multi depot vehicle scheduling problem [1], which is NP-Hard.

We first solved the matching problem (without shifts) to optimality through the Hungarian method in each iteration. This took very much time. Therefore, we considered a specific heuristic, which we call *Reschedule overlapping segments*.

Our computational results indicated that like [4], our algorithm provides solutions with very few declined PRM's. However, for deterministic data(without disturbances), our algorithm finds solutions without any unnecessary waiting time. The *Reschedule overlapping segments* heuristic significantly reduced the computation time. All our solution were found within 2 minutes.

3 Simulation

We have developed an algorithm that tries to find robust schedules. The next question is if the schedule will perform well in the real world where different disturbances occur. Moreover, does the inclusion of robustness pay off with respect to the number of times that we have to reschedule? To answer these questions we made a simulation model. We included three different types of disturbances: the arrival of unbooked PRMs asking for assistance, flight delays and trips taking longer than expected.

We have simulated both the solution with robustness and the solution neglecting robustness. Our results indicate that considering robustness is especially important to reduce the waiting time.

4 Conclusion

Computational experiments have shown that the algorithm of Reinhardt et al [4] could find good solutions in 2 minutes and high quality ones in 10 minutes in which no PRMs get rejected, but there is still unnecessary waiting time. Computational results show that our decomposition approach can solve all static problems to optimality. More importantly, the simulation results show that it is capable of solving the problems in a dynamic environment, even if we add 100 PRMs to the original instance. Due to our inclusion of robustness in the schedule, we can reduce the unnecessary waiting times significantly in a real-time situation.

References

1. G. Desaulniers, J. Lavigne, and F. Soumis. (1998) Multi-depot vehicle scheduling problems with time windows and waiting costs. *European Journal of Operational Research 111*, pp. 479–494.
2. G. Diepen, B.F.I. Pieters, J.M. van den Akker, J.A. Hoogeveen. (2013) Robust planning of airport platform buses. *Computers & Operations Research 40*, pp. 747–757.
3. Richard Freling, Albert P.M. Wagelmans and José M. Pinto Paixão (2001) Models and algorithms for single-depot vehicle scheduling. *Transportation Science 35*, pp. 165–180.
4. L.B. Reinhardt, T. Clausen, and D. Pisinger (2013) Synchronized dial-a-ride transportation of disabled passengers at airports, *European Journal of Operational Research 225*, pp. 106–117.