Transportation Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

The Four-Day Aircraft Maintenance Routing Problem

Kalyan T. Talluri,

The Four-Day Aircraft Maintenance Routing Problem

KALYAN T. TALLURI

Universitat Pompeu Fabra, 27 Ramon Trias Fargas, 08005 Barcelona, Spain.

Federal aviation regulations require that all aircraft undergo maintenance after flying a certain number of hours. Most major U.S. airlines observe the maintenance regulations by requiring that aircraft spend a night at a maintenance station after at most three or four days of flying. In addition, some airlines require that every aircraft goes through a special maintenance station for what is commonly called a balance check. Airlines usually schedule routine maintenance only at night so as not to cut into aircraft utilization. The maintenance routing problem is to find a routing of the aircraft that satisfies the short-term routine maintenance requirements. In Gopalan, R. and Talluri, K. T. ("The Aircraft Maintenance Routing Problem," Opns. Res. *in press) we modeled this problem as one of generating an appropriate directed graph (called a line-of-flight-graph), and of finding a special Euler Tour called the k-day Maintenance Euler Tour (k-MET, for $k = 3, 4, \ldots$) in that directed graph—for finding a maintenance routing in which the aircraft would spend at most k days of flying before overnighting at a maintenance station and have an opportunity for a balance-check. In the same paper we gave a polynomial-time algorithm for finding a 3-MET, if one exists, in the directed graph. In this paper we consider the routing problem when the requirement is to overnight at a maintenance station after at most four days of flying and to undergo the balance check every n days, where n is the number of planes in the fleet of the equipment type under consideration. We show that this problem is NP-complete; in fact, that the k-MET problem is NP-complete for all $k \geqslant 4$. When the number of maintenance stations is exactly one, we show that the 4-MET problem can be solved by solving an appropriate bipartite matching problem; and hence in polynomial time. As a corollary to this result, we show that when there is no balance check station visit requirement, the four-day routing problem, in a given LOF-graph, can be solved (without any restrictions on the number of maintenance stations) in polynomial time. We show how our polynomial-time algorithms for the 3-MET problem and the restricted 4-MET problem can be used to design effective heuristics for the 4-MET problem.*

An airline's schedule, for our purposes, is a list of flight legs specifying the origin and the destination of each flight, the departure and arrival time of the flight, and the days of the week that the flight is scheduled to operate. Table I shows the relevant fields of a partial airline schedule.

The airline has to assign equipment types such as a Boeing 757, Fokker F100, etc., to each of the flight legs in the schedule so that every leg is assigned an aircraft, balance and aircraft count constraints are satisfied, and the sum of the operational costs and the opportunity cost of losing passengers on a flight due to insufficient seat capacity is minimized. It would imply more revenue for the airline, for example, if it assigns a Boeing 757, which has a capacity of 170 passengers in coach, to high-demand routes, and a Fokker 100, which has a capacity of 98 in coach, to low-demand routes.

In recent research, a number of researchers have used mathematical programming models to solve this assignment problem (JARRAH, 1993; DILLON et al., 1993; HANE et al., 1995; SUBRAMANIAM et al., 1994), and have reported considerable progress in solving it to near optimality. All these models assume that the same schedule is repeated every day over a period of time, and that the same equipment types are assigned to the legs of the schedule for every day of this period. These models have come to

Transportation Science
Vol. 32, No. 1, February 1998

0041-1655/98/3201-0043 $01.25

TABLE I
Relevant Fields of a Partial Airline Schedule

Flt. No.	From	To	Dep.	Arr.	Frequency
547	BOS	PIT	525p	711p	12345
1753	BOS	PHL	730p	851p	1234567
...	...	...	...	...	...

be known as the *daily fleet assignment* models. The assumption that the same schedule is repeated every day of the week, although not true on weekends in general, is a reasonable modeling assumption because most of the major U.S. airlines run the same schedule for the weekdays and a reduced schedule for the weekends. The primary motivation for the model is to capture the weekday situation, when the capacity considerations come into play. A schedule for which a fleet type is assigned to each one of the flights (assuming daily fleet assignment) will be called a fleeted schedule and the assignment itself will be referred to simply as a fleeting.

A planning problem that arises after a schedule has been fleeted is finding a routing of the aircraft to satisfy maintenance requirements. Note that a fleet assignment assigns an *equipment type* (757, MD93, F100, etc.) to a flight leg, but does not specify which one of the physical aircraft in the fleet (called *tail* or *nose* numbers in the industry) flies the leg on any given day. For planning purposes, it is important that the tail numbers can be routed so that they are able to overnight at maintenance stations at regular intervals (for example, after at most three days of flying or at most four days of flying), where they can undergo FAA specified required maintenance. This is to ensure that during operations, every aircraft will have the opportunity to undergo routine maintenance. In GOPALAN and TALLURI (1996) we have modeled this planning problem with assumptions similar to the daily fleet assignment model, and showed how to solve the problem efficiently when the maintenance requires that every aircraft overnight at a maintenance station after at most three days of flying.

In this paper, we consider the four-day version of the maintenance routing problem. The FAA regulations for frequency of visits to maintenance stations varies by equipment type. For some equipment types, airlines translate the FAA regulations to requiring visits to a maintenance station after at most three days of flying, and for others after at most four days of flying. For newer aircraft types, for example, a four-day visit requirement is usually sufficient to meet FAA guidelines, even with a safety margin built in for schedule disruptions. Most U.S. airlines have a mixture of three-day and four-day visit requirements (FEO and BARD, 1989; KABBANI and PATTY, 1992).

Broadly speaking, our solution strategy for the routing problem is to fix aircraft routings during the day, and to determine if such a set of over-the-day routings (we will refer to these as lines-of-flight or LOFs) admits an aircraft routing that satisfies all the overnight maintenance requirements—and if it did not, change the LOFs to increase the chances of it doing so. It is for this latter routing problem that we obtain our complexity results and polynomial-time algorithms. For the part about fixing LOFs, we rely on heuristic procedures, whose goal is to satisfy the necessary conditions that we obtain for the latter routing problem.

