SCHEDULING FLIGHTS AT HUB AIRPORTS

DAN TRIETSCH
Department of Administrative Sciences, Naval Postgraduate School,
Monterey, CA 93943, U.S.A.

(Received 10 October 1990; in revised form 15 December 1991)

Abstract—In a typical hub airport, incoming flights from many origins feed outgoing flights to equally many destinations. If an incoming flight is late, outgoing flights that are fed by it may also be delayed. Alternatively, aircraft may leave before some feeding flights arrive, thereby incurring high misconnection penalties. By optimizing the scheduled ground time of each plane, we can minimize the expected sum of costs and penalties. In this paper we develop generalized newsvendor models for this purpose. In particular, we investigate in detail a pure-waiting model in which no misconnections are allowed and discuss a no-waiting model in which aircraft that are ready to leave never wait for late-feeding flights. The models show that to maximize the system level of service at a given cost, the level of service of individual flights should be allowed to vary.

The models can also be applied to similar problems such as express parcel deliveries and ground transportation hubs. The problems we address are nonlinear and highly combinatorial, so for life-size problems it is not practical to solve them to optimality. Therefore, an important part of the paper is devoted to heuristic solutions and the promising numerical experience achieved with them.

INTRODUCTION

Our concern in this paper is scheduling connections at a hub facility, where arrivals and departures are subject to stochastic variation. It is presented in terms of passenger air service but can also be applied to other hub operations, for example, express parcel deliveries and break-bulk/consolidation operations at central warehouses.

The basic principle is that the arrivals feed the departures. Hence the arrivals and the departures must be coordinated. A complicating factor is that the arrivals may run behind schedule. The arrival time is a stochastic variable influenced by a multitude of conditions, both in and out of the airline's control. For instance, weather conditions at the hub itself and in other airports may impact the on-time performance. Similarly, when too many aircraft are scheduled to land or take off within an interval of 2 or 3 min, some stochastic queueing delays will result.

In the main model we present, all departures that need to be fed by late arrivals wait as long as necessary—and thus also may be delayed and incur lateness penalties. We also discuss a model in which departures that are ready to leave do not wait and receive misconnection penalties. The objective in both cases is to minimize the total expected time and penalty expenditure.

By including the utility of the passengers as well as the expenses of the airline in the objective function, we are incorporating two important parts of the system into the model. The argument is that taking the utility of the passengers into account is conducive to business success in the modern competitive environment (Peters, 1987). Thus the price of a delay should include the out-of-pocket expense (e.g. overtime) as well as the passengers' time value. Similarly, misconnection penalties should include actual expense and loss of income to the airline as well as loss of goodwill and the time value of the involved passengers. The models we develop, however, apply even if the airline does not wish to take the passengers' utility into account. In that case, we simply set the passengers' time value to zero.

Alternatively, and perhaps more conventionally, the passengers' consideration could be handled by a service-level constraint. Such service constraints usually call for arrivals (or departures) to be on time a certain fraction of time, say 95%. Nevertheless, our approach yields less arbitrary service-level constraints and allows them to differ between flights. We can also start with a global service-level constraint and translate it to a penalty

This work was supported by the NPS Foundation Research Program.
function as in this paper's models. Using this approach, examples exist that show that a
service level that fits a given passenger distribution may not be appropriate for another
passenger distribution. This argues the superiority of the present model over the level-of-
service approach.

To achieve tractability, we assume that arrival delays are independent of each other.
In reality, this assumption is not totally justifiable, because conditions at the hub may
impact all arrivals and weather conditions across several airports may be correlated. In
practice, if we have adverse weather conditions at the hub, we may have to resort to a
revised schedule that allows more time between flights (Teodoric, 1985). We may also
have to accept more misconnections. If the problems at the hub are due to congestion
(Andreotta and Romain, 1987; Odoni, 1987; Rue and Rosenshine, 1985), we may expect
some degree of dependence among the delays. We reduce this effect, however, by specify-
ingen separation constraints between adjacent arrivals or departures.

The literature on aircraft scheduling to date is concerned mainly with deterministic
models, so it is not directly applicable here. For instance, see Etschmaier and Mathaisel
(1985). A stochastic model similar to ours was presented by Hall (1985). It is concerned
mainly with transit passengers where several feeder lines serve a single train. Delays are
assumed to be exponentially distributed. The policy is no-waiting, assuming there will be
a later train. The model minimizes the total expected passenger time in the system.

The first model in this paper—in which departures wait for all connecting passen-
gers—is an extension of a project purchase orderscheduling model developed by Ronen
and Trietsch (1988, 1991). That model, in turn, is a generalization of the classic newsboy
problem. It is concerned with ordering stochastic-lead-time project-items at the right time
to minimize the holding cost and the expected project lateness penalties incurred if any
items are late. The project items are analogous to our feeder flights, and project lateness
is analogous to departure delays. In the project problem, however, we feed just one
project, instead of several departures. In addition, there is no parallel to the congestion
issue. Thus the model requires considerable modification for our purpose.

In general, even without separation constraints, our objective functions are nonlinear
and not necessarily convex. Therefore, no proof is offered that the minima we find for
them are necessarily global. They also involve integrals and are thus computationally
intensive. To ameliorate these problems, we introduce a bound that does not require
evaluating integrals and yields very close approximations to the optimal solution. At the
very least, this bound makes the search for the minimal cost solution easier (“hot start”),
but its main purpose is to serve instead of the optimal solution.

This solution tends to call for almost simultaneous departures (but much less so for
arrivals, assuming different lead-time variances). Because scheduling the flights too
closely leads to queueing delays, we incorporate separation constraints between adjacent
flights. Once these are introduced, the model also becomes highly combinatorial, because
there are exponentially many possible sequences. This implies that it may not be feasible
to solve large instances of the model to optimality. We show, however, that the bound
solution discussed above is instrumental in developing efficient heuristics for this case as
well. Indeed, one of the main contributions of the paper is a heuristic procedure that
requires about 2 sec central processing unit (CPU) time on an Amdahl (W. R. Church
Computer Center, Naval Postgraduate School) 5900-500 and achieves target function
values that are practically as good as those achieved without the heuristic in 200 to 600
sec CPU time.

Section 1 introduces some notation and formulates the basic waiting model. Section
2 gives an approximate solution when the arrival and departure times (our decision
variables) are not constrained. Section 3 presents a capacitated waiting model in which
separation constraints are imposed. Section 4 presents a capacitated no-waiting model,
with a fixed penalty cost for misconnections. In Section 5 we discuss the interaction
of our models with other operational decisions (e.g. optimizing the airspeed and the
gate-assignment problem). Section 5 also discusses models in which both waiting and
misconnection penalties may be present. Section 6 is the conclusion.
I. A BASIC HUB SCHEDULING MODEL WITH WAITING

Suppose we have \( n + 1 \) origin/destination points indexed by \( i = 0, 1, \ldots, n \), where 0 is the index of the hub. Traffic from origin \( i \) to destination \( j \) (\( i, j \neq 0 \)) may be routed through the hub, using Flight \((i,0)\) connecting to Flight \((0,j)\). We use the following notation:

- \( s_i \): scheduled arrival time for Flight \((i,0)\) — a decision variable
- \( T_j \): scheduled departure time for Flight \((0,j)\) — a decision variable
- \( s_i^* \): optimal \( s_i \) value
- \( T_j^* \): optimal \( T_j \) value
- \( D_{ij} \): average number of passengers from origin \( i \) to destination \( j \) that use the hub
- \( d(i) \): final destination of the aircraft assigned to \((i,0)\)
- \( o(j) \): origin of the aircraft assigned to \((0,j)\); that is, \( d(i) \)'s inverse
- \( I_i \): \( \{ i \mid D_{ij} > 0, i \neq 0 \} \cup \{ o(j) \} \); that is, an index set including all origins from which passengers to destination \( j \) (\( j = 1, 2, \ldots, n \)) go through the hub (except the hub itself) and the origin of the aircraft assigned to \((0,j)\). (A delay in any arrival from \( k \in I_i \) may delay the departure to \( j \).)
- \( J_j \): \( \{ j \mid D_{ij} > 0, j \neq 0 \} \cup \{ d(i) \} \); that is, an index set including all destinations to which passengers from origin \( i \) (\( i = 1, 2, \ldots, n \)) go through the hub (except the hub itself) and the destination of the aircraft assigned to \((i,0)\). (A delay in the arrival from \( i \) may delay the departure to any \( k \in J_j \).)
- \( c \): average time unit value per passenger
- \( p \): penalty per time unit per passenger during unscheduled delays (\( p \geq c \))
- \( b_i \): marginal time unit value of the aircraft and crew assigned to Flight \((i,0)\) during regular operations (thus the value on Flight \((0,j)\) is \( b_{o(j)} \)). Includes only costs that increase if the schedule calls for longer operation hours.
- \( C_i \): \( b_i + \sum_{j \in D_{ij}} p D_{ij} \); \( i = 1, \ldots, n \); that is, the time value of all the passengers on Flight \((i,0)\), except the passengers whose final destination is the hub itself, plus the time unit value of the aircraft. Scheduling Flight \((i,0)\) one time unit earlier (without changing the other flights) will imply a cost of \( C_i \), because the passengers from \( i \) to \( J_j \) and the plane will have a longer wait at the hub (on the other hand, such a schedule change may decrease the penalties).
- \( B_j \): marginal time unit value of the aircraft and crew assigned to the \((0,j)\) segment during unscheduled delays. \( B_j \geq b_{o(j)} \) and is strictly larger when overtime rates are involved.
- \( P_j \): \( B_j + \sum_{i \in J_j} p D_{ij} \); \( j = 1, \ldots, n \); that is, \( P_j \) is the passengers’ penalty cost incurred if the departure of Flight \((0,j)\) is delayed by one time unit, plus the time unit value of the aircraft during delays. The cost of an unscheduled delay of one time unit for Flight \((0,d(i))\) is \( P_{d(i)} \).
- \( F_{i}(t) \): Cumulative distribution function (CDF) for arrival delays in Flight \((i,0)\). Assumed independently distributed i.i.d. \( F(i,0) \) is the probability the flight will arrive on time.
- \( F_{0}(t) \): CDF for delays in takeoff for Flight \((0,j)\) that are not due to waiting for late connecting flights. If all feeding flights are in (or departures do not wait), \( F_0 \) is for delays relative to \( T_j \); otherwise, it is for delays relative to the revised departure time. As an approximation, we assume that \( F_0 \) is identically distributed and independent \( \forall j \) and that it is stationary, that is, it does not vary as a function of the schedule itself or of changes in the schedule. (This CDF will become useful to us only in Section 4.)
- \( f_{i}(t) \): density function of \( F_{i}(t); i = 0, \ldots, n \)

Note that the values \( b_i \) and \( B_j \) do not include the cost of fuel or of owning the aircraft. The argument is that those costs are already sunk by the time we look for the schedule. There may be circumstances under which the aircraft can be utilized elsewhere later if it arrives at \( j \) early enough. In such cases the cost of owning the aircraft may have
to be taken into account. In these cases the lateness penalty may also have to be increased to take into account possible disruptions downstream.

We assume that the departure of Flight \((i,0)\) from \(i\) is scheduled at \(S\) minus the nominal duration of the flight. Our objective is to minimize the total scheduled time costs plus the total unscheduled delay penalties that are a function of the decision variables, that is,

\[
\text{Min } Z = \sum_{j=1}^{m} B_{o_{i}j} T_j - \sum_{i=1}^{n} b_i S_i + \sum_{i=1}^{n} \sum_{j=1}^{m} cD_{i,j}(T_{i} - S_{i})
\]

\[
+ \sum_{j=1}^{n} P_{j} \int_{T_{j}}^{\infty} \left(1 - \prod_{k \neq j} F_{k}(t - S_{k}) \right) dt
\]

(1.1)

The first two elements of the objective function capture the deterministic cost of the time the aircraft and crews are scheduled to be at the hub. For instance, if the plane from 3 continues to 4, then \(b_{o_{i}j} = b_{ij}\) is the value of a time unit of this plane. The plane's contribution to the first two elements of the objective function is \(b_{ij}T_{i} - b_{ij}s_{ij} = b_{ij}(T_{i} - S_{i})\).

Similarly, the third element captures the time value of all the passengers, except those whose origin or destination is the hub itself, under the assumption there will be no delays. The passengers to the hub are excluded because the time they spend in the system is invariant under the choice of \(S\). The passengers from the hub are also excluded because, unless there is a delay, the time they spend in the system is invariant under the choice of \(T_{j}\).

The fourth element of the objective function gives the expected penalty for unscheduled delays in departures. Note that \(P_{j}\) also captures the time value of the passengers originating at 0—as they too are subject to unscheduled delays. The integrand is the probability that at least one feeding flight, or Flight \([o_{ij},0]\), which may not carry passengers to \(j\) [e.g. if \(o_{ij}\) is itself], will not arrive by time \(t\) in such a case Flight \((0,j)\) will not be able to depart by this time either. Thus the integral gives the expected delay. Multiplying the integral by \(P_{j}\) we get the expected penalty for Flight \((0,j)\). Summing for all destinations, we get the total expected penalty. This penalty does not include the expected penalty due to other delays, as per \(F_{i}\), which, under our simplifying assumptions on \(F_{i}\), are not influenced by the decision variables.

The objective function is constructed under the implied assumption that once the last connecting arrival has arrived the departures can take place immediately. This would be true if connections, as well as servicing the aircraft, required zero time. In practice we must add extra time to the model to allow for connections and for servicing the aircraft. For tractability, to take care of the necessary connection time, we add a constant to the model's \(T_{j}\). This is a simplifying assumption because the connection time is really the maximum of several random variables, the distributions of which may be influenced by the schedule we set.

Note that eqn (1.1) assigns no benefit to arrivals ahead of schedule. Such early arrivals are recorded as on-time arrivals. These early arrivals are not beneficial to connecting passengers and are not very useful to the passengers going to the hub because they cannot plan on them.

By minimizing eqn (1.1) we also determine the optimal probability of connection for each origin/destination pair. If these probabilities are large, then it is not likely that arrivals will be scheduled \textit{after} connecting departures. If such a probability is low, however, there is nothing in the model to prevent a negative connection window. Such negative connection windows will probably appear positive in the published schedule, after we add the constant connection time to \(T_{j}\).

To continue, assuming that \(Z\) is differentiable, and using the Leibnitz method, we obtain the following partial derivatives for eqn (1.1):

\[
\frac{\partial Z}{\partial S_{i}} = -cD_{i,j} + P_{j} \int_{T_{j}}^{\infty} f_{i}(t - S_{i}) \prod_{k \neq j} F_{k}(t - S_{k}) dt - b_{ij}, \quad \forall i
\]

(1.2)
Equations (1.2) and (1.3) provide $2n$ equations for $2n$ decision variables. By observing the objective function, however, it is clear that if we add a constant to all the variables, $Z$ will not change. Therefore, after solving the model we can shift the whole schedule by any constant until the result fits our needs best. Thus we can set one of the variables arbitrarily, and use only $2n - 1$ of the equations.

To optimize $Z$, we can search for its minimum directly or set the partial derivatives to zero. Either way, this can be done only numerically. Though eqns (1.1) and (1.2) involve integrals, when we resort to numeric methods we can represent them by appropriate sums for any required degree of accuracy. This is cumbersome, however; so, where possible, we concentrate on approximate solutions that avoid direct computations of integrals. In this context note that eqn (1.3) does not involve integrals.

