Eighth USA/Europe Air Traffic Management Research and Development Seminar (ATM2009)

Airport Service Vehicle Scheduling

Kenneth Kuhn
Aviation Systems Division
NASA
Moffett Field, USA
kenneth.d.kuhn@nasa.gov

Steffen Loth
Institute of Flight Guidance
DLR
Braunschweig, DE
steffen.loth@dlr.de

***Abstract*–Airport service vehicles, such as luggage trailers and passenger buses, service an aircraft after the aircraft arrives and before it departs. The timely arrivals of these vehicles help ensure efficient use of airport resources. This research investigates algorithms for scheduling airport service vehicles. A mixed integer linear program is proposed, minimizing service provider fuel costs and air carrier delays. The formulation of the integer programming problem is modified to aid solution search strategies. A genetic algorithm heuristic borrowed from aircraft arrival scheduling is introduced for finding approximate solutions relatively quickly, in addition to an exact solution method making use of branch and bound techniques specially designed for this problem. The various algorithms are tested using simulations of service provider dispatch problems at Hamburg and Dallas-Fort Worth Airports. Results show that plan based service vehicle scheduling reduces both delay and fuel costs over passive strategies, often 20% or more. The genetic algorithm based heuristic also reduced delay and fuel costs while incurring computational burdens significantly below those of the optimal search strategy.**

Keywords-airport surface; scheduling; service vehicles; decision support tools; vehicle routing;

I. Introduction

Service vehicles approach aircraft that land at an airport and are parked at a gate or stand, as shown in Fig. 1. The quantities of vehicles, fuel, and employees, that an airport service provider utilizes in meeting demands for service from arriving and departing aircraft are determined by the demands and by how the demands are met. Determining how to meet a set of demands is a scheduling problem and will be addressed in this paper.

Airport service vehicle schedules determine how much delay aircraft absorb waiting for service on airport surfaces. Gate and stand positions are in short supply at many airports and delay incurred here by one aircraft can delay subsequent aircraft that might have used the position. An analysis of operational data in [1] revealed that "for most airports there is a dominance of the delays due to gate congestion (gate occupied) over the other delay categories such as ramp and field congestion." Different processes at airports are related (by reliance on ground support vehicles for example) and delays can propagate from one gate or stand to others. The reader is again referred to [1], which contains a subsection entitled "Interdependence between gates." Indeed the entire air transportation system is linked and an aircraft delayed at one airport may disrupt schedules at other airports it arrives at later in that day. To limit delays to the traveling public, it is essential that service be provided in an efficient manner on airport surfaces.

It is for the reasons noted above that the Deutsches Zentrum für Luft- und Raumfahrt (DLR), with partners from industry, national service providers, research institutions, and universities, initiated the Car Management on Aprons (CARMA) project. CARMA will examine the technical feasibility of implementing a cost-effective system at Hamburg airport to track and communicate with airport service vehicles, manage vehicles from stakeholder control centers, and support efficient turnaround processes for aircraft.

The next section provides background on airport service vehicle scheduling. The following section introduces a general framework and a few specific methods for solving dynamic and static airport service vehicle scheduling problems. Computational studies simulating service vehicle scheduling problems at Hamburg and Dallas-Fort Worth Airports follow.

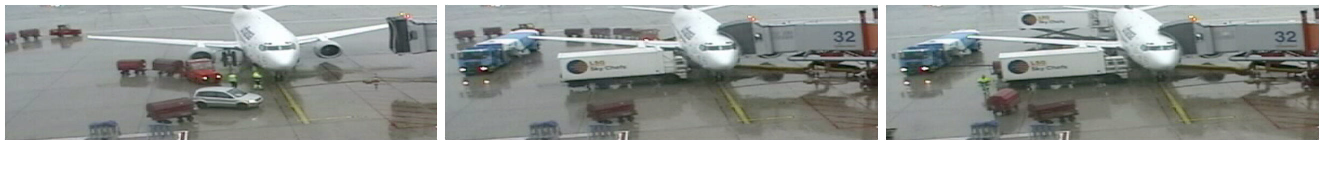

Figure 1. Ground support vehicles in operation. (Photos courtesy of Yves Günther, DLR.)

II. Background

There is a relatively small body of research relating to the operations of airport service vehicles, in part due to the ownership of these vehicles. In Europe, airport management organizations and private companies predominate, while in the United States, airlines manage much of their own services. Government research organizations studying aviation systems have focused mainly on operations managed by public agencies.

Research on service vehicle operations to date has been framed around the turnaround processes of aircraft. Reference [1] includes operational diagram of the turnaround process. The authors broke down the states of aircraft during the turnaround process into states that are and are not observable to air traffic controllers. The lack of full transparency prevents immediate recognition of delay creation. Reference [2] speaks, in general terms, of how a gate management system linked to ground support operational data could provide feasible pushback times for aircraft, necessary for departure planning. Reference [3] studies the uncertainty surrounding when turnaround processes finish. The authors find that even with knowledge of the status of several service tasks it is difficult to predict how much delay aircraft will incur. "As an expected pushback time approaches, the absence of expected status changes (e.g., the end, or even the start, of the boarding process) indicates that the turn is likely to be late, but the exact degree of lateness becomes very difficult to predict" [3]. The data the authors analyzed came from the ALLEGRO project, an effort led by Lufthansa to study and improve the efficiency of ground operations [4]. ALLEGRO is operational at Hamburg Airport and DLR researchers working on the CARMA project have established a link with Lufthansa. In a presentation on the ALLEGRO project, Treude detailed the turnaround process [4]. In this description, Treude defined the different tasks that different service providers have to accomplish and how progress can be measured.

Treude and the other authors referenced above, did not model decision-making by service providers. Doing so may provide insight on how airport service vehicle operations may be made more efficient, reducing delay aircraft absorb and costs service providers accrue. In particular, in this paper the focus is on scheduling for service vehicles.

