10th International Conference on Application of Advance Technologies in Transportation (AATT 2008), Athens, 27-31 May, 2008.

HEURISTICS FOR MAXIMIZING FLEET AVAILABILITY SUBJECT TO FLIGHT & MAINTENANCE REQUIREMENTS

George Kozanidis[1], Andreas Gavranis[2], George Liberopoulos[3]

ABSTRACT. Flight and Maintenance Planning (FMP) addresses the question of which available aircraft should fly and for how long, and which grounded aircraft should perform maintenance operations, in a group of aircraft that comprise a unit. The objective is to achieve maximum availability of the unit over the planning horizon. In this work, we develop three heuristic solution procedures for the FMP problem. We present computational results which illustrate the computational performance of these procedures and evaluate the quality of the solutions that they produce. These results are very satisfactory, because they demonstrate that, under careful consideration, even large FMP instances can be handled quite effectively.

INTRODUCTION

Flight and Maintenance Planning (FMP) addresses the question of which available aircraft should fly and for how long, and which grounded aircraft should perform maintenance operations, in a group of aircraft that comprise a unit. The objective is to achieve maximum fleet availability of the unit over the planning horizon. Fleet availability is expressed in terms of the total number of aircraft that are available to fly (aircraft availability) and in terms of the total residual flight time of all available aircraft (flight time availability). The residual flight time of an aircraft is defined as the total remaining time that this aircraft can fly before it has to be grounded for maintenance check. FMP is a very important decision making problem arising in the operation of numerous types of fleets, involving military or fire-fighting aircraft, rescue choppers, etc.

Kozanidis and Skipis (2006) developed a multiobjective mixed integer linear program (MILP) for the FMP problem. The 4 objectives of that model maximize the minimum aircraft and flight time availability of the unit (e.g., a wing) and of its subunits (e.g., squadrons), respectively. The model was developed for use on a specific aircraft type, but can be applied

[1] Lecturer, Dept. of Mechanical & Industrial Engineering, University of Thessaly, Volos, Greece, e-mail: gkoz@mie.uth.gr
[2] PhD Candidate, Dept. of Mechanical & Industrial Engineering, University of Thessaly, Volos, Greece and Lieutenant, 111 Combat Wing, Hellenic Air Force, N. Anghialos 37400, Greece, e-mail: agavranis@gmail.com
[3] Associate Professor, Dept. of Mechanical & Industrial Engineering, University of Thessaly, Volos, Greece, e-mail: glib@mie.uth.gr

repeatedly until all plans have been issued, if more than one aircraft types are present. Although it considers a single type of maintenance check, it can be easily extended to incorporate additional ones. The computational effort that this model needs in order to reach a nondominated solution increases rapidly with problem size, as is expected for problems of this type and as is verified by computational results. As a result, its applicability on large problems is quite limited. This raises the need to develop alternative intelligent approaches in order to address large FMP instances. To this end, we develop three heuristic solution procedures for the FMP problem in this work. These procedures are called Aircraft Flowchart Heuristic 1 and 2 and Horizon Splitting Heuristic, and are described in the next section. For the remainder of this work, we use the terminology and notation introduced by Kozanidis and Skipis (2006).

Although the research literature dealing with airline operations is quite rich, none of the reported works deals with the problem that we address in this work. To the best of our knowledge, no such work has been published to date, perhaps because most of the publicly available research in this area has been directed towards problems in the commercial airline industry, which have different objectives and requirements than those in the Air Force. The only reference related to the FMP problem that we know of is a Field Manual of the U.S. Department of the Army (US DoA, 2000), which describes a graphical tool for scheduling aircraft for periodic inspection and deciding which aircraft should fly in certain missions.

HEURISTIC SOLUTION PROCEDURES

Aircraft Flowchart Heuristic (AFH)

The Hellenic Air Force (HAF) and many other Air Force organizations worldwide, solve the FMP problem empirically, utilizing in an ad-hoc manner a 2-dimensional graphical tool called the "aircraft flowchart," (see Figure 1 in Kozanidis and Skipis, 2006). In current practice, the aircraft flowchart is at best used as a graphical device by the officer responsible for issuing the flight and maintenance plans. For example, in an aviation maintenance manual of the U.S. Army where this flowchart is described (US DoA, 2000), this officer is simply advised to utilize the flowchart by "flying the aircraft that are above the diagonal to get them down to the line" and "holding the aircraft that are below the diagonal to bring them up to the line". No particular instructions are given on how this can be implemented effectively. Clearly, this procedure is highly subjective and dependent on numerous minor decisions made by the user. The first two heuristics that we propose are aimed at implementing the aircraft flowchart procedure more systematically. The first one takes into consideration the squadron each aircraft belongs to, while the second focuses on the wing and treats the aircraft as if they all belong to the same squadron. We term these two variants AFH1 and AFH2, respectively, and we introduce AFH1 first.

Aircraft Flowchart Heuristic 1

The application of AFH1 requires a number of minor decisions in each period of the planning horizon. For this reason, AFH1 computes a "priority index" for each squadron at the beginning of each period, and makes all relevant decisions thereafter based on this index. The first such decision regards the allocation of the maintenance station's time capacity to the grounded aircraft. After this decision has been made, the number of maintenance dock spaces that will be available at the beginning of the next period is determined. This information is important, because the production of the flight plans of the available aircraft depends strongly upon it.

