Demand Driven Dispatch: A Method for Dynamic Aircraft Capacity Assignment, Models and Algorithms

Matthew E. Berge, Craig A. Hopperstad,

DEMAND DRIVEN DISPATCH: A METHOD FOR DYNAMIC AIRCRAFT CAPACITY ASSIGNMENT, MODELS AND ALGORITHMS

MATTHEW E. BERGE and CRAIG A. HOPPERSTAD

The Boeing Company, Seattle, Washington

(Received September 1990; revision received July 1991; accepted June 1992)

A major problem for the airline industry is the assignment of airplane capacity to flight schedules to meet fluctuating market needs. Demand Driven Dispatch (D^3) is an operating concept that addresses this problem. Utilizing a demand forecast which improves as flight departure approaches, aircraft are dynamically assigned to flights to better match the predicted final demands. The result, demonstrated in studies of actual airline systems, is an increase in passenger loads and revenues with simultaneously reduced costs for a net of 1–5% improvement in operating profits. Concept implementation is simplified by the prevalence of yield management systems which provide the forecasting capability, and the emergence of airplane families which provide the necessary operational flexibility. Implementation also requires frequent solution of extremely large aircraft assignment problems. These problems, which can be formulated in terms of a multicommodity network flow, can be solved with heuristic algorithms shown to exhibit an accuracy and efficiency essential to successful concept implementation.

The stochastic nature of passenger demand contributes to a basic problem in airline operations—selecting an airplane fleet and assigning it to a set of flights. Because the variability of demand is high (typically, standard deviations of 20–50% of the mean demand) and because of the range of the *mean* demands experienced, even the best solutions give rise to average system loads in the region of 65% of capacity. The stochastic nature of the demands is such that even at these low operating efficiencies, significant passenger turnaway (spill) exists. A reduction in total capacity would magnify an already existing passenger spill problem, potentially resulting in a loss of market share. In response to these issues, much effort has been expended recently by airlines to fill empty sets via discount fares. However, even with sophisticated yield management systems, many seats remain empty.

In "The Penultimate Hub Airplane," Peterson (1986) proposed that the time was ripe to consider a fundamentally new airline operating concept. The concept exploits the fact that the ability to forecast final demand improves markedly as departure time approaches. While the ultimate *rubber* airplane could expand or contract to precisely match the final demand, the next best thing would be an aircraft *family* with sufficient operating flexibility to permit dynamic reassignment of the fleet to better match the updated forecasts. This gives rise to a kind of dispatch mode of operation, though not a pure dispatch scheme (e.g., taxi cabs) because the schedule must be flown no matter what the demand. Nevertheless, the proposed Demand Driven Dispatch (D^3) exhibits substantially more dynamism than the existing practice of assigning airplanes at schedule creation, typically 30–120 days in advance of execution.

From an implementation standpoint, Peterson argued that the necessary infrastructure was now (or would soon be) in place. This included:

1. an airplane family of two or more models with different seating capacity but with a common flight crew rating (i.e., an air crewman certified to fly one model is fully qualified to fly any model within the family);
2. computerized yield management systems, which require (in order to set fare class booking limits) sophisticated demand forecasting algorithms;
3. computerized reservation systems capable of accepting (even wholesale) changes to airplane assignments;
4. sufficient computing power and algorithm efficiency to solve giant aircraft assignment problems on a daily basis.

The requirement for a common flight crew-rated family constitutes a critical condition. At any point in time, most crew members are certified to fly only airplanes of a single cockpit design. Thus, if the airplane assignment is changed, for example, from a 727

Subject classifications Networks/graphs: heuristic multicommodity flow algorithm. Transportation, models: dynamic aircraft capacity assignment.
Area of review DISTRIBUTION, TRANSPORTATION AND LOGISTICS (SPECIAL ISSUE ON STOCHASTIC AND DYNAMIC MODELS IN TRANSPORTATION).

0030-364X/93/4101-0153 $01.25

to a DC-9, the flight crew assignment must also be changed. Given typical union contract and government regulatory rules, dynamic flight crew assignment is both difficult and expensive. Operating a common crew-rated fleet (e.g., the Boeing 737 family) eliminates these difficulties by enabling aircraft assignments to be made independently of crew scheduling.

The existence of sophisticated reservation and yield management mechanisms are also critical. All industry trends point to an ultimate world-wide presence of such systems. In North America, all major carriers possess this capability. In the rest of the world, the impetus of deregulation is driving even those carriers in near-monopoly positions to consider the concept.

The development of algorithms to solve the aircraft assignment problem constitutes the primary technical risk. In linear programming terms, a large domestic airline could generate a problem with as many as 100,000 variables and 50,000 constraints even without considering such operational issues as maintenance. Worse yet is the requirement that the decision variables be restricted to integer values. The combined large-scale *and* combinatorial nature of the problem, the required frequency of solution (daily), and the potential need for extensibility to address additional operational constraints all contributed to the decision to direct the research at heuristic methods. This has led to the discovery of extremely efficient algorithms for the solution of this problem.

In a survey of the general aircraft scheduling problem by Etschmaier and Mathaisel (1984), a concept of dynamic scheduling similar to Demand Driven Dispatch is discussed. They observed that while the concept may be viewed as an emerging new airline operating philosophy, no specific work in this area had been published. The purpose of our paper is to address the formulation of models and algorithms whereby the benefit of the concept and the feasibility of its implementation is assessed. The remainder of this paper is organized to reflect these objectives. Sections 1 and 2 describe the development and application of a proof-of-concept model for assessment of the basic utility of D^3. Sections 3 and 4 address concept implementation issues and describe algorithms for the solution of the large-scale assignment problems associated with implementation. Forecasting algorithms are excluded from the discussion because, in an implementation, they would be provided by the (required) yield management system.

The major findings of these investigations are twofold. First, the D^3 concept can be demonstrated to produce benefits in a range from 1–5% improvement in operating profits. Second, the proposed algorithms for solving the assignment problem exhibit both the accuracy and efficiency necessary for concept implementation.

1. MODEL

The task set forth for the D^3 model was to explore, in a proof-of-concept sense, the utility of the D^3 concept. Model development was driven by the need to answer these questions:

1. What is the payoff for swapping aircraft assignments?
2. Can the payoffs be forecast early enough?
3. Can aircraft assignments be juggled sufficiently to realize the payoff?

In the beginning we believed that the answers might be a strong function of the peculiarities of individual airline systems (e.g., big payoffs for hub-spoke type systems only). As a consequence, we decided that the D^3 model must reflect these peculiarities by incorporating actual airline schedules. The question of what degree of verisimilitude should be targeted in modeling other aspects of airline planning and operations remained. The assumptions selected almost always reflect relatively simple views of the phenomena. The adequacy of many of these assumptions has been addressed in the form of side studies. In some instances, alternative (boundary) representations were developed and incorporated in the model.

Figure 1 represents a conceptual flow diagram of the D^3 simulator. On a daily basis, bookings for future flights are accumulated, predictions are made relative to future demands, "decision" profits are estimated for alternative future aircraft assignments and then, based on these profits, aircraft are assigned to future flights. The primary measure of D^3 effectiveness is a

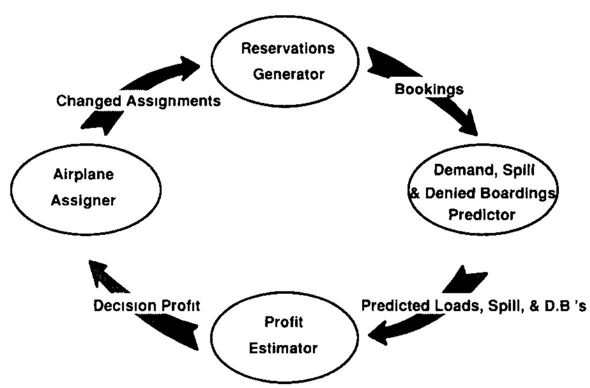

Figure 1. Conceptual flow.

comparison of profits with and without D^3. The method for computing the scores for the case without D^3 is identical to Figure 1 with the exception that the aircraft assignment remains fixed. This fixed aircraft assignment typically comes from an actual airline schedule.