There is a sizable amount of relevant research relating to vehicle scheduling. Reference [5] proposed a heuristic solution methodology for vehicle scheduling that involved ordering tasks/customers by earliest possible service start time and then inserting these tasks/customers one by one into vehicle paths in a greedy fashion. In the same year, [6] introduced an approach that first clustered close customers together and then chose routes through the clusters of customers. More recently, [7] introduced a tabu search heuristic for one formulation of a multi-vehicle routing problem. Reference [8] also introduced an exact solution strategy based on using a branch-and-cut methodology with new valid inequalities for relaxed problems.

The research described above solve 'static' formulations where all parameters are known accurately and provided ahead of time. In 'online' or 'dynamic' formulations customer service requests appear and must be added to vehicle routes in real-time. A few works including [9] and [10] have looked at the single-vehicle version of this problem. Significantly fewer works have investigated the multi-vehicle version of this problem, notably [11]. The two most common strategies proposed for the online problem are 'plan at home' and 'integral/break.' The first strategy involves each vehicle serving a set of customers and then returning to a central depot, at which point the vehicle is assigned another set of customers to serve. The second strategy involves vehicles being given a route to follow until a specified amount of time has elapsed, at which point a new route will be created. Reference [12] has investigated multi-vehicle problems that lie between the static and the dynamic cases. In this work customers have 'disclosure dates' when their location and desire for service become known to the vehicle dispatcher.

Certain characteristics differentiate the problem of airport service vehicle scheduling from previously studied vehicle routing problems. Service vehicles typically do not return to any central depot, or other common location, until the end of the working day. It is often impossible to set aside parking space anywhere near the center of busy airport surfaces. Service providers must plan routes that involve spending varying amounts of time at customers' locations (parked aircraft) and cannot serve multiple customers at once. The problem investigated in this paper involves managing a fleet of service vehicles with complete knowledge of which aircraft have to be serviced, but an incomplete knowledge of when and where these aircraft will be requesting service or precisely how long it will take to service aircraft. Available data are updated over time, so that a dispatcher likely knows quite precisely demands for service for the next fifteen minutes but not so precisely those of the next five hours. It is believed that the characteristics of the airport service vehicle scheduling problem listed above may be common to a variety of vehicle routing and scheduling problems.

III. Framework

A framework for airport surface vehicle scheduling is introduced here. At the heart of this model is a scheduling algorithm. The algorithm receives parameter values from an operational model, and is alerted when vehicle operations deviate from the algorithm's planned schedule. Examples of parameters that may be used include times at which aircraft will require service, lengths of time aircraft will take to service, numbers of different types of service vehicles required, and distances between locations. The operational model takes as input static and dynamic sets of data. Static data of interest may include a map of the airport surface with gate and stand locations, as well as descriptions of aircraft and service vehicle types. Dynamic data of interest includes aircraft flight numbers, aircraft tail numbers, aircraft types, stand or gate assignment, estimated times of arrival at runways, terminal assignment, passenger counts (when available), aircraft status, and estimated in and off block times.

Expert guidance helps define the operational model, as does analysis of service provider data. This data may include timestamps when vehicles accepted tasks, were in position, began tasks, and finished tasks. All of the data mentioned above are currently available to the service provider at the Hamburg Airport working with the CARMA project. These data are likely also available to air traffic controllers and airlines across Europe and the U.S., although it is not clear if it is always shared with service providers. This work provides some indication of what could be accomplished if such information were used to support service vehicle scheduling. A chart of the framework introduced here is shown in Fig. 2.

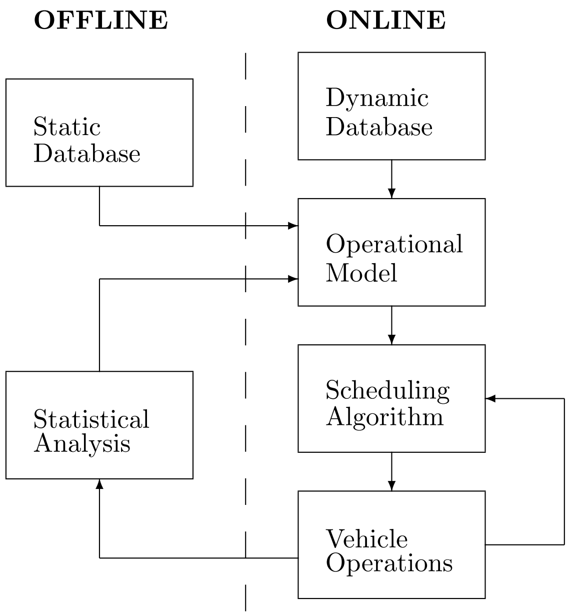

Figure 2. Flow chart of data for scheduling.

A. *Dynamic Scheduling*

The focus in this paper is placed primarily on scheduling algorithms used in the above context. To simplify analysis, a problem involving the management of a fleet of homogeneous vehicles is considered. A number of aircraft are requesting service at various times and locations over the course of a day. The problem is to plan a schedule for a fleet of vehicles such that each aircraft is serviced by a vehicle at, or after, the time the aircraft requests service. Parameters used by scheduling algorithms are defined as follows.

Let I represent the set of all aircraft that will require service and X represent the set of all service vehicles. The service vehicles travel with an assumed and fixed velocity V. Each aircraft $i \in I$ has a time at which it is expected to request service, T_i, and an amount of required service time, S_i. For any i, $j \in I$, $D_{i,j}$ is the distance a service vehicle will have to travel going directly from finishing servicing aircraft i to getting into position to service aircraft j. Note that for buses carrying passengers between terminals and stands $D_{i,j}$ does not necessarily equal $D_{j,i}$. Also, for any $x \in X$, $i \in I$ let $D_{x,i}$ be the distance vehicle x will have to travel to go from its current location to a position where it is ready to service aircraft i. Let F_x represent the time that vehicle x becomes idle, following the schedule currently set. Finally, let C represent the current time and Π the set of aircraft that have not yet been assigned to a service vehicle.