The priority index of squadron m in period t is defined as the flight load of squadron m in period t, divided by its total residual flight time at the beginning of period t. A higher priority index value reveals that the squadron experiences heavier load with respect to its current availability; therefore, higher priority should be given to the grounded aircraft that belong to this squadron. Hence, the maintenance station gives priority to the aircraft that belong to the squadron with the highest priority index first, and so on. In order to free dock space, the maintenance station works continuously on the same aircraft within each period, until either its time capacity is fully utilized, or the service of that aircraft is completed. Of course, if not enough time capacity exists, the service of an aircraft may be spread out over more than one periods. Every time that an aircraft completes its maintenance service, the priority index of the corresponding squadron is recomputed after Y time units are added to its total residual flight time, to reflect the fact that this aircraft becomes available.

After the maintenance plans of all grounded aircraft in period t are produced, the number of dock spaces that will be available at the beginning of period t+1 is determined as well. These spaces will be occupied by the aircraft that will enter the station for service at the beginning of period t+1. Besides determining the order in which the aircraft will receive maintenance service, the priority indices are also used to determine the order in which the squadrons will occupy dock space that is emptied at the maintenance station. Thus, the aircraft of the squadron with the highest priority index value are considered first, and so on. Whether a specific aircraft will be grounded in period t+1 or not, depends on its current residual flight time, as compared to its current proportionate flight load. More specifically, the aircraft with the lowest residual flight time in this squadron will be grounded at the beginning of period t+1 if its proportionate flight load over period t is greater or equal to its residual flight time at the beginning of the same period. The proportionate flight load of each available aircraft is computed by dividing the flight load of the corresponding squadron by the total number of available aircraft in that squadron. If this check denotes that this aircraft should be grounded for maintenance at the beginning of period t+1, then its flight time in period t is set equal to its residual flight time at the beginning of the same period. Additionally, the priority index value of the squadron it belongs to is recomputed, after this time is subtracted from the total residual flight time and from the flight load of this squadron. If additional empty spaces exist, the squadron with the highest priority index is considered next, and so on. This procedure is repeated until either there do not exist empty dock spaces at the maintenance station, or until there do not exist candidate aircraft for entering the maintenance station.

Once the aircraft that will exit and enter the maintenance station in period t+1 are determined, the aircraft that will be available at the beginning of period t+1 are determined as well. Then, the flight time of each available aircraft in period t is determined by solving a simple nonlinear optimization problem that minimizes the total deviation index that will be realized at the beginning of period t+1. Assume that the aircraft of squadron m that will be available in period t+1 (let their total number be N) are arranged on a flowchart in nondecreasing order of their residual flight times at the beginning of period t. On this flowchart, consider the line segment (called “the diagonal”), connecting the origin with the point with coordinates (N, Y), where Y is the maximum time that an aircraft can fly between two consecutive maintenance checks, often referred to as “phase interval” in the related literature. The index of the aircraft with the smallest residual flight time at the beginning of period t is equal to 1 in this arrangement. Therefore, if s is the slope of the diagonal, the residual flight time of this aircraft at the beginning of period t+1 should ideally be equal to $s(1) = s$. Similarly, the index of the aircraft with the next largest residual flight time at the beginning of period t is equal to 2; therefore its residual flight time at the beginning of period t should ideally be equal to $2s$, etc. An aircraft that will exit the maintenance station at the beginning of period t+1 is considered to have residual flight time equal to Y at the beginning of period t in this arrangement, but its flight time during period t is also restricted to 0-value (since this aircraft will be grounded during period t). Clearly, s is equal to Y/N. Thus, the problem of deciding the flight time of each available aircraft of squadron m reduces to a nonlinear optimization problem, in which the total deviation index value that will be realized at the beginning of period t+1 is minimized. The functional constraints of this problem ensure that the sum of flight times of all the aircraft that belong to squadron m lies in the interval $[LS_{mt}, US_{mt}]$, that the flight time of each aircraft does not exceed X_{max} (or 0 in the case of an aircraft that will be grounded during period t), and that the residual flight time of each available aircraft at the beginning of period t+1 is at least equal to Y_{min}. To illustrate this procedure, assume that we have determined the N aircraft of squadron m that will be available at the beginning of period t+1. In order to produce their flight plans, we arrange them in nondecreasing order of their residual flight times at the beginning of period t. If we replace the indices m and n of each aircraft by the index i, for simplicity, the nonlinear optimization problem that arises is the following:

$$\text{Min } \sum_{i=1}^{N} (y_{it+1} - is)^2$$

$$\text{s.t. } y_{it+1} = y_{it} - x_{it}, I = 1,\ldots,N$$

$$LS_{mt} \le \sum_{i=1}^{N} x_{it} \le US_{mt}$$

$$x_{it} \le X_{max}, i = 1,\ldots, N$$

$$y_{it+1} \ge Y_{min}, i = 1,\ldots, N$$

$$x_{it} \ge 0, i = 1,\ldots, N$$

In the objective function, the sum of squares of all vertical deviations from the diagonal that will be realized at the beginning of period t+1 is minimized. The first set of constraints updates the residual flight of each aircraft at the beginning of period t+1, based on its residual flight time at the beginning of period t and its flight time during period t. The second set of constraints ensures that the flight requirements of squadron m in period t are satisfied. The next two sets of constraints impose upper and lower bounds on the flight time and on the residual flight time of each aircraft, respectively, and the last set of constraints accounts for the nonnegativity of the flight times. Note that the y_{it+1}'s and the x_{it}'s are decision variables in this formulation, while the y_{it}'s are known parameters. In order to solve this problem, we obtain, after some basic manipulation, the following equivalent formulation:

$$\text{Min } \sum_{i=1}^{N} ((y_{it} - is) - x_{it})^2 \quad (1)$$

$$\text{s.t. } LS_{mt} \le \sum_{i=1}^{N} x_{it} \quad (2)$$

$$\sum_{i=1}^{N} x_{it} \le US_{mt} \quad (3)$$

$$0 \le x_{it} \le X_{ui} = \min(X_{max}, y_{it} - Y_{min}), i = 1,\ldots, N \quad (4)$$