The model's architecture and the two major elements of the model (reservation emulator and assigner) are discussed in the remainder of this section, with the focus on assumptions. The mathematics of the underlying assignment problem is reserved for Section 4.

1.1. Terminology

Below are the definitions for the terms used in this paper. In general, they are in accord with industry practice.

Flight leg: One city-to-city segment of a flight, i.e., a single takeoff and landing.

Planning point: A point in time prior to the departure of a flight when a set of decisions are made relative to the capacities specified in the reservation system (flight leg/fare class).

Planning horizon: The time from the initial planning point to flight departure.

Booking demand: Passengers wishing to obtain (book) reservations for a particular flight leg.

Demand factor: The ratio of booking demand to airplane capacity.

Booking probability: The cumulative probability that a random passenger attempts to book a reservation as a function of time until flight departure.

Spill: Passengers who were unable to obtain a reservation on a particular flight at a particular fare.

No-shows: Passengers who had reservations but did not appear for their flight.

Denied boardings: Passengers who appeared for their flight but, because of overbooking, could not be accommodated.

Loads: Passengers on-board a flight leg.

Load factor: The ratio of loads to airplane capacity weighted by flight leg distance.

Type: Type of airplane (e.g., 737-500).

Tail: Individual airplane.

1.2. Model Architecture

In the course of D^3 analysis two generations of simulators have been developed. In the first, time jumps from day-to-day along a theoretically infinite scroll, replicating the conceptual flow diagram shown in Figure 1. This initial version of the simulator thus presents a view of time which is nearly equivalent to an airline's operational perspective. However, due to a variety of model extensibility and program maintenance issues, a simplified (nonscrolling) representation of time was considered and incorporated in a second generation model. Specific timing and generic case study results described in this paper refer to this second model.

Two portrayals of the second generation D^3 simulator are provided. Figure 2 represents a macroflow diagram and Figure 3 the controlling time line. Both portrayals reflect multiple samples of the events surrounding a single week's worth of flights, referred to as the *index week.* In each sample, the simulation begins at a time sufficiently before the index week that no reservation requests have been received. Since the only information available at that time is historical, the only planning activity is that of setting fare class booking limits based on historical booking demand estimates. The simulator then jumps time from planning point to planning point, performing the following

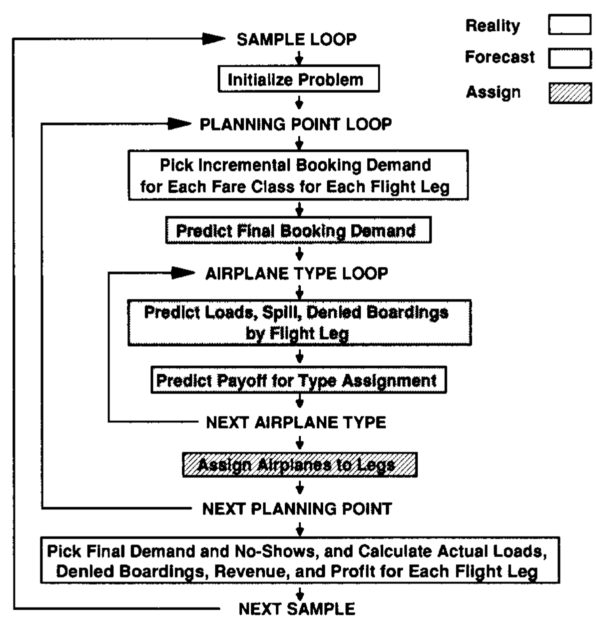

Figure 2. Macroflow.

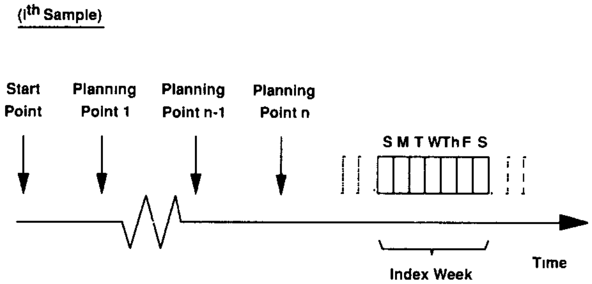

Figure 3. Time line.

functions at each point:

Reality: Compute what "really" happened in terms of bookings since the previous planning point for every flight leg scheduled to depart during the index week.

Forecast: Based on the bookings in-hand plus historical information (including booking probabilities), predict demand, and then predict the outcome at flight departure given each alternative aircraft assignment.

Assign: Based on the above forecast, aircraft are assigned to all flight legs associated with the index week and capacities are updated in the simulated reservation system.

Following the last planning point, time is jumped ahead to flight departure and the real outcomes are calculated, again, for all index week flight legs.

The most important architectural question is whether it is reasonable to examine a week in isolation and, in particular, what constraints should be placed on aircraft positioning at the beginning and at the end of this week. We decided to use the fixed aircraft assignment to constrain aircraft locations at the start of the week and no constraint at the end of the week because this approach is simultaneously D^3 pessimistic (day 1) and optimistic (day 7). The question of whether the pessimism and the optimism cancel each other was tested by comparison with the first generation model and yielded comparable results.

1.3. Reservation Emulator

The reservation emulator represents both the *realities* of revenue (bookings, loads, denied boardings) and the *planning* associated with predictions of revenue. In the course of the evolution of the D^3 simulator a variety of reservation system/passenger behavior issues have been explored with a variety of assumption sets. The assumptions identified below constitute the baseline set. A brief discussion of the side studies associated with alternative sets completes this section.

Baseline Assumptions (Reality)

1. The mean and standard deviation of booking demand is specified by flight leg and fare class in terms of normal distributions (truncated at zero) which are independent between flight legs and fare classes. The independence aspect is questionable. In the United States, about 40% of the passenger trips consist of more than one leg (each way). Nevertheless, much airline analysis of traffic focuses on data at the flight leg level.
2. All spill is out of the fleet for all fare classes. This is a doubtful assumption, particularly for passengers attempting to acquire reservations at deep discounts. The absence of spill recapture is, at least partially, compensated for by the absence of multileg spill. Passengers whose origin-to-destination path consists of several flight legs spill from a path if any leg is unavailable, and thus spill at higher rates than is predicted under the assumption of flight leg independence in 1 above.
3. Booked passengers never cancel reservations although they may or may not show up for the flight. No-show rates are the same for all flight legs and fare classes.

Baseline Assumptions (Planning)

4. The parameters of the basic stochastic processes (mean demands, booking probabilities, etc.) are known exactly by the forecaster. Although this is a somewhat optimistic assumption, it should be recognized that the current success of yield management systems depends, in part, on the fact that these parameters are known with some precision.
5. The forecasted outcomes for alternative aircraft assignments are made assuming the assignments to be final. Similarly, the yield manager selects a strategy assuming there will not be an opportunity to replan.

Critical Issues

Perhaps the most critical issue in the development of the reservation emulator involved the question of time correlation in bookings (do bookings in-hand predict bookings to come?). Because of the importance of the answer to the larger issue of D^3 efficacy, we decided that the model must have the capability to execute with either one of two boundary assumptions. Within each fare class, either:

1. bookings each day come from separate, independent populations (no correlation); or

2. bookings come from a single total population with a multinomial selection of booking day for each generated passenger (high correlation).