This paper makes the following contributions:

1. For a given set of LOFs, we determine the complexity of the following problem: Find a routing that satisfies the four-day (in fact, k-day for all $k \geqslant 4$) and the balance check visit requirement that is NP-complete. Note that in Gopalan and Talluri (1996a, b) we showed that the three-day problem without red-eye flights (flights that originate on the U.S. West Coast at night and arrive on the East Coast the next morning) in the schedule and the three-day problem with at most one red-eye originating maintenance station can be solved in polynomial time.
2. We show that without the balance check visit requirement, even the four-day problem can be solved in polynomial time.
3. We show how the three-day algorithm of Gopalan and Talluri (1996a) and the polynomial time algorithm of (2) can be used to design effective heuristics for the four-day problem.

The heuristic procedures for generating the LOFs are very similar to the ones in Gopalan and Talluri (1996a), and therefore, we will ignore that part of our routing solution strategy, and concentrate instead on routing given a set of LOFs.

The rest of the paper is organized as follows: In Section 1, we state the maintenance requirements, describe the model, our solution strategy, and give a brief survey of the work done on the problem. In Section 2 we prove the complexity results. In Section 3 we show that the four-day routing problem without the balance check visit requirement can be reduced to a bipartite matching problem, and hence can be solved in polynomial time. We also outline how this result applies to a dynamic routing model considered in Gopalan and Talluri (1996a). In Section 4, we describe three heuristics for the four-day problem that utilize polynomial time algorithms for restricted versions of the problem. Finally, in Section 5 we make some concluding remarks.

1. THE FOUR-DAY AIRCRAFT MAINTENANCE ROUTING MODEL

IN THIS SECTION we describe the maintenance requirements, our model and solution strategy for generating a maintenance routing, and give a brief overview of previous work on the problem. Since the contents of this section are covered in detail in Gopalan and Talluri (1996a), we keep our descriptions brief. Throughout this paper we use the words station and airport interchangeably. A maintenance station is an airport where the airline has a maintenance base. The maintenance base is designated by equipment type.

1.1 Lines of Flying

During a typical day, an airplane (tail number) starts early in the morning, visits a series of airports, and finally overnights at some station. Because maintenance can only be performed at night, we consider the day's activity in terms of *lines of flying* (LOFs) that specify the originating station at the start of the day and the destination station at the end of the day for a particular airplane. Thus, for instance, if a plane originates in Washington, D.C., flies to Pittsburgh, flies back to Washington, D.C., and then to Boston to spend the night there, the information is summarized as a LOF from Washington, D.C. to Boston. LOFs are also referred to as *over-the-day routings* in the literature (Kabbani and Patty, 1992).

A set of LOFs can be constructed from a fleeting by using simple rules such as first-in-first-out or last-in-first-out. However, note that the LOFs so constructed may not satisfy the maintenance requirements.

1.2 Maintenance Requirements

The Federal Aviation Administration (FAA) requires several types of aircraft maintenance. The checks are called A, B, C, and D and vary in their scope, duration, and frequency. Of these, only type A checks have to be performed frequently (every 65 flight hours) and are considered routine maintenance. Type A checks involve a visual inspection of all the major systems. If the check is not performed within the specified period, FAA rules prohibit the aircraft from flying. B checks are of longer duration, 10 to 15 hours, and C and D checks take a few days. Different airlines handle these longer duration checks differently. A few airlines break the C check into four quarter-C checks (we will call this a *balance-check,* so that aircraft spend shorter time each time but have to visit the balance check station more frequently.

Industry maintenance practices are much more stringent than FAA rules require them to be. It is a common practice in the industry to allow at most 35 to 40 hours of flying before the plane undergoes what is called a *transit check.* This check involves a visual inspection and a check to see if the plane carries what is called a Minimum Equipment List. This check is usually done whenever a plane visits a maintenance station, regardless of how recently it was last performed. Note that a visit to a maintenance station represents a maintenance opportunity and does not necessarily mean that maintenance *has* to be performed on that aircraft.

We will assume exactly one maintenance station per equipment type where this balance check can be done. Some airlines may have more than one balance check station per fleet type, but typically the number is small.

In order to meet the requirement for transit checks, we require that every tail number spend a night at a maintenance station after at most four days of flying since its last overnight visit to a maintenance station. Note that this requirement is more generous than asking for a routing that has every aircraft visit a maintenance station *exactly* once every four days. We also assume that for every equipment type the station where balance checks for that equipment type can be done is also a regular maintenance station. We route all the aircraft of one equipment type at a time.

Transit checks take a relatively short time to perform. We therefore assume that there is no real capacity constraint for doing the transit check maintenance. However, as balance checks require a considerable number of maintenance personnel, there is a capacity restriction of only one balance check per night. Fortunately, typically, we have considerable leeway in scheduling frequency of visits to the balance check station. It is usually enough if we can ensure in the routing that every tail number passes through the balance check station at some point. For this reason, we will assume that if there are n aircraft in the fleet of a particular equipment type, our requirement is that every one of the tail numbers of that equipment type should visit the balance check station once every n days.

From now on, the term *four-day maintenance routing* will denote a routing of tail numbers (of the equipment type under consideration), such that all the tail numbers visit a maintenance station after at most four days of flying without maintenance and every tail number spends at least one night at the balance check station for that equipment type, at which time it can undergo the balance check. Note that this definition can be extended to define a k-day maintenance routing for any integer k.

Finally, we would like to emphasize that maintenance requirements differ from airline to airline. Some airlines do not have balance checks altogether, while some airlines have more than one balance check station and have different requirements for frequency of balance checks. However, the requirement for transit checks is, as far as we know, very widespread among all the major airlines. The contribution of this paper is primarily to show that the four-day periodic visit problem can be solved very easily as a bipartite matching problem, and then to give heuristics for a balance-check requirement under some assumptions. If an airline has different balance check requirements, similar heuristics can potentially be used to incorporate them.

1.3 The Model

In our routing model, the LOFs are assumed to remain the same, day after day, over an infinite time horizon. We must route every aircraft through a maintenance station once every four days or less and at least once through the balance check station. Note that we are implicitly assuming that the schedule and the fleeting also remain the same every day. This assumption is also made in the popular daily fleet assignment model. Our model for maintenance routing is designed to work with the daily fleet assignment model with an additional assumption on the LOFs. Our assumptions are fairly reasonable when the schedule and fleeting are static Monday through Friday, and planners are concerned mainly with weekdays. This is true for the domestic schedules of most of the major U.S. airlines.