The problem defined by (1)-(4) is a quadratic programming problem. It can be easily shown that its objective function is convex. Therefore, the KKT conditions (see Bazaraa et al., 2006) are necessary and sufficient for optimality. We give next a simple procedure called "Sweep" that can be used to obtain the optimal solution. On the corresponding flowchart described above, consider a line parallel to the diagonal which is placed initially far to the top so that all aircraft lie below it, as shown in Figure 1 (in what follows, we refer to a point on the graph and the aircraft whose residual flight time this point represents, interchangeably). Assume now that this line starts moving towards the diagonal (and past it, while always remaining parallel to it), sweeping along each aircraft that it comes across. As this move takes place, flight times are accordingly assigned to the aircraft in the order that they are swept by the line (each aircraft is allowed to move vertically only). If during this procedure one of the aircraft swept by the line reaches its maximum flight time, X_{ui}, then the line should "disengage" this aircraft, to ensure that the resulting solution will remain feasible. Therefore, the line should continue its move without sweeping further this aircraft in that case.

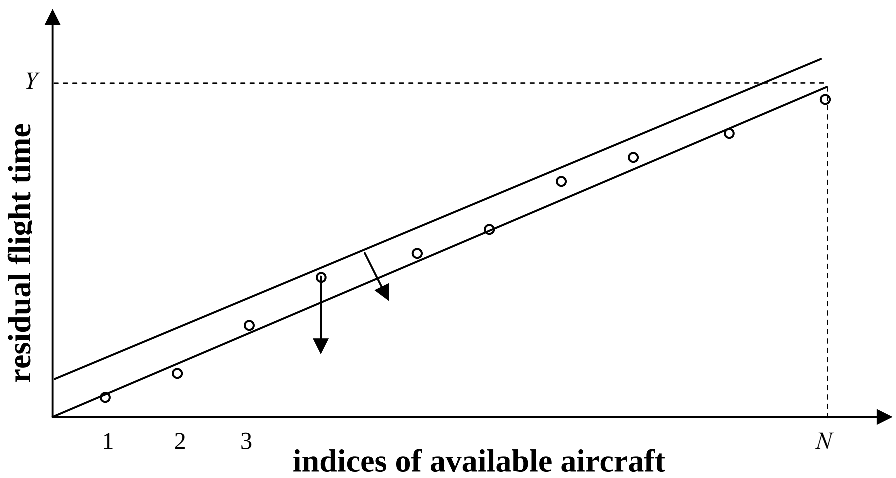

Figure 1. Illustration of the Procedure "Sweep"

Consider now the following 4 solutions obtained during the application of this procedure:

1. The solution in which the sum of the assigned aircraft flight times is equal to LS_{mt}. In what follows, we refer with "L" to the value of this sum.
2. The solution in which the sum of the assigned aircraft flight times is equal to US_{mt}. In what follows, we refer with "U" to the value of this sum.
3. The solution in which each aircraft is assigned its maximum possible flight time. In what follows, we refer with "X" to the sum of the assigned aircraft flight times of this solution.
4. The solution in which the sweeping line coincides with the diagonal. In what follows, we refer with "D" to the sum of the assigned aircraft flight times of this solution.

The following is a very crucial and interesting result, utilized in the development of AFH1:

Proposition 1. If the quantities L, U, X and D are placed in nondecreasing order, then:

a) If, after taking into consideration all ties present (if any), there does not exist an arrangement in which L precedes X, then the problem defined by (1)-(4) is infeasible.

b) If an arrangement in which L precedes X exists, then the optimal solution of the problem defined by (1)-(4) is the one obtained from Procedure Sweep when the sum of the assigned aircraft flight times becomes equal to the quantity that appears second in this arrangement.

Proof. See Appendix. □

The application of Procedure Sweep produces the flight time of each available aircraft, which concludes the computations for the current period. The same steps are then repeated successively for each period, until the aircraft flight and maintenance plans of the entire planning horizon are produced. Based on this discussion, the detailed steps of AFH1 are introduced next. The following additional notation is used in the pseudocode:

Sa_{mt} = number of available aircraft of squadron m in period t
Sx_{mt} = total flight time of squadron m in period t
Sy_{mt} = total residual flight time of squadron m in period t
Sh_{mt} = total maintenance time of squadron m in period t
Sg_{mt} = total residual maintenance time of squadron m in period t
$exit_{mt}$ = number of aircraft of squadron m exiting the maintenance station at the end of period t
B_{res} = residual maintenance time capacity
C_{res} = residual maintenance space capacity