By assuming 1, the forecast of future demand is independent of current bookings and is thus D^3 pessimistic. Because the multinomial process in 2 generates bookings which are highly correlated over time, current bookings can be used to produce substantially improved forecasts over 1 and is thus D^3 optimistic.

A second critical issue involves the form of the yield management system. Perhaps the most significant research and development activity in the air transportation industry over the past ten years has involved definition of the mechanism for establishing booking limits by fare class. In the D^3 simulator the expected marginal seat revenue (EMSR) approach to yield management (see Belobaba 1989) was selected as a representative technique. The major characteristic of EMSR is fare class nesting. In a nested scheme the maximum number of reservations to be accepted is specified for all (discount) fare classes. If demand for a higher fare class intrudes on the available seats reserved for a lower class, then those seats are sold at the higher fare.

Side Studies

The primary side studies focused on alternative forms for the yield manager as well as certain simplifying assumptions:

- Inclusion of cancellations.
- Inclusion of differential (by fare class) no-show rates.
- Inclusion of vertical spill (sell-up) recapture. Vertical spill is defined to be the probabilistic process whereby passengers attempt to book at a higher fare for a flight leg given that they were unable to obtain a discount booking.
- Investigation of the possibility that the yield manager should employ a different strategy when he knows he will have future opportunities to intervene.
- Investigation of the flight leg orientation of the model. A series of simulations were developed which flowed passengers on the basis of origin-to-destination paths, but performed yield management by flight leg.

No side study findings refute the assessment of the relative merit assigned to D^3.

1.4. Assigner

The assigner addresses the problem of selecting assignments of aircraft to flight legs to maximize the sum of the decision profits. These decision profits describe the forecasted economic outcome for each flight leg/airplane type combination. The assignment must be feasible in the sense that it can be flown with the specified fleet. As discussed previously, there are also constraints on the positions of aircraft at the beginning of the index week, whereas their positions may be unrestricted at the end of the week. The major assumptions and associated issues are:

1. The timetable (schedule of arrivals and departures) is specified. For our D^3 studies, actual airline schedules are usually used along with an existing fixed assignment. We also assume that the fleet size is the minimum number of airplanes needed to fly the fixed schedule.

2. Any airplane type can be assigned to any flight leg. The relaxation of this assumption gives rise to "type constraints" which will be discussed in Section 4.5.

3. The flight-times and turn-times are the same for every airplane type in the D^3 fleet. The airplane turn-time is the minimum time from an airplane's arrival to its subsequent departure.

4. Flights are executed as scheduled. However, since the D^3 assignment process can be carried out well in advance (days) of flight execution, the impact of flight cancellations or delays would be no worse without D^3. The recovery process may be performed more efficiently with the additional flexibility and planning capability.

5. Throughout the assignment process only the question of type assignment is addressed. We assume that the determination of specific tail assignments (routings) are accomplished in a postprocessing step.

6. Maintenance constraints are not considered in the construction of the type assignments. Since maintenance is a tail-oriented requirement, this assumption implies that sufficient flexibility exists in postprocessing that tail routings can be constructed to adequately meet maintenance goals. A study, described in Section 4.5 and that addresses a specifically defined maintenance concept, has shown this to be the case.

A detailed formulation of the assignment problem is discussed in Section 4, along with algorithm descriptions, performance analysis, and side studies.

2. CASE STUDY RESULTS

The discussion of D^3 simulator studies and the resultant assessment of D^3 benefits is organized in two subsections: a report of general findings, and a relatively detailed description of a generic case.

2.1. General Findings

Since 1987, versions of the D^3 simulator have been used to investigate a variety of generic cases and more than 20 full-scale airline cases. Perhaps the most remarkable property of these studies has been their consistency of result—a 1–5% improvement in operating profit with the inclusion of D^3. The three questions posed for the D^3 simulator have been answered:

1. There is significant payoff for swapping aircraft assignments.
2. Forecasts are sufficiently accurate to predict the payoff in time to take action.
3. Airline schedules are such that most of the potential benefit associated with juggling airplane assignments can be realized.

In the course of performing these studies, a number of interesting and sometimes unexpected general conclusions were found. The most significant of these are reported in this section and address the nature (composition) of D^3 benefit, the criticality of assumptions relative to the forecasting module, and the relative importance of the airline route structure (e.g., hub-spoke versus linear).

D^3 Benefit Composition

Initially, it was thought that the only significant D^3 benefit would be an increase in revenue associated with producing additional passenger miles. This could be accomplished by swapping airplane assignments to increase total loads or to better serve the higher-fare, longer flight legs. As it turned out, this was only part of the payoff, as an additional benefit came from shifts in utilization from the big to the small airplane with its lower operating cost per flight leg. Results have shown a consistent pattern of D^3 benefit characterized by reduced spill, increased revenue, and decreased operating costs associated with utilization shifts of one to two hours per airplane per day.

Beyond the operational D^3 benefit lies the return associated with more effective fleet planning. Most studies indicate that 5% fewer seats with D^3 will produce as much revenue as the full complement of seats without D^3. A 5% reduction in the number of seats (accomplished by increasing the fraction of small airplanes in the fleet) results in a capital investment savings in the neighborhood of 3%.

Forecasting Assumption Criticality

In Section 1, alternative boundary assumptions were specified for the simulator forecasting module. The upper (optimistic) bound assumed a correlation between current and future bookings whereas the lower (pessimistic) bound did not. Comparisons of simulator runs, differing ony by this assumption, yielded the general assessment that the D^3 benefit is only 10–35% lower with the pessimistic view. This relatively minor impact is thought to be attributable to the fact that at the mid-point of the booking process, the forecasting errors are sufficiently small to (usually) enable the correct identification of the best airplane type. While this is not a complete solution to the forecasting problem, it constitutes a first-order capability.

Route Structure Importance

U.S. domestic (short-haul) systems are typically organized around a set of connect banks consisting of 10–40 closely-spaced flight arrivals followed, in a perfect bank, by an identical number of flight departures. These connect banks, designed to serve a large number (n^2) of origin-destination passenger markets, also give rise to a large number ($n!$) of potential swap opportunities. While the best D^3 benefits are obtained with these systems, the unexpected result is that substantial benefits are also obtained for nonhubbed, linear systems. The prevalence of swapping in situations where only two to four airplanes are simultaneously at an airport was far greater than was generally realized even by analysts associated with the study airlines.

2.2. Generic System Case Study

The generic airline system was patterned after a moderately sized domestic hub-spoke system. The flights operate through a hub with 10–12 airplanes converging at connect banks five times per day. A dozen other cities offer swap opportunities associated with 2–8 airplanes on the ground at the same time. The system, shown in Figure 4, operates 244 flights per day serving 22 airports with a fleet of 40 airplanes of three models from the Boeing 737 family, ranging in capacity from 108 to 148 seats.

Operating economics are representative of U.S. airline experience, and take the form of cost and revenue curves as a function of flight leg distance. For the cases reviewed here, passenger demand was generated in two fare classes, full-fare and discount fare (set at 65% of full fare). As a simplifying assumption a zero no-show rate is imposed. Three planning points were included in the analysis. The associated booking

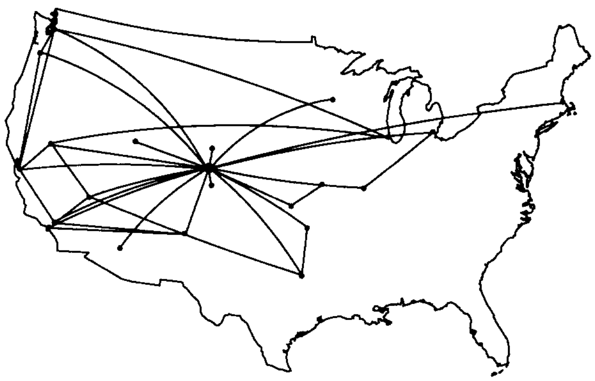

Figure 4. Generic airline system.

probabilities reflect the tendency for discount passengers to book prior to full-fare passengers.