It will be assumed that the parameters T, S and D will be updated based on the dynamic data link previously discussed, while F, C, and Π are updated by proposed scheduling algorithms. To reflect uncertainty in aircraft on or off block times, durations of tasks, or vehicle transit times, it may be necessary to add buffers to T, S, and/or D variables respectively.

Initial discussions with one service vehicle dispatcher at Hamburg Airport provided insight into current operations. The dispatcher monitors regularly updated airport data regarding arriving and departing aircraft. When an aircraft is approximately ten minutes away from requesting service, the dispatcher asks the vehicle that has been idle the longest to go immediately to a position where it can provide service to the aircraft. If no vehicles are idle, the dispatcher waits until the first vehicle becomes idle and then assigns this vehicle this task. (The dispatcher does not have easy access to data regarding the locations of aircraft and service vehicles.) Assuming the dispatcher examines data once per minute, current operations may be modeled as follows.

Algorithm 1: Current operations

```
1.   $C \leftarrow 0$
2.   $F_x \leftarrow 0$ for all $x \in X$
3.   $\Pi \leftarrow I$
4.   REPEAT
5.       UPDATE $T$, $S$, and $D$
6.       For all $i \in \Pi$
7.           If $T_i \leq C + 10$ and min $F_x \leq C$
8.               ASSIGN vehicle $x' = \arg\min F_x$
                     to service aircraft $i$
9.               $F_{x'} \leftarrow C + \frac{D_{x',i}}{V} + S_i$
10.              $\Pi \leftarrow \Pi \backslash \{i\}$
11.      WAIT one minute
12.      $C \leftarrow C + 1$
13.  UNTIL $\Pi = \emptyset$
```

Knowledge of the locations of aircraft and service vehicles is useful in planning efficient vehicle schedules and in reducing delay. A new algorithm could take advantage of this extra information, modifying Algorithm 1 by assigning tasks to service vehicles that are closest geographically rather than those that have been idle the longest. This algorithm is presented below as Algorithm 2. It is labeled "Greedy approach" as the algorithm is always acting greedily to minimize delay and distance traveled, picking the shortest immediately available distance to travel to service an aircraft.

Algorithm 2: Greedy approach

```
1.   $C \leftarrow 0$
2.   $F_x \leftarrow 0$ for all $x \in X$
3.   $\Pi \leftarrow I$
4.   REPEAT
5.       UPDATE $T$, $S$, and $D$
6.       For all $i \in \Pi$
7.           If $T_i \leq C + 10$ and min $F_x \leq C$
8.               ASSIGN vehicle $x' = \arg \min_{x:F_x \leq C} D_{x,i}$
                     to service aircraft $i$
9.               $F_{x'} \leftarrow C + \frac{D_{x',i}}{V} + S_i$
10.              $\Pi \leftarrow \Pi \backslash \{i\}$
11.      WAIT one minute
12.      $C \leftarrow C + 1$
13.  UNTIL $\Pi = \emptyset$
```

The greedy approach is a relatively simple heuristic, and will not always produce very good schedules as measured in terms of vehicle travel distances or aircraft service delays. It will produce poor results, for instance, if, ten minutes before an aircraft requires service, all idle vehicles are far away but one vehicle very close to the aircraft will become idle shortly. The algorithms introduced so far are myopic and responsive, considering tasks seriatim and only when required by an imminent aircraft arrival or departure.

A proactive, plan-based approach considering future demands offers benefits. The 'plan at home' strategy employed in certain vehicle routing algorithms appears poorly suited to this problem due to parking problems discussed earlier. A modified integral/break strategy ([10], [11]) is used instead. Extensions to vehicle paths are planned over short periods of time by solving static vehicle routing problems. Here overlapping time windows are allowed, so that planning may be done every 10 minutes over a 60 minute planning horizon. This approach allows a dispatcher to plan ahead without fixing assignments far in advance, and is presented below as Algorithm 3.

Algorithm 3: Moving time window

```
1.   $C \leftarrow 0$
2.   $F_x \leftarrow 0$ for all $x \in X$
3.   $\Pi \leftarrow I$
4.   REPEAT
5.       UPDATE $T$, $S$, and $D$
6.       $\Psi \leftarrow \{i \in \Pi : T_i \leq C + 60\}$
7.       SOLVE static scheduling problem using data
             regarding all $x \in X$ and $i \in \Psi$
8.       ASSIGN tasks for the next 10 minutes,
             using results of step 7
9.       For any vehicle $x$ assigned to service aircraft $i$
             at time $z$:
10.          $F_x \leftarrow z + S_i$
11.          $\Pi \leftarrow \Pi \backslash \{i\}$
12.      WAIT ten minutes
13.      $C \leftarrow C + 10$
14.  UNTIL $\Pi = \emptyset$
```

The length of the planning horizon and the frequency with which static scheduling problems are solved are variables to be set. In cases where uncertainty is relatively small, it is worthwhile to plan over a longer horizon. This appears to be the case at Hamburg Airport, where traffic is largely comprised of short-haul flights and good data are available regarding the status of arriving and departing aircraft. In cases where uncertainty is larger, it becomes important to reschedule more frequently and to avoid planning far into the future based on imperfect estimates of in and off block times.

B. Static Scheduling

This section describes static scheduling for the moving time window algorithm. Scheduling provides sequences of tasks for service vehicles and a schedule of when tasks are to begin. Decision variables common in route planning are used here.

Let $a^x_{i,j}$ equal 1 if vehicle x serves aircraft j immediately after serving aircraft i, and 0 otherwise. Similarly let $a^x_{0,j}$, $a^x_{j,0}$, and $a^x_{0,0}$ be binary variables representing if a vehicle x serves aircraft j before all other aircraft, after all other aircraft, or if the vehicle serves no aircraft, respectively. For notational convenience, let $I^+ = I \cup \{0\}$. Let b_i be the time at which service begins on aircraft i.