In reality, there are additional constraints imposed. Specifically, it is necessary to maintain minimum separation constraints between flights, so the optimal solution of eqn (1.1) may be infeasible. Even if that is the case, solving eqn (1.1) provides an approximate initial solution that can then be modified according to the additional constraints. We refer to the unconstrained model as the “uncapacitated case,” since it implies that the capacity of the airport to handle arrivals and takeoffs is unlimited. In the next section we continue with the uncapacitated case, deferring the separation constraints issue to Section 3.

2. THE UNCAPACITATED CASE—AN APPROXIMATE SOLUTION

In this section we approximate the solution of eqn (1.1) by a two-stage solution, to which we refer as the “uncapacitated scheduling approximation.” In the first stage we assume all the departures leave at the same time, $t^*$, and schedule the arrivals approximately. In the second stage we hold the resulting arrival schedule constant and optimize the departure schedule.

The first stage, in which we assume all departures are at the same time, is predicated on numerical experience that shows that the mean time between departures in the minimal uncapacitated solution is very small relative to the mean time between arrivals. Departures tend to be close to each other because they (typically) share many feeding flights, so they are likely to be ready to depart at the same time. The (small) differences in departure time, which we calculate at the second stage, are due to two factors: the impact of feeding arrivals that are not common to the departures, and the varying number of passengers who originate at the hub as compared to the total number of passengers and the time value of the aircraft. (See the numerical results below, where we discuss this issue further.)

We refer to $F_i(t^* - S_i)$ as the “service level” of Flight $(i,0)$. That is, the service level of a flight is the probability that it will be on time to feed all the departures as per the schedule. We refer to $t^* - S_i$ as the “safety time” of Flight $(i,0)$. The “optimal service level,” $F_i^*$, is defined as $F_i(t^* - S_i)$. Given $F_i^*$, it is straightforward to deduce $S_i^*$, and vice versa. [Service level is usually defined as $F_i(0)$, i.e. the probability the flight will arrive on time, $S_i$. For our purpose, however, it is the connections that count. Furthermore, by changing $S_i$ nominally, without changing $t^*$, we can manipulate $F_i(0)$ to any level we want, without changing the real schedule. Hence our modified definition.]

We define the “system service level” as the multiplication of the individual service levels of all the arrivals. It approximates the probability that connections will be made as per the schedule and the departures will be on time. [An exact expression for this purpose should take into account that even if a flight is late, some flights do not have to wait for it. It should also take into account the impact of $F_i(0)$]

An upper bound for $F_i^*$, $F_i^U$

To approximate $F_i^*$ (and thus $S_i^*$), we now develop an upper bound for it. This upper bound is valid even without assuming all departures are scheduled at $t^*$, in which case it gives an upper bound on the optimal probability that the arrival of $(i,0)$ will occur before the earliest departure. For this purpose we should treat $t^*$ as the earliest departure time.
We begin by rewriting eqn (1.2), in a slightly rearranged form:

$$\frac{\partial Z}{\partial S_i} = -[b_i + \sum_{j \in b} cD_{ij} + \sum_{j \in b} P_j \int_{s_i}^{t_i} f_j(t-S_j) \Pi_{c \in b \setminus \{j\}} F_i(t-S_k) dt], \quad \forall i (2.1)$$

First note that, by definition, $b_i + \sum_{j \in b} cD_{ij} = C_i$. Next, if we substitute $1$ for $\Pi_{c \in b \setminus \{j\}} F_i(t-S_j)$ in the integrands of eqn (2.1), the integrals can only increase. Setting $t^*$ (the earliest departure time) as the lower limit of integration tends to increase them further. But once we do that, the integrals become identical, and the value of each of them is $1 - F_i(t^* - S_i)$. After some algebra we obtain

$$F^*_i = F_i(t^* - S_i) \leq 1 - C_i/\sum_{j \in b} P_j = F^*_i, \quad (2.2)$$

Let $S_i$ satisfy $F_i(t^* - S_i) = 1 - C_i/\sum_{j \in b} P_j = F^*_i$, and we obtain an approximate arrival schedule, which we propose as the solution of the first stage. Another approximation, available from the author, is slightly more complex, but consistently outperforms the solution above in terms of the first stage. Nevertheless, this alternative solution is usually inferior as a starting point for the second stage. Furthermore, it tends to schedule the arrivals closer together and thus requires more adjustment when separation constraints are introduced later.

Given the first-stage solution, in the second stage we search for $T_i$ values that set the partial derivatives in eqn (1.3) to zero, thus optimizing the departures schedule for the given arrivals schedule. Not surprisingly, when we do this, the optimal departure times tend to be negative (because they correct for the bias of the bound). There is no difficulty, of course, to shift the schedule to fit the desired time of departure. (The superiority of the alternative solution in terms of the first stage is that it has no such bias to correct for in the second stage.)

As reported in more detail below, for problems with $49 + 1$ cities, this solution method yielded typical target function values 0.06% to 0.3% above the minimum obtained by an exhaustive local search. It did so within 0.22 sec CPU time (on an Amdahl 5990-500), as contrasted with 4.5 to 8 min CPU required to find the local minimum in these cases.

### 3. THE CAPACITATED CASE

In the uncapacitated case the solution of eqn (1.1) usually calls for close departures. Therefore, when the capacity of the airport is limited we should schedule the departures as close to each other as possible.

Indeed, some airlines schedule many departures at practically the same time (e.g. 15 flights between 6:00 p.m. to 6:05 p.m.). The purpose of this practice, however, is not necessarily a result of using unconstrained optimization. It may be done to list attractive departure times (i.e. departures likely to appear on the first screen monitored by travel agents). The results are long queueing delays and an obvious decrease in the total system utility.

Fortunately there are indications that this practice may be curtailed in the future. Indeed, some airports (e.g. San Francisco International Airport) now forbid aircraft to taxi when too many other aircraft are in line ahead. They also forbid moving the plane a few feet from the gate to create an "on-time" departure before the plane is allowed to taxi. Such regulations will encourage airlines to take airport capacity into account, if they want to avoid having to report many late departures. Also, airlines may eventually realize that passengers spending 25 min or so in a takeoff queue, on a bright day, and observing that 90% of the aircraft in line are operated by their own carrier, will probably blame it for the delay.²

²This remark is based on the author's personal experience in one of the major hubs of a large airline several years ago. More recently, at another major hub of the same airline, the author counted 37 departures scheduled between 2:44 and 3:00 p.m.
The finite capacity model we'd like to solve would incorporate eqn (1.1) as before, plus an additional set of constraints:

\begin{align}
|S_i - S_j| &\geq G_A, \forall k \neq i \\
|T_f - T_l| &\geq G_D, \forall k \neq j
\end{align}

where $G_A$ and $G_D$ specify minimal scheduled gaps between arrivals and departures, respectively (we do not assume $G_A = G_D$). Unfortunately, such a model would be very difficult to implement, due to the large number of constraints. Therefore we look for simpler models.

Arguably, the inclusion of these separation constraints, which are operational constraints by nature, is not appropriate in a model that is essentially a planning tool. Nevertheless, ignoring these constraints during planning may lead to schedules that are too optimistic, which will later cause operational difficulties. This point is another justification, however, to look for simpler models.

Note that, especially for arrivals, the separation constraints we set in the model need not be exactly the same as the capacity of the airport. Given the arrival time variance, we cannot expect the flights to arrive exactly as per the schedule we set or even in the same sequence. Nevertheless, separation gaps in the schedule will tend to spread the arrivals more evenly. To prevent excessive queueing, it may be a good idea to schedule higher separation constraints than the airport's operational separation constraints. For departures, however, one may argue that the optimal separation constraints should be as low as the airport permits.

To reiterate, the problem is that if we schedule all flights at the times determined by the unconstrained solution, the separation constraints will likely be violated. It is easy to identify blocks, or groups, of flights whose unconstrained schedules cannot coexist without violating the separation constraints. The flights in these groups have to be forced apart enough to accommodate the separation constraints.