We leave aside for the moment the question of how to generate the LOFs and assume instead that we are given a set of LOFs and our problem is to determine if there is a routing satisfying maintenance requirements with this set of LOFs.

The overnighting stations and LOFs can be represented by a directed graph G = (V, E), where the vertices V represent the set of stations where the aircraft overnight and the arcs E represent the LOFs. We will call G an LOF graph. The number of arcs will be equal to n, the number of aircraft in the fleet of the equipment type under consideration. We partition the set of nodes V into a set of nodes M, representing the maintenance stations, and N = V—, representing the nonmaintenance stations. We will refer to nodes in M as M nodes, or maintenance nodes, and nodes in N as N nodes, or nonmaintenance nodes.

In our model, a four-day maintenance routing is an Euler tour (see Bondy and Murty (1978) for basic graph theory definitions) in G that includes at most three nodes of N in succession. Let $(i_1, i_2, \ldots, i_n, i_1)$ be the sequence of nodes in such an Euler tour, with

$$(i_1, i_2), (i_2, i_3), \ldots, (i_{n-1}, i_n), (i_n, i_1),$$

the arcs of the Euler tour. Note that nodes can be repeated in this sequence. We interpret the Euler tour in terms of maintenance routing as follows: At the beginning of day 1, position the n planes of the equipment type in the fleet at the nodes $i_1, i_2, \ldots, i_n$. Then, the plane at node i_k goes to station i_{k+1}, following the Euler tour, i.e., using the LOF represented by the arc (i_k, i_{k+1}). Because the Euler tour passes through a maintenance node after at most three consecutive nonmaintenance nodes, the tail numbers overnight at a maintenance station after at most four days of flying without maintenance. In addition, because a Euler tour is connected by definition, every tail number passes through the balance check station once every n days. As we mentioned before, the balance check station is also a regular maintenance station. Hence, if a tail number visits the balance check station more than once in the n days, it is to undergo regular maintenance the other times.

We call a Euler tour in the LOF graph that meets the four-day requirement, a *Four-Day Maintenance Euler Tour* (4-MET). Clearly, a necessary condition for G to have a 4-MET is for it to be Eulerian, i.e., for it be connected and the indegree is equal to the outdegree at each node. We will assume that the LOF graph generated is always connected and because the number of aircraft of any equipment type originating from a station in the morning is equal to the number of aircraft of that equipment type terminating at that station at night, our graph G is Eulerian.

Our solution strategy for generating a four-day maintenance routing is the same as the one for the three-day problem considered in Gopalan and Talluri (1996a). For that problem we generated the LOFs to satisfy as many of the necessary conditions as possible for the LOF graph to have a 3-MET, and then use the 3-MET algorithm to determine if a 3-MET exists. If it did not have a 3-MET, we changed the LOF graph to improve the chances of it having a 3-MET. The situation for the four-day case is a bit more complicated because determining if there is a 4-MET in the LOF-graph turns out to be an NP-complete problem (cf. Section 2). However, the same solution strategy works well with the heuristics that we develop later for finding a 4-MET. A necessary condition for G to have a 4-MET is for it to have a partition into circuits, with each one of the circuits containing a 4-MET. Call such a decomposition of G into circuits, a *Four Day Maintenance Circuit Decomposition* (4-MCD). In Section 3, we will give a polynomial time algorithm for finding a 4-MCD in G.

Our heuristics are based either on using the polynomial time 3-MET algorithm of Gopalan and Talluri (1996a) or a polynomial time algorithm for finding a 4-MCD (cf. Section 3). Therefore, we have a set of well defined necessary conditions to guide our LOF graph generation. Inasmuch as the LOF graph generation procedures are the same as in Gopalan and Talluri (1996a), we refer the reader to that paper for details about that part of our routing procedure.

1.4 Previous Work

Kabbani and Patty (1992) and Feo and Bard (1989) model the maintenance routing problem as a set-partitioning problem for determining LOFs and use heuristics to form maintenance routings. Feo and Bard also combine the routing problem with the problem of locating maintenance bases. Jarrah and Yu (1990) give an integer programming formulation for the maintenance routing problem. In Clarke et al. (1996), the routing problem is modeled as a Traveling Salesman Problem with side constraints and solved using Lagrangian relaxation and heuristics. Clarke et al. also model the benefits of designating "through" (or direct service) flights. Our model differs from those proposed in the literature, but the underlying objectives and requirements are similar.

2. NP-COMPLETENESS OF THE FOUR-DAY ROUTING PROBLEM

In this section, we show that finding a 4-MET in the LOF graph G is NP-complete.

Theorem 2.1 *Given a connected Eulerian directed graph (indegree equals outdegree at each node)* G = (V, E), *and a subset* M ⊆ V, *the following problem is NP-complete: Does* G *contain a 4-MET.*

Proof. The reduction is from the directed Hamiltonian circuit problem which is well known to be NP-complete (Garey and Johnson, 1997): Given a directed graph, does it contain a Hamiltonian circuit, i.e., a tour that visits every node of the graph.

The 4-MET problem is clearly in NP. Given an instance of the Hamiltonian circuit problem G′ = (V′, E′), we will construct a graph G = (V, E) such

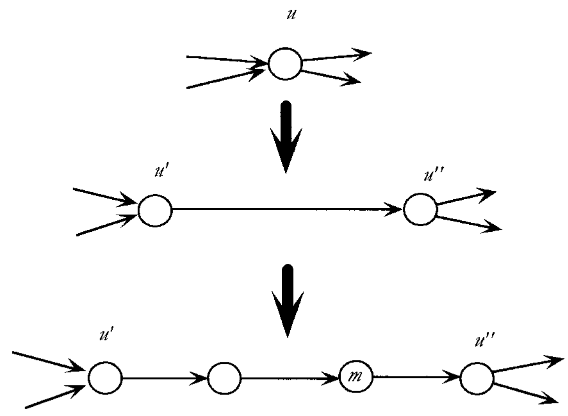

Fig. 1. Subdividing a node arc.

that there is a tour in G′ if and only if G has a 4-MET.