A new model parameter λ relates the two goals of providing service to aircraft as quickly as possible and of reducing travel distances for service vehicles. M is used to create a linear formulation and can be set to any large value. The problem can now be formulated as a mixed integer linear program as shown below.

$$\text{Minimize } \lambda(\sum_{i\in I} b_i)+(1-\lambda)(\sum_{x\in X}\sum_{i\in I^+}\sum_{j\in I} D_{i,j}a^x_{i,j})$$

s.t.

$$\sum_{x\in X}\sum_{i\in I^+} a^x_{i,j} = 1 \qquad \forall j\in I \qquad (1)$$

$$\sum_{j\in I^+} a^x_{0,j} = 1 \qquad \forall x\in X \qquad (2)$$

$$\sum_{j\in I^+} a^x_{j,0} = 1 \qquad \forall x\in X \qquad (3)$$

$$\sum_{i\in I^+} a^x_{i,j} = \sum_{k\in I^+} a^x_{j,k} \qquad \forall j\in I,\ x\in X \qquad (4)$$

$$a^x_{i,j} \in \{0,1\} \qquad \forall i,j\in I^+,\ x\in X \qquad (5)$$

$$b_i \geq T_i \qquad \forall i\in I \qquad (6)$$

$$b_j \geq (F_x + \tfrac{D_{x,j}}{V})a^x_{0,j} \qquad \forall j\in I,\ x\in X \qquad (7)$$

$$b_j \geq b_i + S_i + \tfrac{D_{i,j}}{V} - M(1-a^x_{i,j}) \quad \forall i,j\in I,\ x\in X \qquad (8)$$

The objective function minimizes a weighted sum of aircraft service times and distances traveled by service vehicles. Minimizing service times is equivalent to minimizing delay. Given the assumed constant velocity of service vehicles, minimizing distances traveled is equivalent to minimizing the time vehicles spend driving between aircraft. It would be straightforward to add a new set of weights on service times of different aircraft, say for a case involving shuttle operations which place a relatively high value on reduced turnaround times (see [2]). It would likewise be straightforward to limit delay incurred by each aircraft, or bound the numbers of tasks assigned to each vehicle.

Equation (1) ensures all aircraft are serviced. Equations (2) and (3) ensure all service vehicles begin and end their service tours, respectively. Equation (4) is a flow-balance constraint; if a vehicle arrives at an aircraft, it must leave that aircraft next. Equation (5) ensures tasks are either assigned or not assigned. Equation (6) requires that service cannot begin before an aircraft is ready. Equations (7) and (8) ensure service does not begin before vehicles arrive at an aircraft, whether they travel to the aircraft directly from their current location or after servicing another aircraft.

Reference [8] notes that (8) can be rewritten:

$$b_j \geq b_i + S_i + \tfrac{D_{i,j}}{V} - M(1 - \sum_{x\in X} a^x_{i,j}) \qquad \forall i,j\in I \qquad (9)$$

Taking the summation across the set of service vehicles reduces the number of constraints needed, taking advantage of the fact that an optimal solution assigns only one vehicle to visit each aircraft. The same logic can be applied to (7) as well. This formulation has the added advantage of discouraging the selection of fractional variables when the problem is relaxed (removing the constraint that the $a^x_{i,j}$ variables be binary) and solved.

$$b_j \geq \sum_{x\in X}(F_x + \tfrac{D_{x,j}}{V})a^x_{0,j} \qquad \forall j\in I \qquad (10)$$

In any optimal solution, exactly one vehicle will visit each aircraft exactly one time. This suggests summing across service vehicles and potential vehicle paths. Combining (7) and (8) while maintaining a linear problem structure proves problematic since it is not clear *a priori* when vehicles will service different aircraft. Nonetheless it is possible to tighten (7), recognizing that aircraft cannot be serviced before they are ready, as follows.

$$b_j \geq \sum_{x\in X}\Big[(F_x+\tfrac{D_{x,j}}{V})a^x_{0,j} + \sum_{i\in I}(T_i+S_i+\tfrac{D_{i,j}}{V})a^x_{i,j}\Big] \qquad \forall j\in I \qquad (11)$$

The authors believe (11) is new and offers a significant improvement over (7). The number of constraints required has been reduced, and the selection of fractional variables has been discouraged.

There are $|X|*(|I|+1)^2$ binary variables in the problem stated above, so efficient solution search strategies are required. A branch and bound strategy taking advantage of problem structure is next introduced. A genetic algorithm approach based on research in aircraft arrival scheduling follows. These are two approaches for solving the introduced mixed integer linear program.

C. Branch and Bound

A standard approach to solving mixed integer linear programs is branch and bound, and a specialized form is used here. This algorithm begins by using the greedy approach (or the genetic algorithm introduced later) to provide a good upper bound for the optimal objective function, critical in branch and bound. The static scheduling problem is next relaxed and solved. If the solution assigns binary values to all $a^x_{i,j}$ values, then an optimal solution has been found.

Assuming a binary solution was not found, a particular aircraft i associated with at least one non-binary $a^x_{i,j}$ (or $a^x_{j,i}$) term is chosen. Branching is done by creating $|X|$ new problems where the aircraft i is assigned to each service vehicle x in turn. For each of these subproblems, $a^y_{i,j}$ and $a^y_{j,i}$ are set equal to 0 for any $j{\in}I^+$, for any $y{\in}X$, $y{\neq}x$. The new problems are each relaxed and solved. Note that by branching on vehicle assignment, $(|I|{+}1){*}2{*}(|X|{-}1)$ binary variables were set to 0. A more standard approach branching on a particular $a^x_{i,j}$ would have set one binary variable.