The easiest way to do this is to retain the average times for the group as a whole and to maintain the sequence within the group as in the unconstrained approximate solution. For small separation constraints, it is typical for the arrivals to comprise several such groups (some with just one flight). This is a result of different lead-time variances of the arrivals; if the variances are close to each other, the arrivals will also be close to each other. The departures tend to belong to one group only regardless of the lead-time variances, which do not impact the departure times at all.

When the separation constraints grow, however, this method may push large aircraft with many passengers too far from their optimal schedule while favoring smaller aircraft. Thus a modification is required that also takes into account the costs.

At the other extreme, if we limit our attention to the deterministic costs alone and neglect the stochastic elements of the objective function, then the objective function becomes

\[
\min \left\{ \sum_{i=1}^{n} T_i [b_{d(i)} + c \sum_{j=1}^{n} D_i] - \sum_{i=1}^{n} S_i [b_i + c \Sigma_{j=1}^{n} D_i] \right\}
\]

Equation (3.3) lends itself to decomposition and can be easily optimized by minimizing the sum involving $T_i$ and maximizing the sum involving $S_i$. The values within the brackets measure the time value of each flight—excluding the passengers to or from the hub. All that remains is to sort the flights by these time unit value measures. Arrivals should be sequenced in ascending order and departures in descending order.

Our numerical experience shows, as expected, that for small separation constraint values the stochastic order determined by the solution of the unconstrained version (or by the approximation) performs well, whereas for very large separation constraints (over 5 min in our examples) the deterministic solution takes the lead. At this stage, however, both sequences yield very inferior results relative to other sequences that can be generated. Thus we need a mechanism to generate better sequences. We can do this by switching
adjacent flights until the resulting sequence cannot be improved further by such switches.

In addition to pair-switching, one can also search for the best schedule for a given sequence that maintains the separation constraints. When this is done until the solution cannot be improved by any slight change in the schedule or switching, we obtain a local minimum. For further algorithmic details of the procedures described above, see the Appendix.

As discussed above, the gaps between departures are typically kept to the minimum, that is, \( G_0 \). Therefore, if the first departure has to wait for connecting passengers, then all departures will be equally delayed, with the possible exception of flights that do not have delayed passengers, which may still leave on time while the first departure is waiting. If a departure is delayed due to other problems, the following departures will not be delayed. Be that as it may, this modifies our scheduling problem somewhat. We list two alternative approaches to the issue, in ascending order of complexity and performance:

1. Require that all flights arrive before the first departure and use the results of the former section to calculate the arrival times (where \( t^* \) is defined as the earliest departure).
2. Use the same safety times as above, but for each arrival use the scheduled departure of the first flight it feeds as \( t^* \). For instance, if Flight (1,0) has a safety time of 24 min and the first flight to which it connects leaves at 12:02, then we schedule it at 11:38.

Under the first approach, which is the one used in the numerical examples below, we are in effect treating all departures as a single outgoing flight. This implies that all arrivals must be in before the first departure occurs. Therefore we should sequence the departures as per the deterministic sequence. Note that when we use The Uncapacitated Scheduling Heuristic in this context, we have to replace the exact departure times (obtained by solving 1.3) by a single value. The average of these exact departure times worked very well in our examples.

**Numerical results and a heuristic solution**

The numerical experimentation reported in this paper was done at two stages. First, small examples with four or eight cities were solved and used to look for patterns. Then, in response to a referee's request, a large experimental design was performed, with the intention of corroborating the results observed earlier. Generally, all the expected patterns were verified, and some additional insights were gained at the same time. (The former small examples are available from the author.)

In addition, the new large runs were used to refine a heuristic procedure in which, instead of minimizing the real target function, surrogates are used. The first surrogate is used only to schedule the flights for a given sequence. Instead of minimizing the target function, we search constrained values that will yield the same service level as those yielded by the unconstrained approximation. Thus, the service level is a surrogate for the target function. The second surrogate is used only for switching pairs and calls for minimizing the distance between the current solution and the unconstrained one. On average this heuristic yielded target function values 0.06% (i.e. 0.0006) above those of the best solution obtained without it, and it did so with less than 1% of the CPU time consumption!

The main results of the small runs that we set out to corroborate were as follows:

1. In the unconstrained model, the optimal system service level remains virtually unchanged when \( n \) grows from 4 to 8, as long as the ratio of hub passengers (i.e. passengers that originate or terminate at the hub) to other passengers is not changed (see Table 1, in which examples 1, 2 and 3 have 4 + 1 cities and examples 1a, 2a and 3a have 8 + 1 cities; examples 1 and 1a have \( c = $10, p = $50, b_l = 400, \) and \( B_r = $500; \) examples 2 and 2a have \( p = $20; \) and examples 3 and 3a have \( p = c = 10 \) and \( B_r = b_l = $400)\).
2. In the unconstrained model, flights with relatively high lead-time variances require disproportionally more safety time (i.e. they require higher service levels).
Scheduling flights at hub airports

Table 1. System service levels (in percentages) and target function values for 4 + 1 and 8 + 1 examples

<table>
<thead>
<tr>
<th>Example</th>
<th>Service level</th>
<th>Target value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>74.674</td>
<td>1573.90</td>
</tr>
<tr>
<td>1a</td>
<td>74.610</td>
<td>1802.97</td>
</tr>
<tr>
<td>2</td>
<td>47.457</td>
<td>1257.70</td>
</tr>
<tr>
<td>2a</td>
<td>47.433</td>
<td>1519.53</td>
</tr>
<tr>
<td>3</td>
<td>15.082</td>
<td>955.14</td>
</tr>
<tr>
<td>3a</td>
<td>14.916</td>
<td>1224.67</td>
</tr>
</tbody>
</table>

3. Flights with relatively few hub passengers tend to be scheduled to depart earlier than flights with many hub passengers and this effect is decreasing with $p$.

To corroborate the first point, we started with a 7 + 1 problem and found the optimal service level (associated with the minimal target function value as per a local search). Then, each of the seven cities was split to a block of seven subcities to obtain a 49 + 1 problem. The splitting was done in such a manner that the fraction of hub passengers remained constant, and there was no demand between two subcities in the same block. There are three sizes of aircraft assigned to the blocks (two blocks with small aircraft; three medium and two large). For a particular passenger distribution (see Table 2) and parameters ($b_1 = 400–600$, $B_1 = 480–700$, $c = 15$, $p = 25$), the system level of service for 7 + 1 cities was 37.5% and for 49 + 1 it was 38.2%. The similarity held for each block separately as well; in fact, the service level of a flight in any block of the 49 + 1 problem was very close to the seventh root of the service level of the flight in the 7 + 1 problem from which the block was generated. That is, in the 49 + 1 problem the service level of each flight was much higher, ranging from 96.0% to 99.2%, as compared with 75.8% to 94.2% in the 7 + 1 case. As a result, again, the target function value (after adjusting for the greater number of flights) was higher in the 49 + 1 case by 18.36%. Although the results in Table 1 are even more impressive, the system level of service was indeed essentially the same, and a disutility of scale was detected. Note that because the normal distribution was used for this example, raising the service level beyond a certain limit does not require much additional safety time; other distributions with thicker tails would produce higher additional safety times and thus higher target function increases. For instance, with exponential lead-time distribution, the target function increase in this example would be about 70%. Furthermore, with any distribution, this disutility of scale is higher when separation constraints are imposed.