Aircraft Flowchart Heuristic 1
Step 0: initialization

```
Sa_m1 = 0, Sy_m1 = 0, Sg_m1 = 0, C_res = C
for m = 1 to |M| do
        for n = 1 to |N_m| do
                a_mn1 = A1_mn, Sa_m1 = Sa_m1 + a_mn1, C_res = C_res – 1 + a_mn1
                y_mn1 = Y1_mn, Sy_m1 = Sy_m1 + y_mn1
                g_mn1 = G1_mn, Sg_m1 = Sg_m1 + g_mn1
        end for
        arrange in nondecreasing order of y_mn1 the available aircraft of squadron m
        arrange in nondecreasing order of g_mn1 the grounded aircraft of squadron m
end for
for t = 1 to T do
        Step 1: development of maintenance plans
        B_res = B_t, exit_mt = 0, h_mnt = 0, Sh_mt = 0
        while B_res > 0 and grounded aircraft exist do
                k = argmax_{m∈M: Sg_mt − Sh_mt > 0} S_mt /(Sy_mt + (exit_mt * Y))
                l = argmin_{n∈N_k} (g_knt - h_knt): (g_knt - h_knt) > 0
                if B_res ≥ g_klt
                        B_res = B_res - g_klt , h_klt = g_klt, Sh_kt = Sh_kt + h_klt
                        C_res = C_res + 1, exit_mt = exit_mt +1
                else
                        h_klt = B_res, Sh_kt = Sh_kt + h_klt, B_res = 0
        end while
        Step 2: decision on aircraft that will be grounded
        x_mnt = 0, Sx_mt = 0
        while C_res > 0 and not all squadrons have been considered do
                k = argmax_{m∈M} (S_mt - Sx_mt)/(Sy_mt - Sx_mt + (exit_mt * Y))
                l = argmin_{n∈N_k} (y_knt - x_knt): y_knt - x_knt > 0
                if (S_kt – Sx_kt)/(Sy_kt – Sx_kt ) > y_klt
                        x_klt = y_klt, Sx_kt = Sx_kt + x_klt, C_res = C_res - 1
                else
                        remove k from set of squadron indices that have not been considered
```

end while
Step 3: development of flight plans
for m = 1 to $|M|$ do
using Procedure Sweep, issue the aircraft flight plans of squadron m in period t
end do
Step 4: next period update
$Sa_{mt+1} = Sa_{mt}$, $Sy_{mt+1} = Sy_{mt}$, $Sg_{mt+1} = Sg_{mt}$
for m = 1 to $|M|$ do
for n = 1 to $|N_m|$ do
if (a_{mnt} == 0 and h_{mnt} == g_{mnt})
$a_{mnt+1} = 1$, $Sa_{mt+1} = Sa_{mt+1} + 1$
$y_{mnt+1} = Y$, $Sy_{mt+1} = Sy_{mt+1} + Y$
$g_{mnt+1} = 0$, $Sg_{mt+1} = Sg_{mt+1} - h_{mnt}$
else if (a_{mnt} == 1 and $x_{mnt} < y_{mnt}$)
$a_{mnt+1} = 1$
$y_{mnt+1} = y_{mnt} - x_{mnt}$, $Sy_{mt+1} = Sy_{mt+1} - x_{mnt}$
$g_{mnt+1} = 0$
else if (a_{mnt} == 1 and x_{mnt} == y_{mnt})
$a_{mnt+1} = 0$, $Sa_{mt+1} = Sa_{mt+1} - 1$
$g_{mnt+1} = G$, $Sg_{mt+1} = Sg_{mt+1} + G$
$y_{mnt+1} = 0$, $Sy_{mt+1} = Sy_{mt+1} - x_{mnt}$
else if (a_{mnt} == 0 and $h_{mnt} < g_{mnt}$)
$a_{mnt+1} = 0$
$g_{mnt+1} = g_{mnt} - h_{mnt}$, $Sg_{mt+1} = Sg_{mt+1} - h_{mnt}$
$y_{mnt+1} = 0$
end for
end for
end for □

In order to avoid confusion and keep the above pseudocode clear to the reader, the restriction that a positive residual maintenance time should be at least equal to G_{min} is not taken into consideration in the development of the maintenance plans, although in the actual coding it is.

<u>Computational Complexity of AFH1</u>

For the computational complexity analysis of AFH1, we prove the next interesting result first.

Lemma 1. The problem defined by (1)-(4) can be solved in time O(N), where N is the total number of variables x_{it}.
Proof. The values of L and U are known. The values of D and X can be computed in time O(N), since $\mathrm{D} = \sum_{i=1}^{N} [y_{it} - is]^{+}$ and $\mathrm{X} = \sum_{i=1}^{N} X_{ui}$. Finding if there exists an arrangement of L, U, D and X in which L precedes X requires time O(1). If such an arrangement exists and S is the second quantity in order, the problem defined by (1)-(4) can be equivalently transformed into the following problem in time O(N):

$$\text{Min } \sum_{i=1}^{N} \left(\frac{1}{2} d_i x_{it}^{2} - a_i x_{it}\right)$$

s.t. $\sum_{i=1}^{N} b_i x_{it} = b_0$

$l_i \leq x_{it} \leq u_i$, $i = 1,\ldots, N$,

where $d_i = 2$, $a_i = 2(y_{it} - is)$, $b_i = 1$, $b_0 = S$, $l_i = 0$ and $u_i = X_{ui}$, for $i = 1,\ldots, N$. This problem can be solved in time O(N) (see Brucker, 1984). Therefore, the problem defined by (1)-(4) can be solved in total time O(N). □

Let $N_{\max} = \max_{m \in M} \left| N_m \right|$. Proposition 2 utilizes Lemma 1 in order to analyze the computational complexity of AFH1.