The upper bound is compared to the best objective function value given by binary variables, and updated as needed. The problem that was recently branched, problems with objective values greater than the upper bound, and problems that produced binary solutions are said to be pruned and ignored in the future. Problems that did not produce integer solutions but did produce objective function values below the upper bound are said to be live. The live problem with the best objective function value is next studied. This is what is known as 'best first' branch and bound: high quality solutions are obtained, the upper bound updated, and sub-optimal branches pruned relatively quickly.

Branching is done as before, by assigning an aircraft to different vehicles in turn, if possible. If all aircraft have been assigned and fractional variables remain, a particular aircraft i associated with at least one non-binary $a^x_{i,j}$ (or $a^x_{j,i}$) term is chosen. Branching is now done by considering the path of vehicle x after servicing aircraft i. $|I|+1$ new problems are created where all $j \in I^+$ are chosen in turn, with $a^x_{i,j}$ set to 1 and $a^x_{i,k}$ set to 0 for all $k \in I^+, k \neq j$. Note that by branching on service vehicle path, $|I|+1$ binary variables were set. Again, a more standard approach branching on a particular $a^x_{i,j}$ would have set one.

After some time, all branches will have been pruned. The upper bound has now been verified to be the optimal objective function value, and the solution associated with it defines the optimal service vehicle schedules. It is not clear how long the branch and bound process will take, but it is worth noting that the procedure outlined here lends itself to parallelized computation.

D. Genetic Algorithm

Using a genetic algorithm, it is possible to compare schedules and pick the best that can be found in a fixed, short time. The approach presented here is based on a heuristic originally proposed for aircraft arrival scheduling by [13]. The key insight is that, as in arrival scheduling, airport service vehicle scheduling can be broken down into subproblems of aircraft assignment, sequencing, and scheduling given sequencing instructions. Scheduling given sequencing instructions is trivial. Service vehicles drive to the first aircraft they are to service, service this aircraft when both are ready, move on to the next aircraft to service, and so forth.

Consider $|I|$ random real numbers between 0 and $|X|$. The string of numbers represents vehicle assignment and sequencing instructions for the $|I|$ aircraft, as will be explained. The $|I|$ numbers form an *individual* in the language of genetic algorithms. Table 1 shows an example involving servicing 4 aircraft using 2 vehicles.

The ceiling of a number indicates which service vehicle, indexed from 1 to $|X|$, services the associated aircraft. For example, the numbers associated with aircraft 2, 3, and 4 each have a ceiling of 2 indicating they are serviced by vehicle 2. The ranks of the values of the fractional part of each number represent priority in the order of tasks for each vehicle. So for vehicle 2, aircraft 4 (1.31) is serviced before aircraft 3 (1.48) which is serviced before aircraft 2 (1.63).

The logic of genetic algorithms is as follows: (1) choose an initial population of individuals, (2) test the population, (3) breed new individuals by combining and mutating high performing members of the population, (4) replace low performing members of the population with the new individuals, and (5) return to step (2) unless certain termination criteria are met.

Here choosing an initial population involves generating numerous strings of numbers. Aircraft should be serviced roughly in the order with which they demand service to minimize delay, although adhering to this rule strictly is unwarranted. Here, 20% of individuals' sequencing information is taken from normalized times of

Table I. An individual (numbers in bold) with vehicle assignment and sequencing instructions.

Aircraft	1	2	3	4
Individual	**0.45**	**1.63**	**1.48**	**1.31**
Vehicle Assignment	Vehicle 1	Veh 2	Veh 2	Veh 2
Ordering Instructions	1st Task	3rd	2nd	1st

requested service, $T(i)$ values. The other 80% have sequencing information given by $T(i)$ values added to exponential random variables with the distributional parameter $\frac{\max(T_i - T_j)}{16}$ and then normalized (as in [?]).

Testing the population involves scheduling given the implied sets of instructions, and noting obtained objective function values. Combining individuals means simply taking the mean of values at each spot in each string. Mutating any one individual likewise involves simply replacing any number in an individual with a randomly generated number. Here the genetic algorithm runs, i.e., comparison of alternate schedules continues, until a certain number of generations have been evaluated, i.e., a certain number of calculations have been made.

IV. Computational Studies

A. *Hamburg Airport*

Computational studies were run to test the various approaches to scheduling introduced here. Algorithms were coded in *C++*, with branch and bound techniques for solving static scheduling problems using the *Open Solver Interface*[1] calling the *glpk* solver[2]. The software used is efficient and open source. The first set of studies modeled the operation of passenger buses at Hamburg Airport, home of the CARMA project. Fig. 3 shows the apron and terminal areas at Hamburg Airport, areas modeled in the computational studies presented here. Buses carry passengers both from gates in Terminals 1 and 2 at right to aircraft at stand positions 51-73 at left and vice-versa.

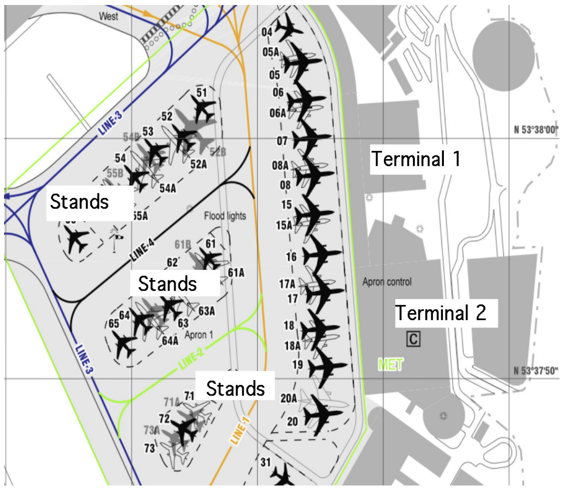

Figure 3. Apron and terminals at Hamburg Airport.