Three cases are considered. The first is conservative, the second is optimistic, and the third addresses fleet downsizing for the optimistic case. The question of conservatism or optimism is relative to both the forecasting assumption and the specification of the variability of the flight leg demands. Demand variation manifests itself in the distribution of mean demands across flight legs, and in the distribution of demand about the mean for a single leg. In the latter case, the variability is typically specified as the *K*-factor, that is, the ratio of the standard deviation to the mean. These cases are described as follows.

Conservative Case: Each flight has a total mean demand specified to be *exactly* 65% of the capacity of the fixed assignment, with the mean demand divided evenly between the two fare classes. A *K*-factor of 0.3 is specified for each class giving rise to a joint *K*-factor of about 0.2. The pessimistic forecasting assumption is employed.

Optimistic Case: Each flight has a total mean demand uniformly generated in an interval from 50% to 80% of the capacity of the fixed assignment. This gives rise to an *average* system demand factor of 65%. Again, the mean demand is divided evenly between the two fare classes. A *K*-factor of 0.6 is specified for each class giving rise to a joint *K*-factor of about 0.4. The optimistic forecasting assumption is employed.

Fleet Downsizing Case: The downsized fleet shifted six units to the smallest airplane type for a total seat reduction of 4%. Total capital investment for the fleet was reduced by $40 million. Otherwise, the case definition is identical to the optimistic case, including the generation of mean demands based on the fixed assignments of the original fleet.

The conservative case represents a perfectly tuned fixed assignment because the assigned capacity is exactly matched to the mean flight leg demands. The only opportunity for improvement comes from satisfying the random variation about the means, also specified conservatively. As seen in Table I, there is no spill associated with the fixed assignment, thus, there is no opportunity for revenue improvement. Nevertheless, a significant D^3 improvement to operating profit is provided via a 2½ hour daily reduction in the use of the larger airplane. By contrast, in the optimistic case, the fixed assignment suffers significant spill due to the large variability in demand. Thus, the D^3 benefit is largely composed of increased revenue associated with the avoidance of more than ⅓ of this spill. At the same time, a 1 hour daily reduction in the use of the larger airplane also contributes to increased operating profits. Finally, in the fleet downsizing example, not only is the capital investment reduced, but the operating profit is improved by nearly 5%. This is accomplished, in almost equal parts, by increased revenue from spill avoidance and decreased operating costs from using more units of the smaller airplane type.

3. CONCEPT IMPLEMENTATION

During D^3 simulator development we learned that at least two airlines, KLM and Australian, regularly swap aircraft assignments in response to demand variation. At both airlines, identifying the aircraft assignments to be changed is a manual process supported by data from reservation and forecasting systems. Changes are not limited to airplanes within the same flight-crew family. At KLM, crew schedules are almost always changed to match airplane reassignments. Since this is difficult to accomplish close to departure, most airplane reassignments are made between 14 and 30 days out. At Australian, the flight crew and airplane assignments are set 6 to 8 weeks out with additional crews on staff to accommodate the most profitable swap opportunities close to departure.

While KLM and Australian provide a precedent for implementation of dynamic, demand-based aircraft assignment, at the time of this writing, no airline has implemented the concept as envisioned in this paper. That is, a full utilization of airplane commonality that involves the uncoupling of the airplane assignment and crew scheduling problems. Nevertheless, there has been considerable interest in the concept as witnessed by the number of studies that have been performed in cooperation with the airlines. In the course of these studies, it has been noted that the issues associated

Table I
Generic Airline Case Study Results

	Case		
	Conservative	Optimistic	Downsized Fleet
Fleet			
Number of Airplanes			
Smallest (735)	12	12	18
Mid-Size (733)	14	14	12
Largest (734)	14	14	10
Results			
Load Factor (Fixed Schedule)	65.1	64.2	64.2
Load Factor (With D^3)	66.7	65.8	67.9
Load Factor Change	1.6	1.6	3.7
Spill % (Fixed Schedule)	0.0	2.9	2.9
Spill % (With D^3)	0.0	1.7	2.0
Percent Change (Fixed to D^3)			
Passenger Revenue	0.0	1.1	0.9
Airplane Operating Cost	−0.7	−0.3	−1.4
Average Hours/Day Reduced Utilization, Largest Type	2.4	0.9	N/A
Operating Profit			
Percent Improvement	1.2	3.7	4.9
Added $ Per Year Per Airplane	50 K	147 K	194 K

with concept implementation differ from carrier-to-carrier as a reflection of different operating and marketing concepts. There has, however, been a consensus on the need to address the tractability of the large-scale assignment problem and the coordination of aircraft assignment with operational constraints such as maintenance. Since the major technical issues associated with implementation include algorithm performance and extensibility, these subjects will be revisited in Sections 4.4 and 4.5, following the discussion of the algorithms.

4. AIRCRAFT ASSIGNMENT PROBLEM

The assigner seeks to solve the problem of assigning airplane types to flight legs to maximize the sum of the decision profits. The development of algorithms for this purpose has been in support of both the simulator and D^3 implementation assessment. In either case, a high degree of optimality is required as D^3 benefit is measured relative to the fixed assignment, which is already quite efficient. Computational efficiency must be adequate to support parametric evaluations of D^3 effectiveness for proof-of-concept studies, and must enable frequent (probably daily) solutions of very large-scale problem instances for concept implementation. In the latter case, there may also be a requirement to address additional operational and maintenance constraints.

To further specify the problem, the flight legs under consideration are all legs departing within a given time period. At the beginning of this time period there is a constraint (starting condition) relative to the number of airplanes of each type positioned at each airport. In a D^3 implementation the time period is the planning horizon and the starting condition is determined from the solution obtained from the previous execution of the assigner. In the second generation D^3 model the time period is usually one week and the starting condition is determined by the fixed assignment. In any case, the assignment must be constrained so that it may be executed by the same number of airplanes required to fly the fixed assignment.

Note that this problem is a special case of the fleet assignment problem described in Abara (1989) and solved using LP and integer LP methods. The D^3 assumptions of identical flight-times and turn-times allow the problem to the formulated as a multicommodity network flow problem on a space-time (dynamic) network. The network structure can then be exploited by heuristic methods to produce excellent solutions at a computational cost far less than what would be required with LP methods.

4.1. Dynamic Network

In the subsequent network construction, a node represents an airport during a block of time. An arc either represents a flight leg or a period of ground time (i.e., a parked airplane). In the latter case the arc is called a null arc. The strategy is to construct *perfect nodes*, that is, nodes with the property that any airplane assigned to an arc into the node is eligible for assignment to any arc out of the node. As a result, a path through this network represents a feasible tail routing.

Consider the stream of arrivals and departures at a specified airport, and suppose that the arrival time also includes the airplane turn-time. Since the D^3 fleet exhibits identical flight-times and turn-times this stream can be constructed without knowing the airplane type assignments. Suppose that a sequence of M arrivals followed by N departures (followed by an arrival) is observed. Install a node associated with the specified airport with M arcs in and N arcs out. If M is greater than N, construct $M - N$ additional null arcs departing the node and arriving at the next node constructed for the same airport. Figure 5 illustrates an example arrival/departure stream with Figure 6 showing the corresponding node and null arc (dashed arc) construction. Figure 7 indicates the fragment of the network associated with the specified airport. Note that each node has an equal number of arcs into and out of the node.