Proposition 2. AFH1 requires time O(|M| max(N_{max}logN_{max}, TN_{max}, TC, T|M|)).
Proof. See Appendix. □

Aircraft Flowchart Heuristic 2

AFH2 acts exactly the same way as AFH1, but assumes that all aircraft belong to the same squadron and does not make the relevant decisions based on squadron priority indices. The next aircraft to receive maintenance service is always the one with the smallest residual maintenance time among all grounded aircraft. Aircraft for entering the maintenance station in the next period are considered in nondecreasing order of their residual flight times, independently of the squadron they belong to. The aircraft flight plans of each period are produced solving one problem such as the one defined by (1)-(4) for each squadron. Note however that, in the corresponding arrangement, the index i of each aircraft determines its order when all the aircraft of the wing (and not only those of the squadron it belongs to) are arranged in nondecreasing order of their residual flight times. Based on this discussion, the detailed steps of AFH2 are:

Aircraft Flowchart Heuristic 2
Step 0: initialization

```
Sa_m1 = 0, Sy_m1 = 0, Sg_m1 = 0, C_res = C
for m = 1 to |M| do
        for n = 1 to |N_m| do{
                a_mn1 = A1_mn, Sa_m1 = Sa_m1 + a_mn1, C_res = C_res – 1 + a_mn1
                y_mn1 = Y1_mn, Sy_m1 = Sy_m1 + y_mn1
                g_mn1 = G1_mn, Sg_m1 = Sg_m1 + g_mn1
        end for
        arrange in nondecreasing order of y_mn1 the available aircraft of squadron m
        arrange in nondecreasing order of g_mn1 the grounded aircraft of squadron m
end for
arrange in nondecreasing order of y_mn1 the available aircraft of the wing
arrange in nondecreasing order of g_mn1 the grounded aircraft of the wing
for t = 1 to T do
        Step 1: development of maintenance plans
        B_res = B_t, h_mnt = 0
        while B_res > 0 and grounded aircraft exist do
                k,l = arg min_{m∈M, n∈N_m} (g_mnt - h_mnt): (g_mnt - h_mnt) > 0
                if B_res ≥ g_klt
                        B_res = B_res - g_klt , h_klt = g_klt, C_res = C_res + 1
                else
```

$h_{klt} = B_{res}$, $B_{res} = 0$
end while
Step 2: decision on aircraft that will be grounded
$x_{mnt} = 0$, $Sx_{mt} = 0$
while $C_{res} > 0$ and not all squadrons have been considered do
$k,l = \arg\min_{m \in M,\, n \in N_m} (y_{mnt} - x_{mnt})$: $y_{mnt} - x_{mnt} > 0$
if $((S_{kt} - Sx_{kt})/(Sy_{kt} - Sx_{kt})) > y_{klt}$
$x_{klt} = y_{klt}$, $Sx_{kt} = Sx_{kt} + x_{klt}$, $C_{res} = C_{res} - 1$
else
remove k from set of squadron indices that have not been considered
end while
Step 3: development of flight plans
for $m = 1$ to $|M|$ do
using Procedure Sweep, issue the aircraft flight plans of squadron m in period t
end do
Step 4: next period update
$Sa_{mt+1} = Sa_{mt}$, $Sy_{mt+1} = Sy_{mt}$, $Sg_{mt+1} = Sg_{mt}$
for $m = 1$ to $|M|$ do
for $n = 1$ to $|N_m|$ do
if ($a_{mnt} == 0$ and $h_{mnt} == g_{mnt}$)
$a_{mnt+1} = 1$, $Sa_{mt+1} = Sa_{mt+1} + 1$
$y_{mnt+1} = Y$, $Sy_{mt+1} = Sy_{mt+1} + Y$
$g_{mnt+1} = 0$, $Sg_{mt+1} = Sg_{mt+1} - h_{mnt}$
else if ($a_{mnt} == 1$ and $x_{mnt} < y_{mnt}$)
$a_{mnt+1} = 1$
$y_{mnt+1} = y_{mnt} - x_{mnt}$, $Sy_{mt+1} = Sy_{mt+1} - x_{mnt}$
$g_{mnt+1} = 0$
else if ($a_{mnt} == 1$ and $x_{mnt} == y_{mnt}$)
$a_{mnt+1} = 0$, $Sa_{mt+1} = Sa_{mt+1} - 1$
$g_{mnt+1} = G$, $Sg_{mt+1} = Sg_{mt+1} + G$
$y_{mnt+1} = 0$, $Sy_{mt+1} = Sy_{mt+1} - x_{mnt}$
else if ($a_{mnt} == 0$ and $h_{mnt} < g_{mnt}$)
$a_{mnt+1} = 0$
$g_{mnt+1} = g_{mnt} - h_{mnt}$, $Sg_{mt+1} = Sg_{mt+1} - h_{mnt}$
$y_{mnt+1} = 0$
end for
end for
arrange in nondecreasing order of y_{mnt+1} the available aircraft of the wing
end for □

Although the order of the available aircraft within each squadron does not change when Procedure Sweep is applied, the order of the available aircraft within the wing may change. For this reason, a rearrangement of this order takes place at the end of each time period.

Computational Complexity of AFH2

Letting $N = \sum_{m \in M} N_m$, we get the following result for the computational complexity of AFH2:

Proposition 3. AFH2 requires time $O(|M|N_{max}\max(\log N_{max}, T\log|M|))$.
Proof. See Appendix. □

Horizon Splitting Heuristic (HSH)

The third heuristic that we propose for the solution of large FMP instances makes use of the simple idea of splitting the original planning horizon into several consecutive ones, and solving an FMP subproblem for each of them. The ending system state of each smaller horizon becomes the beginning state of the next one, and so on. The smaller horizons do not necessarily need to have equal lengths. The quality of the solution obtained this way is expected to be inferior to the one obtained when the problem is solved up front for all the periods of the original planning horizon. On the other hand, the total computational time needed in order to reach a solution is expected, in general, to decrease, especially as the length of the smaller horizons decreases. This is mainly because the computational effort needed to reach an optimal solution with MILP is expected, in general, to increase with problem size.