First, we will construct a graph G″ = (V″, E″) from G′, by replacing every node $u \in$ V′ by two nodes u' and u'' and an arc (u', u''). All arcs that were coming into u in G′, now come into u' in G″ and all the arcs that were leaving u in G′, now leave from u'' in G″. Now replace the arc (u', u'') by a path of three new arcs and two new nodes as shown in Figure 1. The third node on this path will belong to M (labeled m in Figure 1). The graph so constructed is G″. A tour in G′ clearly corresponds to a tour in G″. So in fact, we will assume that our input graph is in the form G″.

We will construct a graph G from G″. Add a new node t to G″, $t \in$ M (see Figure 2). Add arcs from t to the nodes u' and u'', for all $u \in$ V′ as follows: if u' has $k_{u'}$ arcs coming into it, add $k_{u'} - 1$ new arcs from u' to t (call them x-arcs), and if u'' has $k_{u''}$ leaving from it, add ${}_{u''} - 1$ paths of length two from t to u'' (call the second of the two arcs on a path, y-arc; see Figure 2). In addition, add two paths of length two each to node 1′, one going from t to node 1′ and the other from node 1′ to t, again, as shown in Figure 2. Call the resulting graph G.

We will show that G has a 4-MET if and only if G″ has a tour.

G has a 4-MET if G″ *contains a Hamiltonian tour*

Every arc in the Hamiltonian tour of G″ corresponds to an arc in G. Construct the 4-MET in G as follows: Start the Euler tour at t. Every y-arc will belong to a loop that starts from t, uses an arc in G″ that does *not* belong to the Hamiltonian tour of G″, and comes back to t via an x-arc. Thus, we are back at t and we have used all the arcs of G save the ones in the Hamiltonian tour and the two paths between t and 1′ through nodes a and b. Now, all these remaining arcs of G″ will form part of the 4-MET by

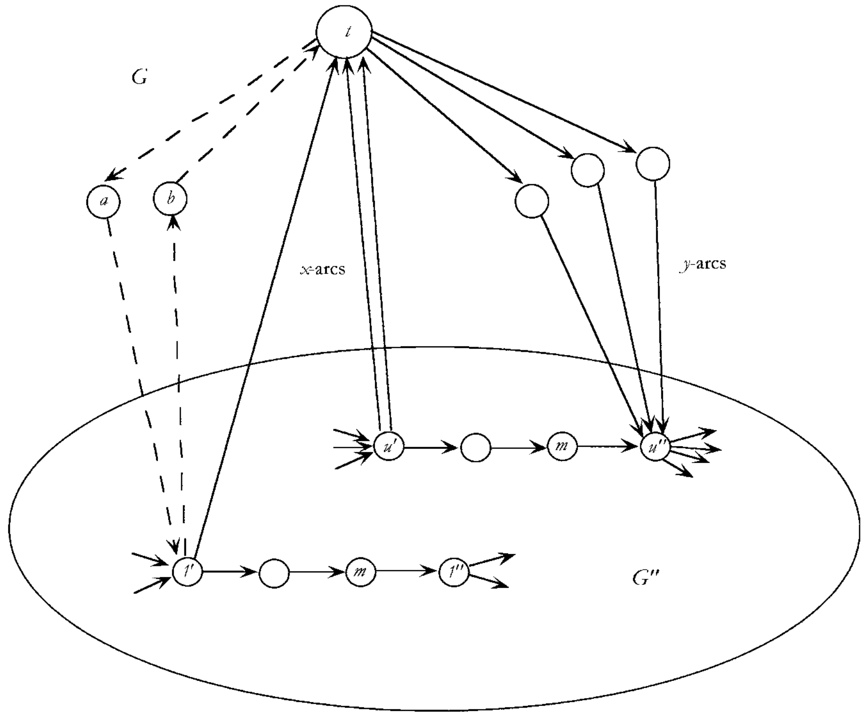

Fig. 2. The construction used to prove that the 4-MET problem is NP-complete.

a Euler tour that goes from t to a to node $1'$, and then goes along arcs of the Hamiltonian tour of G″ to come back to node $1'$ and then back to node t, via node b. Thus, we have constructed a 4-MET, because, from the construction of the graph G″, this Euler tour satisfies the four-day requirement.

G *has a 4-MET only if* G″ *contains a Hamiltonian tour*

Assume, without loss of generality, that G has a 4-MET starting from t. We will construct a Hamiltonian tour of G″. In the 4-MET, the y-arcs can only be part of loops that start at t, use an arc from G″ and come back to node t via an x-arc. Otherwise, we will be violating the four-day requirement. Therefore, once we remove all these loops, because the number of y-arcs at each node u'' is one less than the number of arcs leaving u'', and the number of x-arcs at each node u' is one less than the number of arcs entering u', we will be left with a graph that has exactly one arc coming in and one arc going out at each node in G, except for node $1'$, which, in addition, is connected to t via nodes a and b. Because there is a 4-MET in G, after removing the y-arc loops, we should obtain a connected graph. Because every node in G, after removing the y-arc loops, has one arc coming in and one arc leaving (except for $1'$, of course), the remaining graph has to be a Hamiltonian tour of G″. Therefore G″ has a Hamiltonian tour.

Therefore, determining if a directed Eulerian graph has a 4-MET is NP-complete. □

If, in Figure 1, we replace the path of length two from the left-most node to the M node m, by a path of length $k - 2$ for $k \geqslant 5$, we would have shown that the k-MET problem is NP-complete, for $k \geqslant 5$.

THEOREM 2.2. *Determining if* G *contains a k-MET for* $k \geqslant 5$ *is NP-complete.*

3. POLYNOMIAL-TIME ALGORITHM FOR FINDING A 4-MCD

IN THIS SECTION, we give a polynomial time algorithm for finding a 4-MCD in an LOF-graph.