The construction of the entire network can be accomplished by creating a system-wide arrival/departure stream and processing this stream from earliest to latest while simultaneously maintaining arrival/departure information at each airport. It may be useful to augment the network with a super source and/or a super sink node. A super source would enable the optimization to be carried out to allow the aircraft positions to be unrestricted at the beginning of the planning period, while a super sink corresponds to unrestricted aircraft positioning at the end of the period.

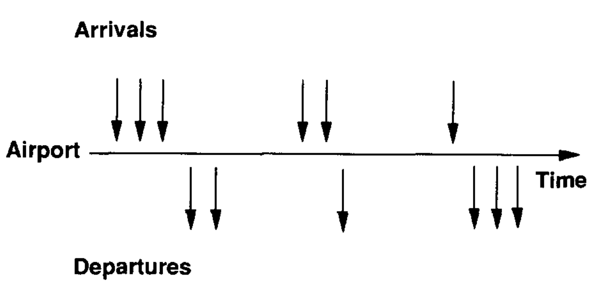

Figure 5. Arrival/departure stream.

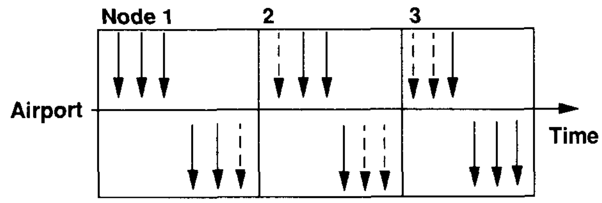

Figure 6. Node/null arc construction.

The algorithms for solving the aircraft assignment problem will take advantage of the special properties of the network. First, the network is acyclic. Consequently, we will assume that the nodes are topologically ordered (numbered) so that for any arc $e = (n1, n2)$, $n1$ is less than $n2$. Second, each arc will require assignment to exactly one airplane type. Third, when the supernodes are installed, then any other node in the network has the property that the number of arcs into the node equal the number of arcs out of the node. This number is referred to as the *node degree*. Note that the third property is implied by the second when a constraint of flow conservation by airplane type is imposed.

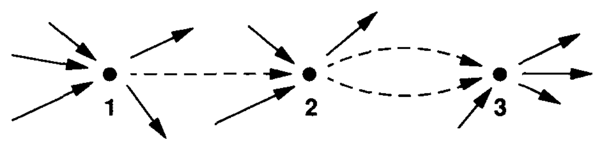

Figure 7. Network fragment.

4.2. Mathematical Formulation

The following formulations will make use of linear programming and network flow theory for both single and multiple commodities. For a discussion of these concepts, the reader is referred to Bazaraa, Jarvis and Sherali (1990). We consider the network described in Section 4.1 and define the following notation:

NNOD: the number of nodes in the network;
NARC: the number of arcs in the network;
NTYP: the number of aircraft types in the fleet;
F_t: the number of aircraft tails of type t in the fleet;
$\mathbf{X}_t$: the *NARC* length decision vector for type t. This vector identifies whether or not the type is assigned to the specified arc. For example, $\mathbf{X}_t(e) = 1$ indicates that type t is assigned to arc e, while $\mathbf{X}_t(e) = 0$ indicates it is not;

$\mathbf{B}_t$: the *NNOD* length supply vector for type t. This vector identifies the number of aircraft or type t that originate or terminate at a specified node. For example, $\mathbf{B}_t(n) = +2$ indicates 2 aircraft of type t originate at node n, while $\mathbf{B}_t(n) = -1$ indicates 1 aircraft of type t terminates at node n;

E: the *NNOD* × *NARC* incidence matrix for the network. For arc $e = (n1, n2)$ column e of the incidence matrix is defined to have an entry of +1 in row $n1$, −1 in row $n2$, and 0 elsewhere. With this definition, the conservation of flow constraint for type t may be expressed as $\mathbf{E} \cdot \mathbf{X}_t = \mathbf{B}_t$;

$\mathbf{P}_t$: the *NARC* length decision profit vector for type t. The decision profit vector entry $\mathbf{P}_t(e)$ represents the forecasted profits of assigning airplane type t to arc e. If arc e is a null arc, then the decision profit for assigning any type to e is zero.

Given this terminology, the problem of assigning aircraft types to arcs to maximize the sum of the decision profits can be written as:

$$\text{maximize} \sum_{t=1}^{NTYP} \mathbf{P}_t^T \cdot \mathbf{X}_t \tag{1a}$$

subject to

$$\mathbf{E} \cdot \mathbf{X}_t = \mathbf{B}_t \quad \text{for all } t \quad \text{(flow conservation)} \tag{1b}$$

$$\sum_{t=1}^{NTYP} \mathbf{X}_t = \mathbf{1} \quad \text{(assignment)} \tag{1c}$$

$$\mathbf{X}_t \geqslant \mathbf{0} \quad \text{for all } t \quad \text{(nonnegativity).} \tag{1d}$$

An integer solution to this linear program is a solution to the aircraft type assignment problem. The LP constraints for a 3-airplane type problem are shown in Figure 8 and consist of:

- *NTYP* · *NARC* variables;
- *NTYP* · *NNOD* flow conservation constraints;
- *NARC* assignment constraints.

The matrix has the special structure of block diagonality for the flow conservation constraints, and generalized upper bounding for the assignment constraints. More specifically, and as motivated by the preceding network description, it is a multicommodity network flow problem where the aircraft types represent the different commodities.

Despite that decomposition methods have been applied effectively to the general multicommodity flow problem, the possible nonintegrality of solutions and extremely large problem size represent major issues. Nonintegrality was occasionally observed in simplex solutions for problems consisting of more than two types. The large-scale nature of these problems is illustrated by a large carrier example across a 4-week planning horizon. This problem has an underlying network of about 6,500 nodes and 35,000 arcs, and for 3-airplane types yields a linear program consisting of 105,000 variables, 19,500 flow conservation constraints, and 35,000 generalized upper bound constraints. The computational effort to apply the established methods to such large-scale problems, combined with the possibility that the "long-awaited" solution could be fractional, played a major role in steering the initial investigations toward finding effective heuristics guaranteed to produce integral assignments.

$$\left[\begin{array}{c:c:c} \mathbf{E} & 0 & 0 \\ \hdashline 0 & \mathbf{E} & 0 \\ \hdashline 0 & 0 & \mathbf{E} \\ \hdashline \mathbf{I} & \mathbf{I} & \mathbf{I} \end{array}\right] \bullet \left[\begin{array}{c} \mathbf{X}_1 \\ \hdashline \mathbf{X}_2 \\ \hdashline \mathbf{X}_3 \end{array}\right] = \left[\begin{array}{c} \mathbf{B}_1 \\ \hdashline \mathbf{B}_2 \\ \hdashline \mathbf{B}_3 \\ \hdashline \mathbf{1} \end{array}\right]$$

Figure 8. LP constraint matrix.

4.3. Alternative Algorithms

This section describes two heuristic algorithms developed to solve the aircraft assignment problem. Both algorithms solve a sequence of subproblems using exact methods. The descriptions will emphasize the formulation and construction of the subproblems, but as these take the form of well known network optimization problems (e.g., minimum cost flow and acyclic longest path), their solutions will not be discussed in detail.

Sequential Minimum Cost Flow Method (SMCF)

The **SMCF** algorithm is based on the following fact:

> For an aircraft assignment problem with only two types, the problem can be reduced to a single commodity minimum cost flow problem. As a result, a variety of efficient algorithms can be applied to produce an exact solution.