To corroborate the other points, additional blocked 49 + 1 examples were run. In one series of 24 runs, the demand distribution was identical within each block, but the arrival time variance ranged from $4^2$ to $28^2$. This series, to which we refer as the "variance testing series," provided ample opportunity to check the second point. In another 24 runs

Table 2. Passenger demand matrix for the 7 + problem

<table>
<thead>
<tr>
<th>To:</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>From:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>9</td>
<td>11</td>
<td>18</td>
<td>20</td>
<td>22</td>
<td>38</td>
<td>42</td>
<td>160</td>
</tr>
<tr>
<td>1</td>
<td>9</td>
<td>0</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>8</td>
<td>11</td>
<td>45</td>
</tr>
<tr>
<td>2</td>
<td>11</td>
<td>5</td>
<td>0</td>
<td>3</td>
<td>6</td>
<td>8</td>
<td>9</td>
<td>13</td>
<td>55</td>
</tr>
<tr>
<td>3</td>
<td>18</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>4</td>
<td>6</td>
<td>34</td>
<td>21</td>
<td>90</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>5</td>
<td>6</td>
<td>4</td>
<td>0</td>
<td>7</td>
<td>18</td>
<td>40</td>
<td>100</td>
</tr>
<tr>
<td>5</td>
<td>22</td>
<td>3</td>
<td>8</td>
<td>6</td>
<td>7</td>
<td>0</td>
<td>32</td>
<td>32</td>
<td>110</td>
</tr>
<tr>
<td>6</td>
<td>38</td>
<td>8</td>
<td>9</td>
<td>31</td>
<td>18</td>
<td>32</td>
<td>0</td>
<td>51</td>
<td>190</td>
</tr>
<tr>
<td>7</td>
<td>42</td>
<td>11</td>
<td>13</td>
<td>21</td>
<td>40</td>
<td>32</td>
<td>51</td>
<td>0</td>
<td>210</td>
</tr>
<tr>
<td>Total</td>
<td>160</td>
<td>45</td>
<td>55</td>
<td>90</td>
<td>100</td>
<td>110</td>
<td>190</td>
<td>210</td>
<td></td>
</tr>
</tbody>
</table>
series, the arrival time variance in each block was held constant (but not across blocks, where, again, it ranged between 4 to 28, and was assigned to blocks randomly), but the fraction of hub passengers varied between 5% and 35% (instead of a constant 20% as in the 7 + 1 problem and the variance-testing series). This series, to which we refer as the “hub passengers testing series,” was designed to investigate the third point. Both series were run at three levels of cost ($p = c = 12, p = 1.5 = 18$ and $p = 2c = 24$), and eight levels of separation constraints (0, 0.5, 1, 1.5, 2, 2.5, 4 and 6 min). The arrival time distribution for all runs was normal.

Each of the four runs in the variance-testing series, with separation constraint value of zero, provided seven blocks of seven flights in which the only difference was the variance. In each such block, the safety time proved to be a function of the standard deviation and the variance, yielding typical $R^2$ values between 0.9999 and 1.0000. The $t$ statistic for the variance was at least 4, and as high as 16. Therefore, one can conclude that the variance (i.e. the square of the standard deviation) was significant in all cases. Nevertheless, by dropping the variance from the regression, the results remained excellent, with typical $R^2$ values of 0.998 to 0.999, indicating that the effect of the variance on the service level may be highly significant but not necessarily important. This may explain the excellent results obtained with a heuristic that utilizes target service levels that are not functions of the arrival time variances. (The heuristic used the output of the unconstrained approximation as the basis for the surrogate functions.)

Each of the four runs in the hub passenger-testing series with separation constraint value of zero provided seven blocks of seven flights in which the only difference was the fraction of hub passengers. In each such block, the safety time proved to be a decreasing function of the hub passengers' fraction. Likewise, the departures in each such block were scheduled later for flights with more hub passengers, and this effect was weaker for high $p$ values. Thus, again, the insights from the small examples proved to be valid.

Note that once we introduce separation constraints to the model, the problem becomes combinatorial. Almost needless to say, for such large nonlinear and combinatorial problems it is not practical to search for the absolute optimum, and thus we cannot report on the performance of the heuristics relative to the absolute optimum. Our conjecture is that the global optimum is close enough to the local minimum as defined above. The evidence shows that such local minima can differ by up to 0.5%, depending on the starting sequence. We refer to these local minima, obtained by exhaustive searches that involved many target function evaluations, as our “benchmark values.”

Looking at the series of all 48 runs, although there was a statistically significant difference between the benchmark values of the variance-testing series and the hub passenger-fraction series, this difference was small in magnitude ($< 1.5\%$ in all cases, with an average difference of 0.06%). The similarity in target function values between the two series indicates that as long as the average hub passenger fraction and the average lead-time variance remain about the same, the target function does not differ by much. Nevertheless, in terms of the schedule itself, the benchmark solutions were very different indeed in the two series, because the variances of the flights were different. As for the effects of $p$ and the separation constraint value, the target function increases with each of these parameters in a concave way (diminishing increase). Table 3 demonstrates these points for separation constraints of up to 2.5 min. For each run, the value reported in Table 3 is the benchmark value.

The average CPU time required for each such benchmark value was 385 sec on an Amdahl 5990-500. These searches were so CPU-time-consuming for several reasons: (a) each iteration required a lengthy target function evaluation of approximately 0.5 sec CPU (due to the involved numeric integration); (b) the group structure was not utilized (because one of the objectives of the numerical experiments was to verify that using it was not detrimental); (c) many target function evaluations were required just to verify that a local minimum had been obtained, whereas most of the improvements over the starting solution were obtained much sooner. To counteract these problems, a heuristic procedure was devised such that (a) instead of calculating the target function at each iteration, surrogate target functions were used; (b) the group structure was used; (c) with
the surrogate functions taking negligible CPU time, the extra evaluations became a moot issue (otherwise, we could recommend not to verify convergence if the recent improvements were below a small threshold). Table 4 lists the target function values obtained by this procedure for separation constraints of up to 6 min. It also lists new benchmark values obtained by starting the benchmark search procedure where the heuristic stopped, the difference in percentages between the heuristic and the new benchmark and, where

Table 3. Benchmark target function values

<table>
<thead>
<tr>
<th>( p ) Value</th>
<th>Low</th>
<th>Medium</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Delta ) 0</td>
<td>192,401</td>
<td>191,691</td>
<td>204,180</td>
</tr>
<tr>
<td>0.5</td>
<td>240,783</td>
<td>240,380</td>
<td>252,532</td>
</tr>
<tr>
<td>1</td>
<td>291,672</td>
<td>289,330</td>
<td>303,164</td>
</tr>
<tr>
<td>1.5</td>
<td>354,468</td>
<td>352,011</td>
<td>361,239</td>
</tr>
<tr>
<td>2</td>
<td>434,987</td>
<td>434,497</td>
<td>441,322</td>
</tr>
<tr>
<td>2.5</td>
<td>519,505</td>
<td>523,217</td>
<td>528,531</td>
</tr>
</tbody>
</table>

\( p \) Value: low = $12$, medium = $18$, high = $24$ (all with \( c = 12 \)).

Design: \( H \) = hub passenger testing series runs.
\( V \) = Variance testing series runs.

Table 4. Results of the heuristic relative to the benchmarks

<table>
<thead>
<tr>
<th>( p ) Value</th>
<th>Low</th>
<th>Medium</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Delta ) 0</td>
<td>192,934</td>
<td>192,254</td>
<td>204,424</td>
</tr>
<tr>
<td>0.5</td>
<td>241,335</td>
<td>240,752</td>
<td>252,804</td>
</tr>
<tr>
<td>1</td>
<td>291,461</td>
<td>289,827</td>
<td>303,334</td>
</tr>
<tr>
<td>1.5</td>
<td>353,974</td>
<td>352,043</td>
<td>362,782</td>
</tr>
<tr>
<td>2</td>
<td>432,935</td>
<td>431,814</td>
<td>441,245</td>
</tr>
<tr>
<td>2.5</td>
<td>519,935</td>
<td>523,292</td>
<td>525,496</td>
</tr>
<tr>
<td>4</td>
<td>787,703</td>
<td>795,197</td>
<td>805,619</td>
</tr>
<tr>
<td>6</td>
<td>1,165,208</td>
<td>1,179,431</td>
<td>1,201,832</td>
</tr>
</tbody>
</table>

First entry — heuristic result.
Second entry — new benchmark (run after heuristic).
Third entry — difference in percentages (first minus second).
Fourth entry — heuristic over old benchmark in percentages.
appropriate, the difference in percentages between the heuristic and the old benchmark. Note that eight heuristic values as reported in Table 4 are actually better than their counterpart benchmarks reported in Table 3. These are the cases for which the fourth entry value is negative. All these cases occurred for the hub passengers-testing series and with separation constraints of at least 1.5 minutes. The new benchmark values were better than the former ones by about 0.115% on average, but some were inferior (the standard deviation of the difference was 0.174%). The CPU time consumption for the heuristic, including the generation of the approximate unconstrained solution, was approximately 2 sec per run. The CPU time consumption of the new benchmarks was, on average, 350 sec, that is, about 90% of the former benchmark CPU time consumption, but the variance was large, and in many instances it actually took longer.