First, we show that finding a 4-MET in G can be solved in polynomial time if $|M| = 1$. As a corollary to this, we show that the four-day maintenance routing problem without the balance check station visit requirement can be solved in polynomial time. We also sketch how this result extends to a dynamic finite-horizon model that was considered in Gopalan and Talluri (1996a). In this dynamic model, the LOFs can vary from day to day, and the routing problem is to find a routing of the aircraft over a finite horizon that satisfies the four-day visit to maintenance sta-

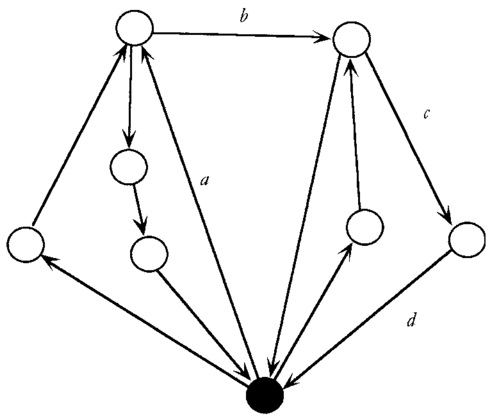

Fig. 3. Removing an arbitrary loop in an LOF graph with one M node does not lead to a 4-MET.

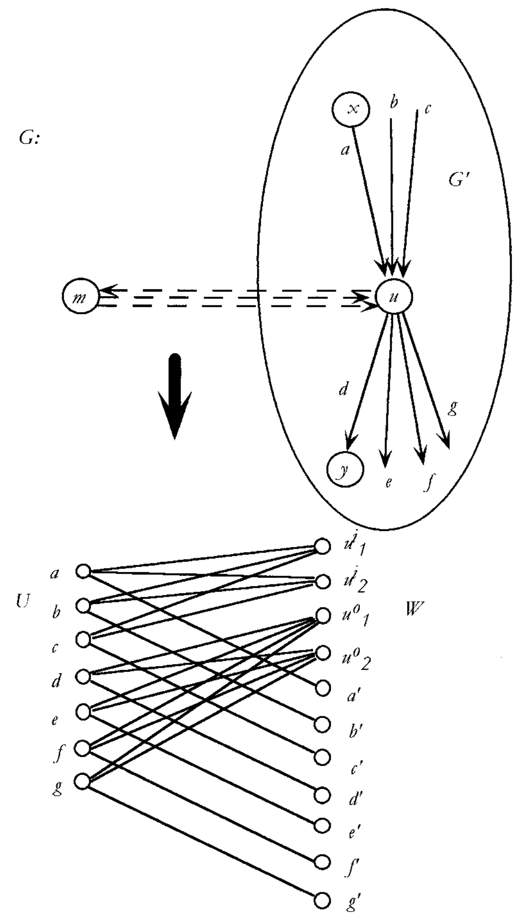

Fig. 4. The construction of the bipartite graph H.

tions requirement. Figure 3 shows that the problem of finding a 4-MET in a LOF graph with just one M node is nontrivial, in the sense that, we cannot remove arbitrary loops of length at most four starting at the M node one at a time, and hope to find a feasible solution—removing the $a-b-c-d$ loop does not lead to a 4-MET, even though the LOF graph has a 4-MET.

Consider the 4-MET problem where the LOF graph G = (V, E), has exactly one maintenance station m. We reduce the 4-MET problem to a bipartite matching problem. Let G′ = (V′, E′) be the graph obtained from G by deleting node m and all the arcs incident to it. Construct a bipartite graph H with bipartition U and W as follows: Every arc a of G′ will have a node a in U and a node a' in W, as shown in Figure 4. H will have edges of the form (a, a') for every arc a in G′. In addition W will have the following nodes: For a node u of G′, let m_u^i be the number of arcs going from node m to u in G and let o_u be the outdegree of u in G′. Let $k_u := \max(0, o_u - m_u^i)$. Then, W will contain nodes u_j^i and u_j^o for $j = 1, \ldots, k_u$. For $j = 1, \ldots, k_u$, the graph H will have edges from u_j^i to all the nodes of U that correspond to arcs in G′ that come into node u, and node u_j^o will have edges to all the nodes of U that correspond to arcs in G′ that go out of node u (see Figure 4). We formulate a matching problem on the bipartite graph H as follows. Every node in U will have a requirement of one, i.e., they have to be matched by exactly one edge. Nodes in W of the form u_j^i or u_j^o will also have a requirement of one, and the nodes of W of the form a' that correspond to arc a of G′ will have no requirement on them.

PROPOSITION 3.1. *Let* G *have exactly one* M *node, m. Then, there exists a 4-MET in* G *if and only if the above bipartite matching problem on* H = (U, W), *with the specified matching requirements, has a solution.*

Proof. We will construct 4-MET in G from a matching that satisfies the requirements, and a matching that satisfies the requirements from a 4-MET, to prove the theorem.

(if): Let the solution to the matching problem on H be a set of edges F. We will construct a 4-MET in G from the edges in F. Note that since G has only one M node, a 4-MET in G is a set of loops of length at most four that start at the M node, m.

Suppose node u has $k_u > 0$. Then, k_u nodes of U (corresponding to arcs coming into u in G′) get matched to nodes of the form u_j^i and k_u nodes of U (corresponding to arcs going out of u in G′) get matched to nodes of the form u_j^o. We will join these incoming and outgoing arcs at u in G′ together as part of k_u four-arc loops of a 4-MET in G that start at m, goes to the tail node of one of these k_u arcs that come into node u, along one of these k_u outgoing arcs that go out of u, and returns to node m.

To illustrate this, in Figure 4, let (a, u_1^i) and (d, u_1^o) be part of the solution to the bipartite matching, F. Then, what we are doing is fixing the cycle m–x–u–y–m as part of the 4-MET. Now, we have to deal with the question if there are enough arcs going in and coming out of m to the nodes of G′ that make such a procedure valid. For example, what is the guarantee that there is an arc (or, in general, an appropriate number of arcs) from m to node x? We claim that this is guaranteed by our choice of the number k_u for all the nodes u of G′.