COMPUTATIONAL RESULTS

AFH1 and AFH2 were coded in C/C++ and the coding is available upon request. For the solution of MILP, we applied the weighted sums approach (see Steuer, 1986), by introducing strictly positive weights w_1, w_2, w_3 and w_4 such that $w_1 + w_2 + w_3 + w_4 = 1$. Since the residual flight time of an available aircraft is equal to $Y/2$ on the average, we multiplied the first objective with $Y/2$, the third objective with $|M|(Y/2)$, and the fourth objective with $|M|$ for scaling reasons. A single criterion problem was obtained this way with objective $Z = (Y/2)w_1z_1 + w_2z_2 + |M|(Y/2)w_3z_3 + |M|w_4z_4$. The same transformation was adopted for the application of HSH. The resulting mixed integer linear programs were solved using version 9.1 of AMPL/CPLEX (see Fourer et al., 2002), with default values where possible. All experiments were performed on a Pentium IV/1.8 GHz dual core processor, with 1 GB system memory.

Due to space limitation, instead of presenting the results of the experiments that were conducted, we present the most important conclusions reached from the analysis of these results. A first observation that can be made is that, besides problem size, the actual values of the problem parameters also have a strong influence on the total computational effort needed to reach an optimal solution in the case of MILP and HSH. This is supported by the fact that, even for the same problem size, a large variance is exhibited in the computational times of MILP and HSH. One of the parameters that have strong impact on this computational effort is the flight load (parameter S_{mt}). For example, for a particular problem size with $|M| = 3$, $|N_m| = 4$, $T = 10$ and a certain way for producing random instances, the average computational time over 10 instances that AMPL needed to reach an optimal solution was 36.97 seconds and the maximum time 148.22 seconds. When the average flight load was increased by 30%, the average computational time over 10 instances that AMPL needed to reach an optimal solution was 3512.4 seconds (almost 100 times larger) and the maximum time 8280.4 seconds. A similar behavior was observed for different problem sizes and parameter combinations. Therefore, as the flight load increases, the computational effort required by AMPL in order to reach an optimal solution seems to increase considerably. On the other hand, the computational effort of AFH1 and AFH2 on the same problem size does not vary significantly even when the exact values of the problem parameters vary considerably. Therefore, the actual values of the problem parameters do not have a strong influence on the computational effort of AFH1 and AFH2.

The total computational time needed in order to reach an optimal solution decreases significantly when HSH is used instead of MILP. For example, for the aforementioned problem size and the same average flight load as in the first set of experiments, the average computational time over the same 10 problem instances that HSH needed to reach an optimal solution was 0.98 seconds and the maximum time 2.21 seconds. On the other hand, the computational time required by AFH1 and AFH2 on practical problem sizes is negligible. This is also supported by the results of Table 1, which show the average and maximum computational time (in seconds) that AFH1 and AFH2 needed to find the optimal solution, over 10 random large scale problem instances. These results confirm that the computational effort needed for the execution of AFH1 and AFH2 is not very significant, even for problems whose size is exceptionally larger than the typical problem size arising in practical applications. For the same problem size, the computational effort required by AFH2 seems to be higher than that of AFH1, an observation which is in agreement with the computational complexity analysis of the previous section.

The significant computational savings of the three heuristic procedures come at a price, since the quality of the solutions that they produce is inferior to the quality of the solutions obtained by MILP. HSH exhibits a rather myopic behavior. It focuses on maximizing fleet availability in the initial periods first, but this may result in low availability over the next periods. On the other hand, a more conservative planning over the initial periods may, in some cases, result in higher availability over the entire planning horizon. Nevertheless, the solution obtained by HSH is quite satisfactory in most cases. Therefore, HSH can also be considered alternatively for obtaining an acceptable solution when the size of the problem prohibits its solution using MILP. In general, the number of periods of each smaller horizon has a strong effect on the quality of the obtained solution. An interesting conclusion that arises from this observation is that, since this is an on-going problem repeatedly solved in successive horizons, the length of the horizons for which the wing command issues the flight requirements has a strong impact on the long term availability of the unit. As the number of periods over which the command issues the flight requirements increases, the fleet availability of the unit is expected to increase, too.

Table 1. Computational times of AFH1 and AFH2 on large scale problems.

$\|M\|$	$\|N_m\|$	T	AFH1		AFH2	
			Avg	Max	Avg	Max
50	50	50	0.33	0.34	0.47	0.49
50	50	100	0.69	0.71	0.97	1.00
50	100	50	0.64	0.65	0.96	0.97
100	50	50	0.75	0.76	1.25	1.26
50	100	100	1.33	1.35	1.96	2.05
100	50	100	1.57	1.58	2.64	3.44
100	100	50	1.44	1.46	3.38	4.07
100	100	100	3.05	3.10	6.97	8.69

With respect to the quality of the solutions produced, MILP always produces the solution with the highest quality, which is additionally guaranteed to be nondominated (see Steuer, 1986). Moreover, the optimal values of the 4 objectives in the solutions produced by MILP appear to be close (approximately 2 to 3% on the average) to their ideal values (see Ehrgott, 2000). This is because the objectives of the model are not in direct conflict with each other, but there exists a certain degree of synergy among them. It should be noted though, that the best solutions are obtained when all of them are treated as objectives, otherwise the obtained solution may not be satisfactory with respect to the objective that was omitted. This is due to the fact that if we replace one of the objectives with a corresponding constraint, the model will only focus on providing a given lower bound for it, without any special concern for optimizing it.