Two hundred scenarios were created, each modeling typical service vehicle dispatch problems at Hamburg Airport involving servicing seventeen aircraft using six passenger buses over the course of an hour. Model parameters were set using data from a previous effort of Airbus in association with the CARMA project. Each scenario was given to the scheduling algorithms, with both branch and bound and the genetic algorithm heuristic used to solve static problems. Note that there are 6^{17} (1.69 x 10^{13}) ways to assign aircraft to service buses, and as many as 17! (3.56 x 10^{14}) ways to sequence tasks after aircraft assignment.

The parameter λ from the objective function of the mixed integer linear program was set equal to 0.9, equating one minute of delay absorbed by an aircraft with nine kilometers of service vehicle travel. It is unclear how sensitive results are to the parameter λ. Future research could study this sensitivity or look for a socially optimal schedule, setting λ based on marginal social costs of a minute of delay absorbed by an aircraft (factoring in potential delay propagation) and a kilometer of service vehicle travel.

Fig. 4 shows histograms of delay absorbed by the seventeen aircraft and distance traveled by the six passenger buses when scheduling using the different algorithms introduced in this paper. There were many instances where delay was zero minutes, indicating no aircraft were delayed waiting for a passenger bus.

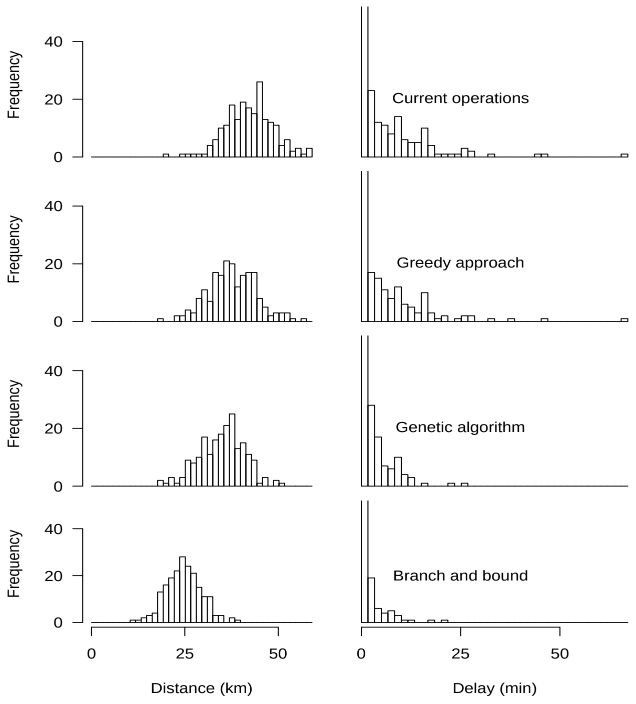

Figure 4. Travel distance and delay absorbed by aircraft at Hamburg Airport.

[1]Information about the Open Solver Interface is available at *https://projects.coin-or.org/Osi*.
[2]Information about glpk is available at *http://www.gnu.org/software/glpk/*.

Planning vehicle routes intelligently significantly reduces the distance passenger buses have to travel. Mean distance traveled per scenario, for the set of all service vehicles, was 42.4 kilometers under current operations. It was 37.8 across the same set of scenarios using the greedy approach, a savings of around 10 percent. Mean distance traveled was 34.9 when using the genetic algorithm, a savings of around 20 percent, and 24.8 when using branch and bound, a savings of 40 percent. While the absolute differences in distances traveled may appear small, keep in mind that service providers will face dispatch problems of the type modeled here several times a day. The cumulative costs of inefficient schedules and fuel costs for a service provider even at a relatively small airport can be significant.

The delay aircraft had to absorb was reduced as well. Mean delay per scenario went from 6.2 minutes under current operations to 5.6 using the greedy approach (roughly -10%). Mean delay was 2.5 minutes when using the genetic algorithm, (-60%), and 1.3 when using the branch and bound approach (-80%).

Looking at Fig. 4, there is significant variation in delay and distance traveled within each of the sets of schedules generated by any one scheduling algorithm. This variation is caused by variation in aircraft arrival and departure schedules. Seventeen aircraft requesting service over the course of an hour may or may not pose a problem for a fleet of six service vehicles, depending on exactly when and where the aircraft request service.

The objective function values less the value when using optimal scheduling, multiplied by ten, provides a metric of schedule inefficiency in minutes of delay and kilometers of vehicle travel. Fig. 5 shows the mean, across scenarios, of schedule inefficiency and computation time using different scheduling approaches. A green triangle marks the performance of current operations, a purple square marks the greedy algorithm, and a red circle marks branch and bound (generating the optimal schedules with no schedule inefficiency by definition). A trail of blue x's tracks the performance of the genetic algorithm heuristic. The performance of the best schedules found at generations 1 and 400 are highlighted. The genetic algorithm surveyed a population of 10,000 possible schedules over 400 generations of 100 births.

Note that the genetic algorithm heuristic is relatively quickly able to generate efficient schedules. In generation 1, ten thousand random perturbations of the schedule provided by the greedy algorithm were studied. This took a few seconds, but often provided a schedule significantly more efficient than those provided by the passive algorithms introduced first in this paper. Once the genetic algorithm was initiated, progress was slow and the algorithm often failed to find better schedules even when the optimal schedule had not yet been found.

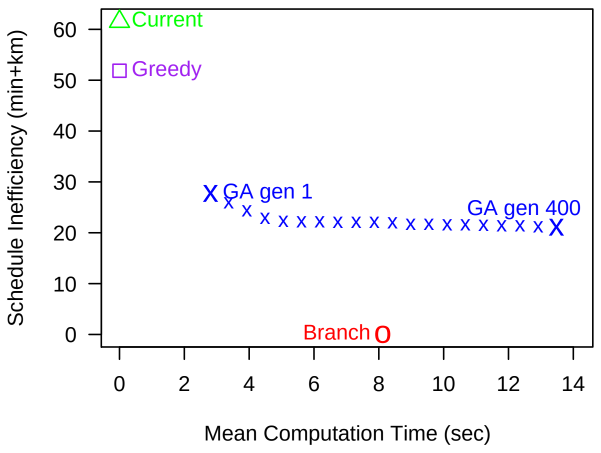

Figure 5. Computational burden and schedule quality.