Consider the situation at nodes x and u in Figure 4. To illustrate the flavor of the argument, we will show why, if (a, u_1^i) and (d, u_1^o) are part of F, x has at least one arc coming in from m. If, suppose, $m_x^i = 0$. Then, this implies that $k_x = o_x$, the outdegree of x in G′. If this is the case, then there will be nodes of the form x_j^o, for $j = 1, \ldots, k_x$ $(= o_x)$ in W. Each of the x_j^o will be connected by edges of H only to the nodes of U that correspond to the o_x arcs going out of x in G′. This implies that in F, as x_j^o nodes have a requirement of one, all the nodes of U corresponding to arcs leaving x in G′, will be matched to a node in W of the form x_j^o. This contradicts the fact that arc a leaving x in G′, is matched to node u_1^i, a node not of the form x_j^o. Therefore, it cannot be that $o_x = k_x$. Because H has a matching satisfying all the node requirements, $o_x > k_x$, and therefore, there has to be an arc from m to x. (A similar argument shows that there is at least one arc from node y to m in G.) In fact, the number of arcs from m to x are exactly equal to $o_x - k_x$, which is exactly equal to the number of nodes in U corresponding to the arcs leaving x that are not matched to nodes of the form x_j^o. Therefore, there are always enough arcs going from m to the nodes of G′, and from the nodes of G′ to m, that make valid the above construction of loops of length four in G starting from m and using the two arcs specified by the matching.

If the matching has edges of the form (a, a'), then this corresponds to a length three path in the 4-MET, starting at node m, going to the tail node of a, using arc a to the head node of a and returning to m.

(only if): Let G have a 4-MET. We will construct a bipartite matching F in H, that satisfies all the requirements. A 4-MET consists of a set of loops of length two, three, or four starting from node m and going through nodes of G′. We construct a bipartite matching as follows: A length four loop will consist of two arcs of G′, say a and d in Figure 4. Then the two edges (a, u_j^i) and (d, u_j^o) will be part of the matching, for an unmatched j in $j = 1, \ldots, k_u$. A path of length three will use exactly one arc of G′. For every such arc a, the matching will consist of the edge (a, a'). Ignore length two loops. The argument used in the "if" portion of the proof above also shows that this constitutes a bipartite matching that satisfies all the requirements. □

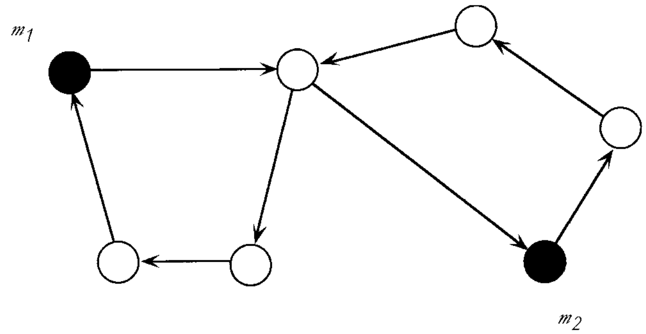

Fig. 5. An example of an LOF graph without a 4-MET but with a 4-MCD.

Now, consider the problem where the LOF graph G has no restriction on the number of maintenance stations. Suppose we want to solve a maintenance routing problem without the balance check station visit requirement. That is, the only maintenance requirement now is to ensure that the routing overnights every aircraft at a maintenance station after at most four days of flying. In terms of our LOF graph, such a routing is a decomposition of the Eulerian graph G into circuits (cycles with repetition of nodes allowed), such that going along each circuit, we encounter at most three N nodes in succession—the 4-MCD that we had defined in the Introduction. An LOF graph can have 4-MCD without having a 4-MET as can be seen from the example in Figure 5.

A 4-MCD can be obtained in polynomial time as a very simple consequence of Proposition 3.1. In G, compress all the M nodes into one node m. Find a 4-MET as in Proposition 3.1. This gives us a partition of the arcs of G into paths of length at most four that go from an M node to another M node in G. Because G is Eulerian, these paths can be combined to give the required circuit decomposition. We formalize this in the following Corollary.

COROLLARY 3.2. *The four-day routing problem without the balance check station visit requirement can be solved in polynomial time.*

Our polynomial time algorithm for finding a 4-MCD shows that the difficulty in finding a 4-MET is in combining the Eulerian routing part with the four-day periodicity part, rather than in just the four-day periodicity requirement.

Another consequence of our 4-MCD algorithm is an application to a dynamic finite-horizon model of routing that we considered in Gopalan and Talluri (1996a). Under this model, the LOFs are allowed to change from day to day, and we consider the four-day routing problem over a fixed number of days. This model is very useful when the schedule changes from day to day or when the flights are of the long-haul variety, spanning multiple days. The argument used to prove Corollary 3.2 also shows that, the four-day routing problem without the balance check station visit requirement can be solved in polynomial time even for routing under this dynamic model. Boundary conditions on maintenance requirements at the beginning of day 1 and at the end of day n can be handled by adding artificial arcs of appropriate lengths. We direct the reader to Gopalan and Talluri (1996a) for more on this model, and leave out the details of this result as they are very similar to preceding arguments.

4. HEURISTICS FOR FINDING A 4-MET

4.1 The Arc-Ranking Heuristic

The arc-ranking heuristic is based on the polynomial time algorithm of Gopalan and Talluri (1996a) for finding a 3-MET in an LOF graph G. We briefly review the 3-MET algorithm. In the 3-MET algorithm, we partition the arcs coming into, and going out of each N node j into four sets:

$m^i_{j\mathrm{N}}$ = Number of arcs coming into j from nonmaintenance stations.
$m^i_{j\mathrm{M}}$ = Number of arcs coming into j from maintenance stations.
$m^o_{j\mathrm{N}}$ = Number of arcs going out of j to nonmaintenance stations.
$m^o_{j\mathrm{M}}$ = Number of arcs going out of j to maintenance stations.

We split every $j \in \mathrm{N}$ as shown in Figure 6 and find a Euler tour in the resulting directed graph. The splitting ensures that at most two nonmaintenance stations are visited in succession in G. A necessary condition for G to have a 3-MET is that every nonmaintenance station $j \in \mathrm{N}$ has $m^o_{j\mathrm{M}} \geqslant m^i_{j\mathrm{N}}$. For the 4-MET case, it is possible that G has a $j \in \mathrm{N}$ with $m^o_{j\mathrm{M}} < m^i_{j\mathrm{N}}$, and still has a 4-MET. We will extend our 3-MET algorithm to handle this case.