By analyzing the figures in Table 4 we find that the heuristic is 0.06% above the benchmark, on average, with a standard deviation of 0.23 and a range of -0.57% to 0.37%. Note that one could not reject the hypothesis that the heuristic is as good as the benchmark. When the same analysis is carried out for the new benchmark values (which, recall, could not have been achieved without the heuristic to begin with), the difference is 0.136%, the standard deviation 0.12% and the range 0.004 to 0.42. These results exceed the wildest expectations of at least this author and compare very favorably with those of several other heuristics that were tested and rejected.

4. THE NO-WAITING CASE

So far we have assumed that all departures wait if necessary for any number of passengers who have been delayed. Some airlines assign such passengers to alternative flights, including flights offered by competitors or have them stay overnight at the airline's expense. Virtually every airline does so if waiting for a flight would disrupt the schedule beyond some acceptable level. Otherwise, the resulting delays might have a domino effect in other airports where passengers may have to make further connections. We consider a pure no-waiting policy, in which departures do not wait even for a single minute. In case of misconnections, there is a given penalty per passenger, denoted by $q$. This penalty represents the average additional cost involved plus the imputed value of the passenger's dissatisfaction. In the next two subsections, we present an approximate solution for this model and a more exact one.

The approximate solution we present is only valid when the level of service is high. The more exact model, in contrast, does not assume a high level of service, but it requires excessive computation time. Thus this section is not presented as a complete and satisfactory solution but rather as an introduction to a still open research question.

An approximate model

If flight $[o(j),0]$ is delayed, Flight $(0,j)$ may not be ready to depart to $j$ in time. Our approximation here is to ignore this possibility. By construction, this approximation is good when the level of service is high.

Delays in departures can still occur as per $F_0(0)$, however. Such delays may make it possible for some late arrivals to make their connections. To model the effect of delayed departures, let $H_{ij}(z)$ denote the probability that the passengers from Flight $(i,0)$ will make the connection to Flight $(0,j)$ if their scheduled arrival is $T_i - z$. If $j = d(i)$, then $H_{ij} = 1$; otherwise, given $F_o, F_i$ and $F_0$, we obtain

$$H_{ij}(z) = F_i(z) + \int_0^z (1 - F_0(x))F_j(z + x)dx; \quad \forall z \geq 0 \quad (4.1)$$

The first element of eqn (4.1), $F_i(z)$, is simply the probability that the connection will be made according to the schedule; that is, the probability $(i,0)$ will arrive by $z = T_i - S_i$. If $(i,0)$ is delayed by more than $z$, the connection can still be made if $(0,j)$ is also delayed. The integral sums the probability the connection will be made this way. In detail, the probability that $(0,j)$ will be delayed $x$ time units or more is $1 - F_j(x)$. The probability
Scheduling flights at hub airports 145

(0, i) will arrive during an infinitesimal period starting at x and ending at x + dx is 
f(z + x)dx. Thus there is a probability of [1 - F_0(x)]f(z + x)dx that a connection will be
made during that infinitesimal period. The integral sums these infinitesimal probabilities,
yielding the required late connection probability.

Recall that under our assumptions F_0 is identically distributed for all departures;
hence, H_{i,j}(T_j - S_i) is also identically distributed ∀j ≠ d(i).

Given H_{i,j}, we can obtain its density function, h_{i,j}, by differentiation:

\[ h_{i,j}(z) = \begin{cases} f_i(z) + \int_0^z \frac{\partial f_i(z + x)}{\partial z} [1 - F_0(x)] dx, & j \neq d(i) \\ 0, & j = d(i) \end{cases} \]

\[ (4.2) \]

Our objective function, Z, includes the value of the scheduled passengers’ time and
the expected number of misconnections multiplied by q. Because planes do not wait for
connecting passengers and we ignore the possibility that departures are delayed due to
late arrivals, there are no delays except those modeled by F_0(0). The penalty for these
delays is not influenced by our decision variables, so Z does not take them into account.
Also, the time value of passengers whose origin or final destination is 0 is not included,
because it is not influenced by our decision variables either. Thus

\[ Z = \sum_{i=1}^{n} \sum_{j=1}^{n} D_{i,j} \cdot c(T_j - S_i) \]
+ q[1 - H_{i,j}(T_j - S_i)] + \sum_{i=1}^{n} b_{i,j}T_j - \sum_{j=1}^{n} b_jS_j \]

\[ (4.3) \]

D_{i,j}c(T_j - S_i), when summed over all i and j, is the time value of all the passengers.
[1 - H_{i,j}(T_j - S_i)] is the probability that the passengers from i will fail to make the
connection to flight (0, j). Multiplied by D_{i,j}q and summed over i and j, it gives the
expected penalty. Note that the passengers from i to d(i) do not contribute to this part of
the objective function because 1 - H_{i,d(i)} = 0. Finally, the sum of b_{i,j}T_j over j, minus
the sum of b_jS_j over i captures the marginal time value of the aircraft and crews.

Taking the partial derivative of Z by S_i, we obtain

\[ \frac{\partial Z}{\partial S_i} = \sum_{j=1}^{n} \sum_{d(i)} \cdot D_{i,j} \cdot qh_{i,j}(T_j - S_i) - c \cdot b_j, \forall i \]

\[ (4.4) \]

Again, we assume that the T_j values are given. That is, the sequence of departures is
determined outside our model, and the gap between adjacent departures is G_p. To opti-
mize the scheduled arrivals, we need to set eqn (4.4) to zero, if possible, ∀i. Alternatively,
we could search for the minimum directly, because in this case it is as easy to evaluate Z
as it is to evaluate the derivative. We opt for the derivative because the same construct
can then serve us for more elaborate objective functions.

Our procedure is to look for the integer S_i values for which eqn (4.4) is as close to
zero as possible. Then, if the resulting arrival times violate G_A for some adjacent arrivals,
we can employ a grouping procedure, similarly to the way we did it for the waiting case.

Note, that for a group of flights whose unconstrained arrivals violate the separation
constraints, we can look at the sum of partial derivatives and drive that sum to zero,
instead of driving each of them to zero separately. Nevertheless, if we wish to change the
sequence, we cannot use the derivative to judge which sequence is better. In such a case
we have to compare the values of the target function directly. An open research question
is whether there is a good surrogate function that can help here, as in the waiting case.

A more exact model

The major assumption we made in the previous section was that all aircraft arrive in
time to support their own scheduled departures. This assumption, we noted, is not strong
when the level of service is high. In practice, especially when q is low, it is not necessarily
true that the level of service is high enough to justify this assumption. Therefore, we
drop it now.
Our first correction to the model is to replace $F_o(t)$ in eqn (4.1) with an expression that considers delays in the departure to $j$ due to late arrival of Flight $o(j),0$. Let $H_{o,j}(t)$ denote the probability that Flight $(o(j),0)$ will depart by $T_j + t$, then

$$H_{o,j}(t) = F_o(t - S_{o(j)}) \cdot F_o(t) + \int_0^t f_{o(j)}(T_j - S_{o(j)} + x)$$

$$\cdot F_o(t - x)dx \leq F_o(t) \cdot F_{o(j)}(T_j - S_{o(j)} + t) \quad (4.5)$$

The first element of $H_{o,j}(t)$ is the probability Flight $(o(j),0)$ arrives in time, and any further delays are at most $t$. If Flight $(o(j),0)$ arrives by $T_j + x$, however, and any further delays are at most $t - x$, Flight $(o(j),0)$ can still depart by $T_j + t$. The integral captures the probability of this event. Finally, to prove the inequality [upper bound on $H_{o,j}(t)$], substitute $F_o(t)$ for $F_o(t - x)$ in the integral. The inequality arises because the integral can only increase by the substitution.