The values of the 4 objectives in the solutions produced by AFH1 and AFH2 appear to be close (approximately 10% on the average) to the corresponding values of the solutions produced by MILP. This percentage did not remain constant but varied significantly between 2% and 20% in our experiments. The reason that this percentage is high in some cases is because the aim of MILP does not exactly coincide with the aim of AFH1 and AFH2, since MILP maximizes fleet availability, while AFH1 and AFH2 minimize the total deviation indices. As a result, even when a solution with higher fleet availability exists, AFH1 and AFH2 will prefer another one with lower fleet availability, if this leads to a lower deviation index. Despite the existence of such extreme peculiarities, AFH1 and AFH2 perform quite satisfactorily in general, mainly because they prevent bottlenecks in the maintenance station and ensure a smooth utilization of the maintenance station. Moreover, our computational experience shows that certain enhancements can be used to improve the quality of the solutions produced by these heuristics, which suggests that this is an area for future research with lots of potential.

The solutions produced by HSH, AFH1 and AFH2 are not nondominated in general. In some extreme cases, they may be infeasible, although the original problem is not. For HSH, this happens when it “exhausts” the system in order to optimize it in one of the smaller horizons, but then the solution obtained at the end of this horizon is inadequate and can not satisfy the flight requirements of the next one. On the other hand, the reason that this may sometimes happen in AFH1 and AFH2 is because they make the relevant decisions sequentially in each period; thus, it is possible to reach a point in later periods where some of the problem’s constraints can not be satisfied. In such cases, the user has to go back and revise his previous decisions in order to reach a feasible solution. In general, a careful design should address accordingly such peculiarities that may arise.

CONCLUSIONS

In this work, we proposed three heuristics for solving large FMP instances and we summarized some interesting results regarding their computational performance and the quality of the solutions that they produce. The results are very satisfactory, because they demonstrate that, under careful consideration, even large FMP instances can be handled quite effectively. Future research should be directed towards investigating the performance of these heuristics further, and towards exploring the extent to which they can be enhanced, or new better ones can be developed.

The aim of the present research is not to promote one particular tool that will be used exclusively for addressing the FMP problem, but rather to provide a set of tools that can be used collectively in order to address this important problem. The final decision on how to utilize these tools in order to get the most effective solution relies upon the user and depends on many parameters, such as the desired compromise between computational time and solution quality. In fact, one of the major goals of this research is to eventually produce an online decision support system for use by the HAF, which will enable the user to enter a specific instance of the FMP problem and combine the above tools in order to solve it.

APPENDIX

Proof of Proposition 1

Let λ_1, λ_2 and u_i (i = 1,…, N) be the nonnegative dual multipliers of constraints (2), (3) and (4), respectively. In addition to the original constraints of the problem, the KKT conditions are:

$$-2(y_{it} - is - x_{it}) - \lambda_1 + \lambda_2 + u_i \geq 0,\ i = 1,...,N \quad (5)$$

$$x_{it}[-2(y_{it} - is - x_{it}) - \lambda_1 + \lambda_2 + u_i] = 0,\ i = 1,...,N \quad (6)$$

$$\lambda_1(LS_{mt} - \sum_{i=1}^{N} x_{it}) = 0 \quad (7)$$

$$\lambda_2(\sum_{i=1}^{N} x_{it} - US_{mt}) = 0 \quad (8)$$

$$u_i(x_{it} - X_{ui}) = 0,\ i = 1,...,N \quad (9)$$

$$\lambda_1 \geq 0, \lambda_2 \geq 0, u_i \geq 0, i = 1,\ldots, N \quad (10)$$

Since L $\leq$ U always, 12 different arrangements of the quantities L, U, D and X exist. When X precedes L and this can not change using any ties present, the problem is clearly infeasible, since the flight requirements (constraint (2)) can not be satisfied, even when each aircraft is assigned its maximum possible flight time. In each of the remaining 8 cases, it is clear that the solution obtained from the application of the Procedure Sweep when the sum of the assigned aircraft flight times is equal to the second quantity in the arrangement, satisfies (2)-(4) and is therefore feasible. We show next that this solution also satisfies conditions (5)-(10), and is therefore optimal, too.

Case 1: The arrangement is {L,U,D,X} or {L,U,X,D}.
In this case, the sum of the assigned aircraft flight times in the obtained solution is equal to the second quantity in the arrangement, U. We partition the indices of the decision variables of this solution into 4 sets:
a) Set S_1 contains the indices of the variables x_{it} such that $x_{it} = 0 = X_{ui}$.
b) Set S_2 contains the indices of the variables x_{it} such that $x_{it} = 0 < X_{ui}$,
c) Set S_3 contains the indices of the variables x_{it} such that $0 < x_{it} = X_{ui}$,
d) Set S_4 contains the indices of the variables x_{it} such that $0 < x_{it} < X_{ui}$.