Figure 7 depicts the idea behind the arc-ranking heuristic. A node $j \in \mathrm{N}$ that has $m^o_{j\mathrm{M}} < m^i_{j\mathrm{N}}$ must be the middle N node of a N–N–N sequence exactly $m^i_{j\mathrm{N}}$–$m^o_{j\mathrm{M}}$ times in any 4-MET. The ranking heuristic attempts to ensure that the N sequence is not extended beyond three in a row by ensuring visits to "good" N nodes after j. Figure 7a shows the split of node j at the beginning of the heuristic. In the next

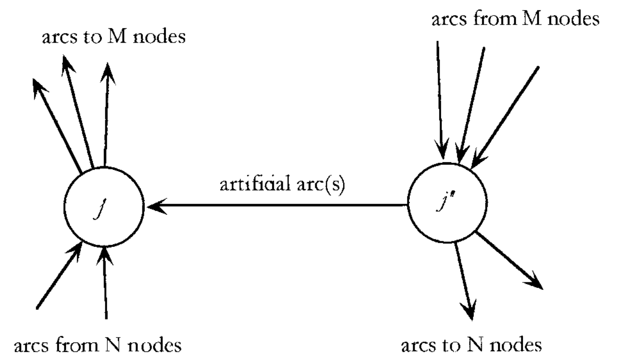

Fig. 6. Splitting nonmaintenance nodes to find a feasible 3-MET ($m^o_{j\mathrm{M}} \geqslant m^i_{j\mathrm{N}}$, $\forall j \in \mathrm{N}$).

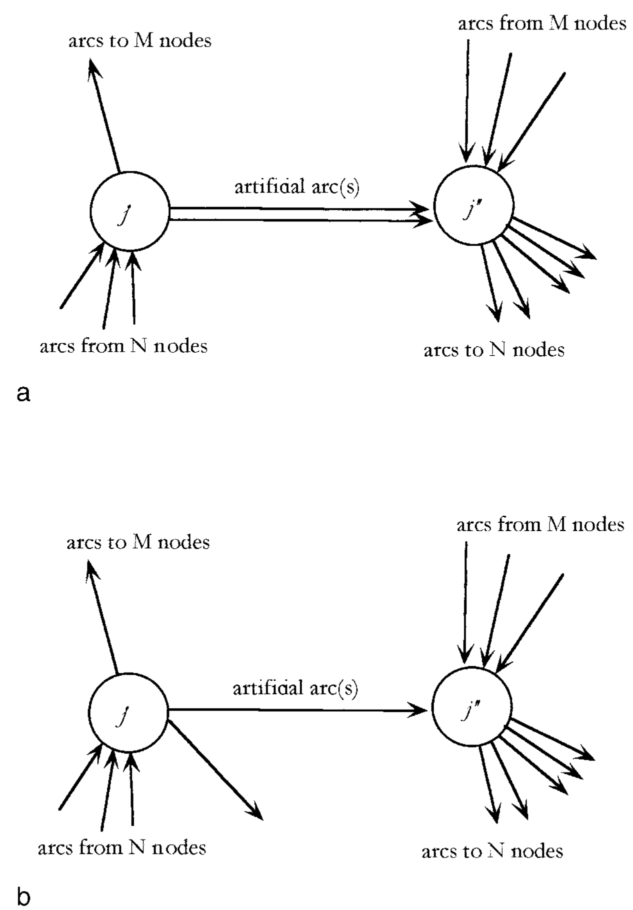

Fig. 7. Splitting nonmaintenance nodes to find a feasible 4-MET ($m^o_{j\mathrm{M}} < m^i_{j\mathrm{N}}$). a. First iteration of arc splitting heuristic. No arcs removed from outlist of j'. b. Second iteration of arc splitting heuristic. Best arc removed from outlist of j'.

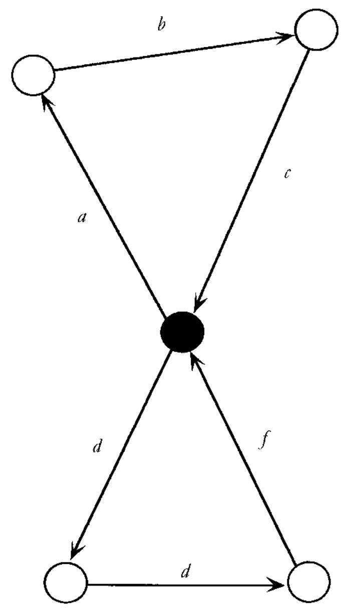

Fig. 8. Forcing an arc sequence may not lead to an Euler tour.

step, we examine all arcs going out to N nodes from j and rank them in decreasing order of $m^o_{j\mathrm{M}}$–$m^i_{j\mathrm{N}}$. The "best arc" going out of j is then the arc going to the N node with the highest rank. We remove this "best arc" from the out list of node j'' and insert it into the out list of node j', simultaneously reducing the number of artificial arcs between j'' and j' by one. However, we can perform this action only if a certain condition called the connectivity condition is satisfied. We elaborate on the connectivity condition in a moment. In the next iteration of the arc ranking heuristic, we remove the next best arc from the out list of j'' and insert it into the outlist of node j' (and reduce the number of artificial arcs), provided again that the connectivity condition is satisfied after this maneuver. We perform arc ranking for every j that has $m^o_{j\mathrm{M}} < m^i_{j\mathrm{N}}$.

The arc-ranking procedure constrains the number of choices we have in finding a Euler tour, by encouraging visits to some N nodes, but not to others. We are enforcing a specific sequence of visits in the Euler tour. However this could prove detrimental, as Figure 8 illustrates. In Figure 8, if arc c is forced to turn to arc a, arc f must turn to d, and we cannot find a Euler tour in the resulting graph. In any Euler tour in Figure 8, arc c must turn to d and arc f to a.

To prevent situations such as the one in Figure 8, at every stage, we test if connectivity is maintained. To test for connectivity at any stage, (say if we are trying to determine if c should turn to a), replace arcs c and a by a single arc starting at the tail of c and ending at the head of a, and check if the resulting graph is connected.