To illustrate this, consider a two-type fleet and imagine an initial assignment of everything to type 1. Since this requires $F_1 + F_2$ aircraft, it is necessary to replace F_2 of the initially assigned type-1 aircraft by type-2 aircraft. The associated difference in profit to perform this replacement on an arc e is $\mathbf{P}_2(e) - \mathbf{P}_1(e)$. This replacement problem can be expressed as the following maximization version of the minimum cost flow (MCF) problem:

$$\text{maximize } (\mathbf{P}_2 - \mathbf{P}_1)^T \cdot \mathbf{X} \tag{2a}$$

subject to

$$\mathbf{E} \cdot \mathbf{X} = \mathbf{B}_2 \quad \text{(flow conservation)} \tag{2b}$$

$$\mathbf{X} \leqslant \mathbf{1} \quad \text{(arc capacity)} \tag{2c}$$

$$\mathbf{X} \geqslant \mathbf{0} \quad \text{(nonnegativity).} \tag{2d}$$

The implied solution to the original two-type multicommodity problem is $\mathbf{X}_2 = \mathbf{X}$ and $\mathbf{X}_1 = \mathbf{1} - \mathbf{X}_2$. Since these proposed solution vectors $\mathbf{X}_1$ and $\mathbf{X}_2$ are complementary, it is obvious that the assignment and nonnegativity constraints are satisfied in the multicommodity problem. Using this fact, it is also easy to see that the objective function for the multicommodity problem (1a) differs from the objective function for the **MCF** problem (2a) by a constant:

$$\begin{aligned}
&\mathbf{P}_1^T \cdot \mathbf{X}_1 + \mathbf{P}_2^T \cdot \mathbf{X}_2 \\
&\quad = \mathbf{P}_1^T \cdot \mathbf{X}_1 + \mathbf{P}_2^T \cdot \mathbf{X}_2 - \mathbf{P}_1^T \cdot \mathbf{X}_2 + \mathbf{P}_1^T \cdot \mathbf{X}_2 \\
&\quad = (\mathbf{P}_2 - \mathbf{P}_1)^T \cdot \mathbf{X}_2 + \mathbf{P}_1^T \cdot (\mathbf{X}_1 + \mathbf{X}_2) \\
&\quad = (\mathbf{P}_2 - \mathbf{P}_1)^T \cdot \mathbf{X}_2 + \mathbf{P}_1^T \cdot \mathbf{1}.
\end{aligned} \tag{3}$$

Finally, assuming the assignment of an aircraft to each arc is feasible, which may be expressed by $\mathbf{E} \cdot \mathbf{1} = \mathbf{E} \cdot (\mathbf{X}_1 + \mathbf{X}_2) = \mathbf{B}_1 + \mathbf{B}_2$, it is clear that $\mathbf{E} \cdot \mathbf{X}_2 = \mathbf{B}_2$ implies $\mathbf{E} \cdot \mathbf{X}_1 = \mathbf{B}_1$. Thus, all the constraints of the multicommodity problem are satisfied by $\mathbf{X}_1$ and $\mathbf{X}_2$. Since the objectives differ only by a constant, the two-type reduction to **MCF** is established.

The **SMCF** algorithm constructs and solves a sequence of ($NTYP - 1$) two-type assignment problems. Each problem is formulated and solved as a minimum cost flow, as described above. The steps of the algorithm are:

STEP 1. Index the airplane types in order of increasing capacity.

STEP 2. Construct the dynamic network as in Section 4.1.

STEP 3. Set the current type index to $t = 1$.

STEP 4. Solve the two-type problem as described earlier for $t2 = t$ and $t1$ representing the aggregation of all types with indices greater than t.

STEP 5. Remove from the network all arcs assigned to $t2$ in the solution from Step 4.

STEP 6. Set $t = t + 1$. If $t = NTYP$, then STOP, otherwise go to Step 4.

To clarify the algorithmic strategy consider a 3-type example. Suppose that an oracle, who knows the optimal solution for all three types, announces the $t = 1$ portion of the solution. Then the removal of arcs assigned to $t = 1$ leaves a 2-type problem on the residual network, which can be solved exactly to complete the solution. Thus, in the 3-type case, the critical step is to obtain a good estimate of the $t = 1$ part of the solution. The strategy of the algorithm is based on the notion that the decision of where to place type 1 (in general, type t) can be based on considering only alternative aircraft of the next largest capacity, that is, type 2 ($t + 1$). So the algorithm will determine the assignment of type 1 assuming all other assignments are type 2 when they must be a mixture of types 2 and 3.

We believe that this method is dependent on certain continuity or regularity conditions on the D^3 fleet. While we have not investigated this issue in depth, we suspect that these conditions are related to the fact that for a given flight leg, the decision profits may only exhibit certain orders. For example, consider airplane types (1, 2, 3) in increasing order of capacity. For a small predicted demand one would expect the decision profits to be in the order (1, 2, 3), while for a large demand the order (3, 2, 1) would be anticipated. In no case would orders (1, 3, 2) or (3, 1, 2) be expected. Experimentation has shown that interchanging operating cost functions between airplane family members sometimes degrades the effectiveness of the algorithm. In reality, such an irregularity in operating efficiency could not occur within a family inasmuch as families, almost by definition, share the same technology base.

The implementation of **SMCF** uses a successive shortest path method to solve the underlying minimum cost flow problems. The method exploits the fact that arcs have unit capacity. The shortest path algorithm is a label correcting method which utilizes the heuristic rule of placing nodes at the front of a double ended queue if they have been previously examined and at the back otherwise. This algorithm, described in Gallo and Pallottino (1986) and known as the D'Esopo-Pape algorithm, is found to be very effective for this problem despite the fact that its worst case complexity is not polynomially bounded.

Delta Profit Method (DELPRO)

The **DELPRO** method begins with a feasible assignment and performs multiple improvement steps. An improvement step involves finding nodes $n1 < n2$, airplane types $t1$ and $t2$, and paths $p1$ and $p2$, so that:

- path $p1$ is from node $n1$ to $n2$ along forward arcs assigned to type $t1$;
- path $p2$ is from node $n1$ to $n2$ along forward arcs assigned to type $t2$;
- swapping the assignments for $p1$ and $p2$ (i.e., change the assignment of $p1$ to $t2$ and $p2$ to $t1$) improves the assigned decision profit sum.

Figure 9 shows an example of such an "improvement cycle." If the solution prior to the path swap is feasible, then the cycle construction will guarantee feasibility after the swap. Given problem $(n1, t1, t2)$, the solution $(n2, p1, p2)$ can be found in the following way:

1. For the subnetwork associated with assignment $t1$, assign each arc e the delta-profit distance $\mathbf{P}_{t2}(e) - \mathbf{P}_{t1}(e)$. This quantifies the benefit of replacing the current assignment of $t1$ to arc e with new assignment $t2$. Find the longest path in this subnetwork along forward arcs from $n1$ to every node $n > n1$. Denote this distance function by DP(n, $n1$, $t1$, $t2$).
2. Similarly consider the subnetwork associated with assignment $t2$, compute $\mathbf{P}_{t1}(e) - \mathbf{P}_{t2}(e)$ for each arc e, and find the longest path from $n1$ to every node $n > n1$ to define DP(n, $n1$, $t2$, $t1$).
3. Select $n2$ as the $n > n1$ that maximizes

   DP(n, $n1$, $t1$, $t2$) + DP(n, $n1$, $t2$, $t1$).

For a given subproblem ($n1$, $t1$, $t2$), this procedure finds the best possible solution ($n2$, $p1$, $p2$).