Fig. 5 shows the mean computation times taken by the various algorithms. One difficulty in using an optimal solution search strategy like the specialized form of branch and bound used here is that the upper bound on how long an algorithm may take to find an optimal schedule is unknown or very large. In the two hundred trials run, the branch and bound algorithm one time took a little over eleven minutes and another time took almost three minutes to find an optimal schedule. Also note that the computational burden of the genetic algorithm approach will scale much better than the branch and bound approach as problem size increases.

B. *Dallas-Fort Worth Airport*

Next, a set of studies was constructed to simulate service vehicle scheduling problems at a geographically larger and busier airport. Dallas-Fort Worth International Airport (DFW) is remarkable both for its size and its high volume of traffic. Distances between gates at DFW are often much greater than at Hamburg Airport. While DFW does not make extensive use of passenger buses, there are plenty of service vehicles including fuel trucks and luggage trailers that require scheduling.

As for Hamburg Airport, scenarios were created simulating operations at DFW. Gate locations and aircraft demand sets were obtained by analyzing data from the Surface Management System at DFW. In 100 scenarios, 1,000 aircraft requested service from 25 service vehicles over the course of 1,080 minutes (18 hours). Branch and bound techniques for optimally solving static scheduling problems were not used, as computation times and memory requirements posed practical problems. Fig. 6 shows histograms of delay absorbed by the aircraft and distances traveled by the service vehicles.

As in the Hamburg Airport study, the distance service vehicles had to travel was significantly reduced. Mean kilometers traveled were reduced roughly 50% when go-

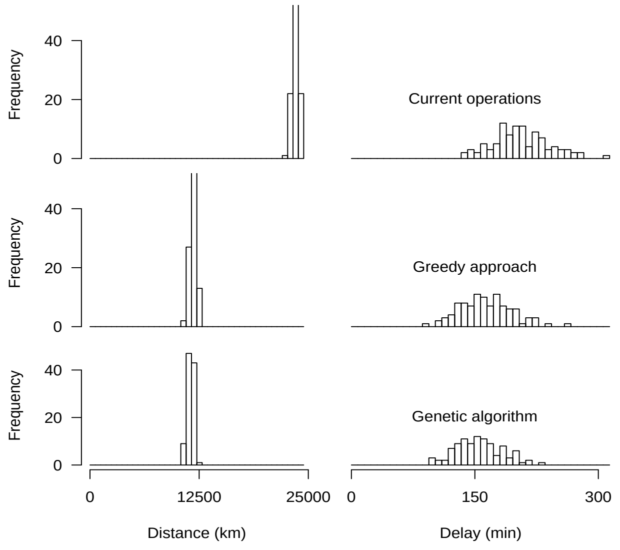

Figure 6. Scheduling 25 vehicles at DFW.

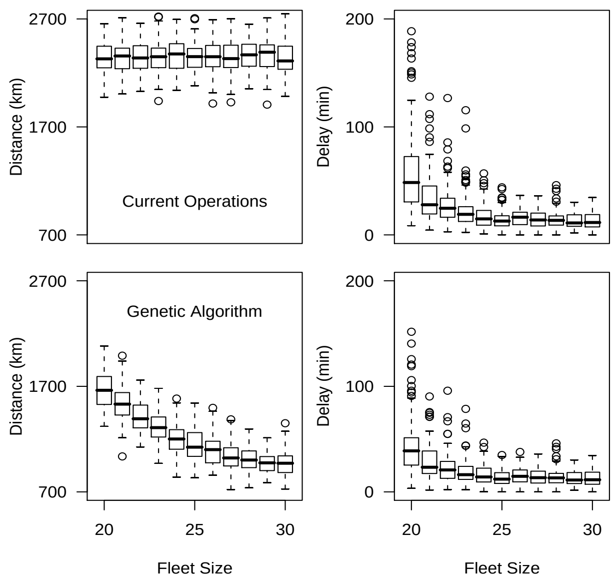

Figure 7. Box-plots of distance and delay vs. fleet size for current operations and genetic algorithm scheduling.

ing from current operations to either a greedy approach or genetic algorithm based scheduling. The genetic algorithm approach reduced mean distance traveled per day about 300 kilometers when compared to the greedy approach. While this is a relatively small amount, the benefits of saving 300 kilometers of travel per day can be significant for an airport service provider. Mean delay absorbed by aircraft was reduced 20% when going from current operations to the greedy approach and 25% when going to the genetic algorithm approach.

A sensitivity study was next performed, investigating how delay and distance numbers change as the number of service vehicles changes. Information from such a study, together with knowledge of the costs of employing vehicles and drivers, could be used to plan vehicle fleet size. The study that generated the results shown in Fig. 6 was replicated eleven times, adjusting fleet size from 20 to 30 vehicles. Fig. 7 shows box-plots of distances service vehicles traveled plus delay aircraft absorbed as a function of vehicle fleet size.

Fig. 7 shows that the distance travelled by all service vehicles decreases as the number of vehicles increases, when using genetic algorithm based scheduling but not under current operations. This indicates that the benefits of intelligent scheduling increase alongside the number of vehicles being scheduled. There are diminishing returns, so that the delay and travel distances saved when going from using 29 to 30 vehicles are lower than those saved when going from 20 to 21 vehicles.

V. Conclusion

The problem of defining schedules for airport service vehicles is introduced in this paper. A framework for addressing such problems is formulated. Algorithms for scheduling are introduced, including one based on solving static vehicle scheduling problems within a moving time window. A novel formulation of such problems is introduced, and complemented by a specially designed branch and bound solution search strategy. Similarities between the airport service vehicle scheduling problem and the airport arrival scheduling problem are noted, and a genetic algorithm heuristic designed for the latter is modified to solve the former. Results of computational studies simulating service vehicle scheduling at one European and one American airport are presented.