When we substitute $H_{o,j}(t)$ for $F_o(t)$ in eqn (4.1), we obtain an expression that is no longer the same for any $j$:

$$h_{o,j}(z) = f_o(z) + \int_0^\infty [1 - H_{o,j}(x)]f_o(x + z)dx, \forall z \geq 0 \quad (4.6)$$

$h_{o,j}(z)$, the density function, may be obtained by differentiation, as in eqn (4.2). Once we allow the probability of delays in departures to become significant, however, we also have to consider the delay penalties that are incurred. When we do that, our target function and its partial derivative by $S_i$ are

$$Z = \sum_{r=1}^\infty \sum_{s=1}^\infty \sum_{\pi=1}^\infty \sum_{\sigma=1}^\infty D_{r,s}\{c(T_j - S_i) + q[1 - H_{o,j}(T_j - S_i)]\} + \sum_{r=1}^\infty \sum_{s=1}^\infty \sum_{\pi=1}^\infty \sum_{\sigma=1}^\infty h_{o,j}(z)T_j$$

$$- \sum_{r=1}^\infty \sum_{s=1}^\infty \sum_{\pi=1}^\infty \sum_{\sigma=1}^\infty P_{d(r)} \int_{r_{o(i)}}^\infty [1 - F_i(t - S_i)]dt \quad (4.7)$$

$$\frac{\partial Z}{\partial S_i} = \sum_{r=1}^\infty \sum_{s=1}^\infty \sum_{\pi=1}^\infty \sum_{\sigma=1}^\infty D_{r,s}[qh_{o,i}(T_j - S_i) - c] - b_i + P_{d(r)}[1 - F_i(T_{d(r)} - S_i)], \forall i \quad (4.8)$$

Equation (4.7) is really an approximation, too. To make it exact, we need to correct the value $P_{d(r)}$ in the last element to avoid assessing the penalty on passengers that fail to make the connection (for whom we already paid $q$). As it stands, this model may push for higher service levels than necessary. There is no conceptual difficulty in making this correction, but we omit the details.

Note that the derivative, eqn (4.8), does not involve integration, whereas the objective function, eqn (4.7), does. This is why it is easier to look for the values that set the derivatives to zero than to minimize the objective function directly. Nevertheless, the observation we made at the end of Subsection 4.1 still holds: if we need to compare different sequences we must use the objective function directly (or find a good surrogate function).

Our procedure to obtain the more exact solution is iterative. We start by solving the approximate model as in Subsection 4.1. Using the resulting $S_i$ values, we calculate $H_{o,j}$ and $h_{o,j}$, $\forall i,j$. Then we search for updated tentative values for $S_i$ that drive the partial derivative in eqn (4.8), or the sum of several such partial derivatives that belong to a group, as close to zero as possible. With the new tentative $S_i$ values we update $H_{o,j}$ and $h_{o,j}$, and so on. We stop when the tentative $S_i$'s become stable. (At this stage some very small examples, available from the author, converged to a stable solution in one iteration. We have no proof of general convergence, however.)

5. RELATED ISSUES

In this section we discuss the relationship between our models and other operational issues in the system. We also discuss how to combine the benefits of our two pure models in practice. Most of the results merit further research, and a thorough treatment is beyond the scope of this paper.
Scheduling flights at hub airports

The optimal airspeed as a function of expected lateness

Most delays in flights are due to problems at the origin or at the destination. At the origin, delays may be the result of weather conditions, queueing, problems in processing all boarding passengers in time (especially if stand-by passengers are involved), problems in preparing the plane for takeoff (fueling, loading, mechanical), and so forth. At the destination, delays may be caused by weather conditions or by queueing.

Assume now that a flight has departed behind schedule, and has a considerable distance to go toward the destination. Then it may be possible to recapture the delay, at least partly, by a higher speed. This will be justifiable if other aircraft are likely to have to wait for this flight, or if misconnections are expected. Thus the optimal airspeed of an aircraft that departed late is likely to be higher than otherwise.

If other flights are late, too, they may all have to increase their speed, but not necessarily by as much as each would have to on its own, because they now share the penalty. Also, in this case, it may be that even at maximal speed one flight will be so late that it does not make a great difference what speed the others will use—except for the passengers they carry to the hub itself.

In the hub-scheduling model, as discussed above, we did not take into consideration the ability of long flights to compensate for some delays by increased speed. As an approximation, one may say that the arrival time distribution of far origins can be adjusted to include this compensation. Still, incorporating such considerations in the scheduling problem optimally is an open research question.

Implications for the gate assignment problem

The existing literature on gate assignment strives to minimize the total walking distance for all passengers (Babic, Theodorovic, and Tosic, 1984; Mangoubi and Mathaisel, 1985). This objective does not consider the time allotted for making a connection. Nevertheless, if a flight is about to leave, it is more important that it be at a nearby gate than otherwise. This establishes a link between the gate assignment problem and the scheduling problem.

An obvious heuristic suggested by this consideration is to assign the latest arrivals and the earliest departures to central gates. This approach, however, ignores the traditional objective function (total walking distance).

An alternative approach is to "cheat" the regular gate assignment algorithms in such a manner that they will favor tight connections. Using our model output, we can do this by dividing $D_{ij}$ (a required input for the regular algorithms) by the probability of connection (the individual level of service as optimized by our models) raised to some power not less than one (e.g. squared). If the level of service is close to one, $D_{ij}$ will not be altered by much. When the level of service is low, $D_{ij}$ will be significantly inflated.

In our uncapacitated model, the level of service is likely to be roughly equal across all connections. Hence the gate assignment is not likely to be changed. In the capacitated models, however, some connections are forced to assume lower levels of service, whereas others are forced to wait longer. In this environment, the gate assignment is likely to be significantly different.

Finally, if we know that the tight connections are going to be made from favorable (central) gates, and assuming that walking time is the major consumer of connection time, we can reduce the connection time that we add to our model $T_j$ values. So, the gate assignment problem may influence the exact schedule, thus closing a cycle of influences between the gate assignment and the schedule.

Combining delay and misconnection penalties

The pure models described in this paper are designed to be used during the planning stage. The question whether a particular flight should be delayed to wait for late-connecting passengers is, however, a dynamic operational problem that needs to be addressed on a real-time basis.

Making the operational decision whether to delay an aircraft requires analyzing large amounts of data and justifies the use of computerized decision support systems (DSS).
The design of such DSS is beyond the scope of this paper. The fact remains, however, that some flights will wait and others will accept misconnections. This implies that neither of the pure schedules will be truly optimal. The true optimal schedule will probably be somewhere in between the two pure ones; for example, if the waiting model calls for a flight to arrive at 11:45 and the no-waiting one calls for 11:39, the flight should probably be scheduled sometime between 11:39 and 11:45.

Assigning aircraft to flight segments

As discussed in Section 4, the final destination of the $(i,0)$ aircraft $(V_i)$ is an important factor vis-à-vis the expected number of misconnections. It also impacts the gate assignment problem, because the $(i,d(i))$ passengers enjoy zero walking distance to their departure.

A possible approach to the problem is by minimizing the number of passengers who have to change aircraft. We can also do this subject to constraints on the allowed pairs. Such constraints may be used to promote good load leveling among the aircraft.

We assume all aircraft are of the same type and capabilities. In practice this is not likely. Nevertheless, the aircraft type is usually assigned to each segment in advance. Thus, we can solve the segment assignment problem for each aircraft type and the segments associated with it separately. Therefore this assumption is not restrictive.