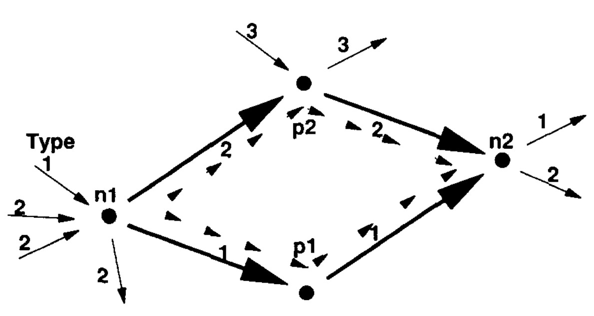

Figure 9. DELPRO improvement cycle.

If the quantity

DP($n2$, $n1$, $t1$, $t2$) + *DP*($n2$, $n1$, $t2$, $t1$)

is greater than zero, then the path swap will improve the score by exactly that amount and the swap should be performed. The overall algorithm proceeds by specifying a sequence of subproblems of this form. For a candidate node $n1$, each type pair ($t1$, $t2$) currently assigned into $n1$ gives rise to a subproblem ($n1$, $t1$, $t2$), which is solved exactly once. An effective processing order for the nodes has been found to be reverse topological order. For example, if node $n1$ has types $t1$, $t2$, and $t3$ into it, then problems ($n1$, $t1$, $t2$), ($n1$, $t1$, $t3$), and ($n1$, $t2$, $t3$) are each solved once and processing then proceeds to node $n1 - 1$. Processing every node in the network defines a "pass" of the algorithm which may then be repeated. A single pass of **DELPRO** may be summarized as:

STEP 1. Set node counter $n1 = NNOD - 1$.

STEP 2. For each distinct ($t1$, $t2$) into node $n1$ perform a and b:

a. Formulate and solve problem ($n1$, $t1$, $t2$) once to produce ($n2$, $p1$, $p2$) and DELTA = DP($n2$, $n1$, $t1$, $t2$) + DP($n2$, $n1$, $t2$, $t1$).

b. If DELTA > 0, then swap assignments of paths $p1$ and $p2$.

STEP 3. Set $n1 = n1 - 1$. If $n1 = 0$ STOP, otherwise go to Step 2.

The longest path calculations can be performed extremely efficiently because the network is acyclic. The implementation uses the method of reaching described in Denardo (1982), in which the shortest (or longest) path from $n1$ to each node $n > n1$ can be found by examining the arcs out of every node n exactly once. The overall efficiency is further improved by combining reaching with a look-ahead strategy to constrain the search for $n2$. This method produces the best $n2$ in the range of $n1$ to $n1 + NNLK$, where $NNLK$ = the number of nodes in the look-ahead. The result is that in the course of executing a single pass of the algorithm each node is scanned at most $NNLK$ times. Thus, for a fixed number of airplane types, the number of computational steps in a pass is bounded above by a constant times $NARC \cdot NNLK$.

It turns out that values of $NNLK$ can usually be selected to greatly reduce the computation time for a negligible reduction in the score. This selection is essentially independent of the number of weeks in the planning horizon. Since $NARC$ is a linear function of planning horizon weeks, the above discussion suggests a linear relationship between runtime and planning horizon length.

As shown for algorithm **SMCF**, the problem for two types can be solved exactly as a minimum cost flow problem. It turns out that this is not the case for **DELPRO**. From network flow theory, a proposed feasible solution to a minimum cost flow problem is only optimal if it does not admit a flow augmenting cycle which improves the score. However, a flow augmenting cycle for an **MCF** is more general than the improvement cycle defined above. With this as a guide, it is not difficult to construct two-type cases which do not admit an improvement cycle even though a better assignment exists. Experience with **DELPRO** for D^3 assignment problems has indicated that this shortcoming is not a major problem. The definition of the improvement cycle as above is meant to allow the exploitation of the acyclic network in the longest path calculation. Studies to date based on real airline flight schedules indicate that the associated degradation in the score is negligible.

4.4. Algorithm Performance

The objective of this section is to show that both **DELPRO** and **SMCF** possess the accuracy and efficiency necessary to support concept implementation. Program storage issues are not discussed because computing platforms appropriate to implementation would easily accommodate the algorithms. The D^3 simulator is used to randomly produce problem instances for solution by the optimizers. The analysis is performed on a VAX-8600 computer under VAX/VMS V5.3-1 with the simulation and assignment algorithms implemented and compiled under VAX FORTRAN V5.2. As the interest here is algorithm performance, a number of simplifying assumptions are made. These assumptions do not have an effect on the realism of the network topologies, but mainly affect the reservation emulator, and, consequently, the resultant decision profits. Our case study experience has been that the performance indicated in this section holds across a broad spectrum of reservation emulator assumptions.

Each assignment problem is constructed from the flight legs associated with N consecutive days of operation. These flights are derived from the subfleets of three different U.S. domestic carriers with schedules extracted from the Official Airline Guides. The operating patterns of all three are typical of domestic hub-spoke operations. The cases shown in Table II are named to differentiate the carriers and to identify the number of airplane types employed. The dynamic networks are constructed as in Section 4.1 with the exception that reduction techniques are used to combine flight legs where possible to reduce the network size. Runtimes are measured in average number of CPU seconds per sample, and do not include time spent constructing the network or performing reservation emulation functions. The major reservation system simplifications involve the use of a single fare class, a single planning point, and a "perfect forecast." The latter is employed so that the decision profits are equal to the actual profit outcomes, thus providing maximum sensitivity of D^3 effectiveness to the performance of the optimizers.

Table II
Case Descriptions

Case	Number of Airplanes #735	#733	#734	Total	Flight Legs Per Week
A-2	30	12	0	42	1,966
A-3	24	12	6	42	1,966
B-2	23	27	0	50	2,212
B-3	19	23	8	50	2,212
C-2	104	107	0	211	8,310
C-3	104	36	71	211	8,310

As discussed in Section 4.3, the **DELPRO** algorithm may be controlled by the number of passes to be executed and a specification of a look-ahead value. Analysis has shown a relatively small payoff for additional passes. **DELPRO** performance is shown for two passes and it should be noted that with one pass the runtime could be approximately halved for a small reduction in the score. An important finding has been that runtimes can be reduced significantly at a negligible reduction in the score, through the use of a constrained look-ahead. Reasonable look-ahead values were determined to be 100 nodes for A and B, and 300 nodes for carrier C.

Tables III and IV summarize the results of the performance analysis. In each case, multiple samples of a problem are generated and solved using each of

Table III
LP/Heuristic Optimality Comparison

Case	Samples	Days	Nodes-Arcs	Ratio to Best Result **DELPRO**	**SMCF**	**LP**
A-3	10	1	109–287	0.9992	1.0000	1.0000
A-3	5	3	183–653	0.9996	0.9999	1.0000
B-3	10	1	115–352	0.9999	1.0000	1.0000
B-3	5	3	197–784	0.9999	1.0000	1.0000

Table IV
DELPRO/SMCF Optimality and Runtime Comparison

Case	Samples	Days	Nodes-Arcs	Ratio to Best Result (CPU-Sec.) **DELPRO**	**SMCF**
A-2	50	7	325– 1,319	0.9999 (1.32)	1.0000 (0.61)
B-2	50	7	361– 1,648	1.0000 (1.56)	1.0000 (1.52)
C-2	5	7	1,806– 9,055	0.9999 (38.1)	1.0000 (61.1)
C-3	5	7	1,806– 9,055	0.9999 (39.3)	1.0000 (62.8)
C-3	2	14	3,424–17,577	1.0000 (79.8)	1.0000 (123.2)
C-3	1	28	6,660–34,621	0.9999 (161.1)	1.0000 (221.3)

the candidate algorithms. The average scores are normalized by the average best result produced (which is optimal except in the lower part of Table IV). The number of samples were selected based on the practical considerations of total runtime as opposed to statistical convergence.