Results show significant benefits from planning service vehicle routes based on future demands as opposed to waiting to react to demands as they arise. Both delay absorbed by aircraft and distances traveled by service vehicles can be reduced by 20% or more. Additional research that examines the sensitivity of results to model parameters would be worthwhile. Further research is also needed to test uncertainty in airport arrival and departure schedules, and how this uncertainty impacts service vehicle scheduling. Service vehicles can be a significant source of delay in airline schedules and it is hoped that this paper spurs research interest in airport service vehicle operations.

Acknowledgment

This work was done during the first author's extended visit to Braunschweig, Germany in support of Action Plan 21. The authors would like to sincerely thank the National Aeronautics and Space Administration for granting us this opportunity and the Deutsches Zentrum für Luft- und Raumfahrt for arranging the visit. The exchange would not have been possible or fruitful were it not for the efforts of Sandra Lozito, Christoph Meier, Dietmar Bohme, Daniel Schaad, Friedrich Posnien, Christoph Mohlenbrink, Bassam Mussafar, and Harry Swenson.

References

[1] H. Idris, B. Delcaire, I. Anagnostakis, W. D. Hall, R. J. Hansman, E. Feron, and A. Odoni, "Observations of departure processes at Logan Airport to support the development of departure planning tools," 2nd USA/Europe air traffic management R&D seminar, 1998, Orlando, USA

[2] I. Anagnostakis, H. R. Idris, J.-P. Clarke, E. Feron, R. J. Hansman, A. Odoni, and W. D. Hall, "A conceptual design of a departure planner decision aid," 3rd USA/Europe air traffic management R&D seminar, 2000, Naples, Italy.

[3] F. Carr, G. Theis, J.-P. Clarke, and E. Feron, "Evaluation of improved pushback forecasts derived from airline ground operations data," Journal of Aerospace Computing, Information, and Communication, 2005, Vol. 12, pp. 25-43.

[4] R. Treude, "Optimizing performance by monitoring the turnaround process. 5th R& D Symposium, 2005, Braunschweig, Germany.

[5] J. Jaw, A. Odoni, H. N. Psaraftis, and N. H. M. Wilson "A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows," Transportation Research Part B, 1986, Vol. 20, pp. 243-257

[6] L. D. Bodin, and T. Sexton, "The multi-vehicle subscriber dial-a-ride problem," TIMS Studies in Management Science, 1986, Vol. 26, pp. 73-86.

[7] J.-F. Cordreau and G. Laporte, "A tabu search heuristic for the static multi-vehicle dial-a-ride problem," Transportation Research Part B, 2003, Vol. 37, pp. 579-594.

[8] J.-F. Cordreau, "A branch-and-cut algorithm for the dial-a-ride problem," Operations Research, 2006, Vol. 54, pp. 573-586.

[9] E. Feuerstein and L. Stougie, "On-line single-server dial-a-ride problems," Theoretical Computer Science, 2001 Vol. 268, pp. 91-105.

[10] S. O. Krumke, W. E. de Paepe, D. Poensgen, and L. Stougie, "News from the online traveling repairman," Theoretical Computer Science, 2003, Vol. 295, pp. 279-294.

[11] V. Bonifaci and L. Stougie, "Online k-server routing problems," 4th Workshop on Approximation and Online Algorithms, Lecture Notes in Computer Science, 2006, Zürich, Switzerland.

[12] P. Jaillet and M. Wagner, "Online routing problems: value of advanced information and improved competitive ratios," Transportation Science, 2006, Vol. 40, pp. 200 -210.

[13] A. T. Ernst, M. Krishnamoorthy, and R. H. Storer, "Heuristic and exact algorithms for scheduling aircraft landings," Networks, 1999, Vol. 34, pp.229-241.

Author Biographies

Kenneth D. Kuhn (BA'01-MS'02-PhD'06) was born in Washington, D.C. on October 26, 1978. He obtained a BA degree in math at the Johns Hopkins University in Baltimore, Maryland, USA in 2001, a MS degree in operations research at the University of California in Berkeley, California, USA in 2002, and a PhD in civil (transportation) engineering at the University of California in Berkeley, California, USA in 2006.

He worked as an Aviation Systems Researcher at the University Affiliated Research Center after graduation. He currently works as an Aerospace Engineer at NASA in Moffett Field, California, USA. He has four published journal articles and made numerous conference presentations. His research interests include near terminal aviation operations, the impacts of weather on aviation, and, more generally, mathematical representation of transportation related systems.

Dr. Kuhn is a member of the American Institute of Aeronautics and Astronautics, as well as the Institute for Operations Research and the Management Sciences.

Steffen Loth was born in Ludwigsfelde near Berlin, Germany, 09.09.1971. After his apprenticeship as a skilled electronic worker he started his academic studies in the field of Transport, Aeronautics (flight guidance and air transport) at the Technical University Berlin, Institute of Aerospace, Berlin, Germany 10/92 12/02. He got his diploma in Aeronautics in 2002.

From 1991 to 1992 he he was doing his military service. During his studies he worked as a working student at BMW Rolls Royce Aero Engines and the Gesellschaft fuer Luftverkehrsforschung mbH (GfL) and as a tutorial assistant at the Institute of Flight Guidance at the Technical University Berlin. After his studies he worked as a junior consultant with GFL for two years and was involved in different projects concerning safety, capacity and environment of German and European airports. In 2005 he joined the Institute of Flight Guidance in Braunschweig of the German Aerospace Center (DLR) as a research assistant. He is involved in the research of Advanced Surface Movement Guidance and Control Systems (A-SMGCS) and the optimization of airport operations (including ground handling). He works as a coordinator of various projects at Hamburg Airport.

Mr. Loth is the representative of DLR within the EUROCAE (European Organization of Civil Aviation Equipment) Working Group 41 (Surface Movement Guidance and Control System).