We proceed by constructing an $n \times n$ matrix in which the element in the $i^{th}$ row and $j^{th}$ column is $\Sigma_{i,j}D_{i,j}$ (i.e. the average number of passengers who will have to change aircraft if we choose $d(i) = j$). Next, if certain pairs have a total distance that is too low or too high, we can prevent them by adding large penalties to their entries. Now we can use the Hungarian method to assign the destinations to the origins.

A major problem with this solution is that it may take up to $n$ days for an aircraft to return to its first origin. A practical policy may be to impose symmetry, that is, set $d(i) = iV_i$. We can optimize the assignment under this policy by comparing the $n(n - 1)/2$ combinations explicitly. This is possible only for an even $n$, however. If $n$ is odd we have to choose one flight to return to its origin or select three flights to rotate on a 3-day basis. The former choice will increase the number of alternatives to $n(n - 1)(n - 2)$, or $O(n^3)$. The latter choice yields $n!/(12(n - 5)!)$ alternatives, or $O(n^6)$.

Scheduling the durations of flights

So far, we assumed that the duration of each flight is predetermined. We also assumed that $F$ is given $\forall i$. Note now that if we allow more (or less) time without changing the speed choice, we can shift $F$ to the left (or to the right). This will increase (decrease) the probability of arrival for any given time. Because our models work in terms of the probability of connection, such changes in the assigned duration would just cause them to change $S$, in such a manner that the probability of connection would remain exactly the same. Actually, only the $D_{i,0}$ passengers are influenced by such a change—their time value increases to $p$ if the flight is delayed beyond $S$. Therefore, we have to optimize the nominal duration for these passengers only. This is achieved by setting the probability of delay times $p$ equal to the complementary probability times $c$ (as per the newsboy model). For instance, for $c = 12$ and $p = 24$, the probability of nominal delay should be set to $12/(12 + 24) = 33.33\%$. Thus the levels of service for hub passengers should be $66.67\%$. Correspondingly, a decision to set the level of service for hub passengers to, say, $90\%$ is equivalent to setting $p = 9c$.

6. CONCLUSION

We presented two basic models—with and without waiting—for the hub-scheduling problem. We also discussed how to combine the models and their connections to other operational issues.

Our exact solutions involve numerical integrals, which render them computationally intensive. Therefore, we also developed heuristics for the waiting case that do not involve integrals and take less than 1% of the CPU time to converge. Our numerical experience
suggests that the results are very close to the optimal solution, so much so that in several cases our heuristic produced better results than exhaustive searches. Moreover, it requires a small fraction of the computational effort. This argument is strengthened by the observation that (at least for our numerical examples) the objective function tends to be flat in the neighborhood of the optimal solution. Furthermore, because the truly optimal solution is different from those based on either pure policy assumption (whether to wait or not), one can argue that to find the exact optimum for a pure policy is not of paramount importance.

Our models compare favorably with the more conventional method of setting a single level of service objective for all flights. One advantage is that they allow the level of service to vary for different flights. Thus, they can achieve any given system level of service more efficiently. Furthermore, they set a less arbitrary system level of service.

The target-function level includes fixed elements and a variable element that is roughly proportional to the standard deviations of the arrival times. This implies that reducing these variances is conducive to lower costs (as is so often the case with variances). Because we identified the infinite capacity assumption as a cause of queueing in departures and arrivals, we can argue that by using the capacitated models (at the originating airports and at the hub) the variances can be reduced. This, in turn, reduces the cost of service and increases its quality. Thus, using the models may reduce the initial variances and achieve a double benefit.

We also demonstrated a disutility of scale in the hub and spoke scheme. We showed that increasing the number of cities served by a single hub increases the cost per passenger. This is true under the infinite capacity assumption and is exacerbated when we add congestion to the picture. On the other hand, increasing the number of cities served by a hub tends to consolidate passengers onto larger aircraft, which are more economical to operate. Thus a trade-off is involved. 1

REFERENCES


APPENDIX

In this Appendix we list some details of the procedures used to obtain the numerical results with separation constraints reported in the paper. It includes scheduling procedures for a given sequence that maintains the necessary separation constraints (using the original target function or the surrogate of matching the service level) and a pair-switching procedure. The latter also includes the optional surrogate function based on measuring a norm between the constrained solution and the unconstrained safety times.

The block scheduling procedure (for a given sequence)

0. (Initialize) Let $k(i)$ index the flights from $i$ in reversed order of scheduled arrival. We refer to $k$ as the ordinal index. Let $U_k$ be the unconstrained safety time of arrival $k$ (as determined by the uncapacitated model or the heuristic).

1This trade-off was pointed out by an anonymous referee. In general, the author is grateful to the referees for their thoughtful comments.

REFERENCES


APPENDIX

In this Appendix we list some details of the procedures used to obtain the numerical results with separation constraints reported in the paper. It includes scheduling procedures for a given sequence that maintains the necessary separation constraints (using the original target function or the surrogate of matching the service level) and a pair-switching procedure. The latter also includes the optional surrogate function based on measuring a norm between the constrained solution and the unconstrained safety times.

The block scheduling procedure (for a given sequence)

0. (Initialize) Let $k(i)$ index the flights from $i$ in reversed order of scheduled arrival. We refer to $k$ as the ordinal index. Let $U_k$ be the unconstrained safety time of arrival $k$ (as determined by the uncapacitated model or the heuristic).
The scheduling improvement procedure

Without changing the sequence, search for arrival times that improve the target function as much as possible. One way to do this is to treat each block as a unit that gets moved up and down together, until the best time frame is found for it. This search need not make use of the blocks, however, in which case we treat each flight as its own block. In either case, when trying to increase or decrease the safety time of one block, we may have to push other blocks along with it to prevent conflicts. (For the purpose of obtaining the benchmark solution reported in the paper, each flight was treated as its own block, i.e. the blocks were not used. For the service-level heuristic discussed below, however, the blocks were used.)

The pair switching improvement procedure

1. Two flights are said to form an adjacent pair if their schedules differ from each other by the exact value of the separation constraint; for example, any two consecutive flights in a block form an adjacent pair. Switch the flights in an adjacent pair to each other's schedule if it improves the target function value, and re-index the flights according to the new sequence. Stop when no improvements are possible for a full round.

2. If any pairs were switched in Step 1, return to the scheduling improvement procedure. Otherwise, stop; the current solution is a local optimum.

This procedure can be used either with the original target function, as was done to obtain the benchmark solution, or with the surrogate target function \( \sum_{i=1}^{m} \alpha |v_i - u_i| \), where \( \alpha \) is a positive number. By the numerical experience obtained for this paper, \( \alpha = 1.375 \) was excellent for separation constraints of up to 2.5 min, but for larger separation constraints the optimal \( \alpha \) value approached 1.00.

The service-level scheduling heuristic procedure

This procedure replaces the scheduling improvement procedure. It is designed to be used with blocks, that is, after the blocking procedure. The unconstrained solution values that are needed for this procedure are provided by the uncapacitated heuristic.

1. (Initialize) The target level of service for Block \( j \), \( L_j \), is obtained by multiplying the unconstrained approximation's levels of service of the flights in the block. The target cumulative level of service for Block \( j \), \( L_c \), is defined by multiplying the levels of service of all blocks up to and including \( j \).

2. (Compute blocks' levels of service) The level of service of Block \( j \), \( K_j \), is the produce of the levels of service of the flights included in the block for the current feasible solution. Given \( V_i \), compute the block level of service for each block. The cumulative level of service for Block \( m \), \( K_m \), is obtained by multiplying the levels of service of all blocks up to and including \( m \).

3. (Adjust blocks' level of service) Starting with Block 1 (i.e., \( m = 1 \)) and repeating successively for all blocks, look for the best feasible \( V_i \) values to minimize \( |L_j - K_j| \).