The analysis is divided into three separate studies. The results of Table III are oriented toward comparing the optimality of the results of the heuristics with the LP solutions (which happened to always produce integer solutions in this analysis). Only small problem instances could be solved by LP due to excessive runtimes. The study shows that the accuracy of the algorithms are in excess of 99.9% of optimal for **DELPRO** and in excess of 99.99% of optimal for **SMCF**. For comparison, the fixed assignment in the first A-3 case is about 97.6% of optimal.

The top set of results in Table IV compares **DELPRO** to **SMCF** for medium-sized (one week's worth of flight legs) problems consisting of two airplane types. Since **SMCF** is guaranteed to be exact in such cases, this provides a measure of the optimality of **DELPRO**. The study shows that this accuracy is in excess of 99.99% of optimal.

The last set of results (the bottom portion of Table IV) focuses on examining the runtimes for **DELPRO** and **SMCF** for problems of increasing size (planning horizon length). Since LP could not be executed for these problems due to excessive runtime, and since they consist of three types, it was not possible to compute the exact optimums. The best result was virtually always returned by **SMCF**. In terms of runtimes, both algorithms exhibit an approximately linear relationship between runtime and planning horizon length. While this was predicted for **DELPRO** in Section 4.3, it is strictly an empirical observation for **SMCF**. Finally, even the largest problem instance (28 days of C-3), with an LP matrix consisting of over 100,000 columns and over 50,000 rows, could be solved in less than 4 minutes with **SMCF** and less than 3 minutes with **DELPRO**.

4.5. Extensibility Issues and Side Studies

The last section shows that algorithm performance is more than adequate to support a D^3 concept implementation. However, the formulation of the assignment problem may be complicated by the need to relax certain of the operational assumptions described in Section 1.4. This section summarizes the side studies conducted to address these issues. The results have supported both the feasibility of D^3 concept implementation and demonstrated the robustness and extensibility of the aircraft assignment methodologies. An additional benefit was the discovery that the assignment methods could be modified to address the fixed assignment and fleet mix optimization problems. These problems are important elements of the airline fleet planning process even without D^3.

Type Constraints

One of the operational assumptions discussed in Section 1.4 was that any airplane type could be assigned to any flight leg. A type constraint is a relaxation of this assumption in which various airplane type/flight leg combinations are prohibited. An example of this situation occurs when the range capabilities between airplane types in the D^3 fleet differ sufficiently to create a situation in which one airplane type may be assigned to a flight leg while, due to insufficient range capability, another type cannot. An obvious approach is to address the problem simultaneously with the assignment problem by introducing negative decision profits of sufficient magnitude to deter the forbidden assignments. This strategy is very effective when the **DELPRO** algorithm is used. If the starting assignment is feasible then it stays feasible, otherwise **DELPRO** finds swaps which reduce the number of infeasibilities.

Thru-Flights

In a D^3 operation there is a potential loss of thru-flight service. In this regard, the assignment problem can be constructed in such a way which would respect thru-flights for selected flight leg combinations. This can be accomplished in the network construction process by concatenating the flight legs that require the same airplane assignment into the same arc. While this method can be used to measure the D^3 benefit lost by keeping thru-flights, it is also necessary to measure the airline market share impact of eliminating thru-flights. In general, a tradeoff of D^3 benefit versus marketing impact is required to identify what thru-flight service to retain. In any case, the overall importance of thru-flights has declined with the advent of hub-spoke operations.

Maintenance Constraints

In the absence of D^3, maintenance planning consists of determining tail routings compliant with the schedule and fixed assignments which meet the specified maintenance constraints. Scheduled maintenance events are denoted as *A*, *B*, *C*, or *D* "checks" in increasing levels of attentiveness and in decreasing required frequency. The *A* check, of greatest frequency, poses the most difficult problem for the maintenance planner. Though based on aircraft flight hours, some airlines have imposed a 4-day maximum time span between inspections. In a D^3 operation, meeting these constraints is complicated by the fact that the aircraft type assignments are continually changing. This forces a repeated solution of the tail routing problem to meet these constraints with no a priori guarantee that a solution even exists.

To address this issue, a side study was performed to measure the likelihood of being able to meet the 4-day constraint under dynamically changing D^3 type assignments. The analysis assumed that the tail routings could be generated independently for each airplane type, and that within each type the airplanes would be routed according to strict 4-day rotations. It was also assumed that the type assignments would be fixed at least four days prior to departure. The problem was to determine, given routings for the tails scheduled for maintenance in the next 3 days, routings for tails needing maintenance on the fourth day. A simulation and a tail routing algorithm were developed and utilized to randomly generate and solve problems of this form.

A study was performed for the system of carrier *C* (see Section 4.4), and we assumed the same distribution of maintenance facilities employed by that carrier without D^3. Preliminary results suggest a high probability of finding a feasible tail routing even if the maintenance interval is two or three days instead of four. This conclusion supports the view of treating the assignment problem as a type problem versus a tail problem. Additional background and analysis on the airline maintenance problem may be found in Feo and Bard (1989).

5. SUMMARY

Demand Driven Dispatch offers a way to exploit the technologies of forecasting and optimization in a fundamentally new airline operating concept. If implemented, it would have significant industry impact in terms of both airline profits and operating philosophy. The findings presented in this paper are the result of studies performed with airlines over the last few years. Some airlines, impressed by the demonstrated D^3 benefit, are ready to cooperatively pursue much more detailed evaluations for the purpose of deciding the implementation question. We anticipate that the operational issues will become a bigger part of these evaluations and that the effectiveness and extensibility of alternative assignment algorithms will play an important role.

ACKNOWLEDGMENT

The authors acknowledge the many contributors in the development of the D^3 concept and simulation. Robert Peterson conceived of and initiated the D^3 project at Boeing. Also of particular note is Don Kitch who undertook the difficult job of integrating the first generation version of the simulation. Other contributors to the first generation model include Jim Hinkhouse, Charles Soncrant, and Myles Winbigler. Also, thanks to the reviewers for their many helpful suggestions and, in particular, to Virginia Sweetland, current project leader, and Nick Walker for their excellent reviews of the draft materials. Finally, the authors wish to acknowledge the senior management of the Boeing Commercial Airplane Group for sustaining a long-term commitment to the research and development of the D^3 operating concept.

REFERENCES

ABARA, J. 1989. Applying Integer Linear Programming to the Fleet Assignment Problem. *Interfaces* **19,** 20–28.

BAZARAA, M. S., J. J. JARVIS AND H. D. SHERALI. 1990. *Linear Programming and Network Flows.* John Wiley, New York.

BELOBABA, P. P. 1989. Application of a Probabilistic Decision Model to Airline Seat Inventory Control. *Opns. Res.* **37,** 183–197.

DENARDO, E. V. 1982. *Dynamic Programming: Models and Applications.* Prentice-Hall, Englewood Cliffs, N.J.

ETSCHMAIER, M. M., AND D. F. X. MATHAISEL. 1984. Aircraft Scheduling, The State of the Art. In *Proceedings of the AGIFORS Symposium,* 181–225.

FEO, T. A., AND J. F. BARD. 1989. Flight Scheduling and Maintenance Base Planning. *Mgmt Sci.* **35,** 1415–1432.

GALLO, G., AND S. PALLOTTINO. 1986. Shortest Path Methods: A Unifying Approach. *Math Prog. Study* **26,** 38–64.

PETERSON, R. M. 1986. The Penultimate Hub Airplane. Internal Memo, Boeing Commercial Airplane Group, Seattle, Wash.