4.2 The Matching Based Heuristic

This heuristic is based on the polynomial time algorithm of Section 3 for finding a 4-MCD in an LOF graph. A 4-MET is also a 4-MCD. Therefore, having a 4-MCD is a necessary condition for G to have a 4-MET. We therefore first find a 4-MCD in G. If G does not have a 4-MCD, we change the LOF graph appropriately to satisfy the necessary conditions for it to have a 4-MCD, and repeat the procedure.

If, suppose, we were able to find a 4-MCD in G, then we will try to obtain a 4-MET from it by patching the circuits to get a 4-MET. One simple approach to do this is to fix all the turns at N nodes from the 4-MCD and then try to find a Euler tour in the resulting graph. A more general approach is to fix the turns at N nodes while we are in the process of finding the Euler tour, at each stage ensuring that the graph obtained by fixing that turn has a 4-MCD.

Because solving a bipartite matching problem is computationally very easy, this procedure, although slower than the arc-ranking heuristic, is still very practical.

4.3 The Simple Euler Tour Heuristic

In any graph, several alternate Euler tours are available, and this heuristic takes advantage of this fact to generate a 4-MET. Given a LOF Graph, we find an arbitrary Euler tour first; if this tour does not contain a sequence of four nonmaintenance stations, we are done. Else, we pick some violating sequence of four nonmaintenance stations $n_1n_2n_3n_4$ in the Euler tour and attempt to break it up. If we choose to break the sequence after n_2, we find a cycle $\mathrm{C}_{n_2} = n_2m_1 \cdots m_2n_2$ in another part of the Euler tour, and insert this sequence of visits after n_1. Because we ensure that m_1 follows n_2, we have broken up the sequence (actually we do not require that m_1 immediately follow n_2, only that it happens sufficiently early to prevent violation of the 4-MET condition). Also because m_2 precedes n_2 prior to the termination of the cycle, we now have only three N nodes in sequence in the Euler tour. One more condition needs to be observed in this reinsertion procedure. If the original Euler tour was A–C_{n_2}–B–$n_1n_2n_3n_4$–C, where A, B, and C are a series of nodes, then after reinsertion, blocks A and B are contiguous in the Euler tour. We make sure, while reinserting, that we do not create further violations of the 4-MET condition.

5. COMMENTS

IN THIS PAPER we have proved complexity results regarding the four-day routing problem and the three-day routing problem with red-eyes. We have also given a polynomial time algorithm for finding a 4-MCD in a given LOF graph. This result is of independent interest to airlines that do not have a balance check requirement (Kabbani and Patty, 1992).

The polynomial time algorithm for the 3-MET presented in Gopalan and Talluri (1996a) and the polynomial time algorithm for the 4-MCD presented in this paper, form the core of our heuristics for finding a four-day maintenance routing. More generally, our solution strategy of generating an LOF graph and finding a routing in it differs considerably from previous approaches to the problem. Our strategy exploits the nice underlying structure of the problem, is very fast, and, in our experience, very effective.

There are a number of issues related to the situation when the LOF graph, or even the flight graph, is disconnected (such situations are commonly called locked rotations). The LOF graph generating procedures of Gopalan and Talluri (1996a) try to ensure that we always generate a connected LOF graph. When the flight network for an equipment type itself is disconnected, we have no choice but to swap equipment types to make it connected (making sure we do not make the flight networks of the other equipment types disconnected in the process). This is an interesting research problem that needs further investigation. In TALLURI (1996), a heuristic for this problem using Steiner trees and an equipment swapping algorithm is presented, but the complexity of this problem is open—perhaps it can be solved in polynomial time.

ACKNOWLEDGMENTS

I WOULD LIKE TO THANK Ram Gopalan for many discussions on this topic and for suggesting the Arc Ranking heuristic which was primarily his idea. This work was done while I was at the USAir Operations Research Department. Sridhar Ramachandran was one of the key players in implementing the routing modules and I have benefited from discussions with him, along with other members of the FAM team, Russ Rushmeir and Jeff Dillon. Thanks also to the referees for many valuable comments and suggestions.

REFERENCES

BONDY, J. A. AND U. S. R. MURTY, *Graph Theory with Applications.* McMillan, London, 1978.

CLARKE, L. W., E. L. JOHNSON, G. L. NEMHAUSER, AND Z. ZHU, "Aircraft Rotation Problems," *Ann. OR Math. Ind. Syst.* (1996). (To appear)

DILLON, J., R. GOPALAN, S. RAMACHANDRAN, R. RUSHMEIER, AND K. T. TALLURI, "Fleet Assignment at USAir," in *TIMS/ORSA Joint National Meeting,* Phoenix, AZ, 1993.

FEO, T. A. AND J. F. BARD, "Flight Scheduling and Maintenance Base Planning," *Manage. Sci.* **35,** 12, 1415–1432 (1989).

GAREY, M. A. AND D. S. JOHNSON, *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman & Co., 1979.

GOPALAN, R. AND K. T. TALLURI, "The Aircraft Maintenance Routing Problem," to appear in *Opns. Res.* March, 1998

HANE, C. A., C. BARNHART, E. J. JOHNSON, R. E. MARSTEN, G. L. NEMHAUSER, AND G. SIGISIMONDI, "The Fleet Assignment Problem: Solving a Large-Scale Integer Program," *Math. Program.* **70,** 211–232, 1995.

JARRAH, A. I. "The Fleet Assignment Model," in *TIMS/ORSA Joint National Meeting,* Chicago, IL, 1993.

JARRAH, A. I. AND G. YU, A Mathematical Programming Model for Airline Maintenance Scheduling, Working paper 89/90-3-3, Department of Management Science and Information Systems, University of Texas at Austin, 1990.

KABBANI, N. M. AND B. W. PATTY, "Aircraft Routing at American Airlines," in *Proceedings of the 32nd Annual Symposium of AGIFORS,* Budapest, Hungary, 1992.

SUBRAMANIAM, R., R. P. SCHEFF, J. D. QUILLINAN, D. S. WIPER, AND R. E. MARSTEN, "ColdStart—Daily Fleet Assignment Model," *Interfaces* **24,** 104–119, 1994.

TALLURI, K. T., "Swapping Applications in a Daily Airline Fleet Assignment," *Transp. Sci.* **30,** 237–248, 1996.

(Received, August 1996; revisions received: January 1997; accepted: March 1997)