We set $\lambda_1 = 0$ and $\lambda_2 = 2(y_{it} - is - x_{it}), i \in S_4$. Note that the value of λ_2 is the same for any $i \in S_4$, since set S_4 contains the indices of all the variables which lie on the sweeping line at the current solution, and, as a result, have equal perpendicular distances from the diagonal, $y_{it} - is - x_{it}$. We also set $u_i = \max(2(y_{it} - is - x_{it}) - \lambda_2, 0)$ for $i \in S_1$, $u_i = 0$ for $i \in S_2 \cup S_4$ and $u_i = 2(y_{it} - is - x_{it}) - \lambda_2$ for $i \in S_3$. This last quantity is always nonnegative, since set S_3 contains the indices of the variables which were initially swept and later disengaged by the sweeping line because they reached their upper bound; therefore, the perpendicular distance of each of these points from the diagonal can not be smaller than the perpendicular distance from the diagonal of any point that lies on the sweeping line at the current solution. For $i \in S_1$ and $i \in S_3$, constraints (5) and (6) are clearly satisfied. Constraints (5) are clearly satisfied as an equality for $i \in S_4$; therefore, constraints (6) are satisfied, too. For $i \in S_2$, constraints (6) are clearly satisfied and constraints (5) are satisfied if $\lambda_2 \geq 2(y_{it} - is - x_{it})$, which is true, since set S_2 contains the indices of all the variables which have not been swept by the line yet; therefore, their perpendicular distance from the diagonal can not be larger than the perpendicular distance from the diagonal of any point that lies on the sweeping line at the current solution. Finally, constraints (7)-(10) are clearly satisfied, too. Hence, the current solution together with λ_1, λ_2 and u_i (i = 1,…, N) as the dual multipliers satisfies the KKT conditions and is therefore optimal.

Case 2: The arrangement is {L,D,U,X} or {L,D,X,U}.
In this case, the sum of the assigned aircraft flight times in the solution obtained is equal to the second quantity in the arrangement, D. We partition the indices of the decision variables of this solution into the same 4 sets as in Case 1. We set $\lambda_1 = \lambda_2 = 0$, $u_i = \max(2(y_{it} - is - x_{it}), 0)$ for $i \in S_1$, $u_i = 0$ for $i \in S_2 \cup S_4$ and $u_i = 2(y_{it} - is - x_{it})$, for $i \in S_3$. This last quantity is always nonnegative, since set S_3 contains the indices of the variables which were initially swept and later disengaged by the sweeping line because they reached their upper bound; therefore, since the sweeping line coincides with the diagonal at the current solution, the perpendicular distance of each of these points from the diagonal, $y_{it} - is - x_{it}$, can not be negative. For $i \in S_1$ and $i \in S_3$, constraints (5) and (6) are clearly satisfied. For $i \in S_2$, constraints (6) are clearly satisfied and constraints (5) are satisfied if $-2(y_{it} - is - x_{it}) \geq 0$, which is true, since set S_2 contains the indices of the variables which have not been swept by the line yet;

therefore, since the sweeping line coincides with the diagonal at the current solution, each of these points has nonpositive perpendicular distance from the diagonal. Set S_4 contains the indices of the variables which lie on the sweeping line at the current solution. Since the sweeping line coincides with the diagonal at this solution, the perpendicular distance of each of these variables from the diagonal, ($y_{it} - is - x_{it}$), is equal to 0. As a result, constraints (5) and (6) are also satisfied for $i \in S_4$. Finally, constraints (7)-(10) are clearly satisfied, too. Hence, the current solution together with λ_1, λ_2 and u_i ($i = 1,\ldots, N$) as the dual multipliers satisfies the KKT conditions and is therefore optimal.

Case 3: The arrangement is {D,L,U,X} or {D,L,X,U}.
In this case, the sum of the assigned aircraft flight times in the solution obtained is equal to the second quantity in the arrangement, L. We partition the indices of the decision variables of this solution into the same 4 sets as in Cases 1 and 2. We set $\lambda_2 = 0$ and $\lambda_1 = -2(y_{it} - is - x_{it}), i \in S_4$. Note that the value of λ_1 is the same for any $i \in S_4$, since set S_4 contains the indices of all the variables which lie on the sweeping line at the current solution, and, as a result, have equal perpendicular distances from the diagonal, $y_{it} - is - x_{it}$. Additionally, each of these distances is nonpositive, since the fact that D appears first in the arrangement implies that the sweeping line can not lie above the diagonal at the current solution; therefore, the value of λ_1 is nonnegative. We also set $u_i = \max(2(y_{it} - is - x_{it}) + \lambda_1, 0)$ for $i \in S_1$, $u_i = 0$ for $i \in S_2 \cup S_4$ and $u_i = 2(y_{it} - is - x_{it}) + \lambda_1$ for $i \in S_3$. This last quantity is always nonnegative, since set S_3 consists of all the variables which were initially swept and later disengaged by the sweeping line because they reached their upper bound; therefore, since the sweeping line can not lie above the diagonal at the current solution, the perpendicular distance of each of these points from the diagonal can not be smaller than the perpendicular distance from the diagonal of any point that lies on the sweeping line. For $i \in S_1$ and $i \in S_3$, constraints (5) and (6) are clearly satisfied. Constraints (5) are clearly satisfied as an equality for $i \in S_4$; therefore, constraints (6) are satisfied, too. For $i \in S_2$, constraints (6) are clearly satisfied and constraints (5) are satisfied if $-\lambda_1 \geq 2(y_{it} - is - x_{it})$, which is true, since set S_2 contains the indices of all the variables which have not been swept by the line yet; therefore, since the sweeping line can not lie above the diagonal at the current solution, the perpendicular distance of each of these points from the diagonal can not be larger than the perpendicular distance from the diagonal of any point that lies on the sweeping line. Finally, constraints (7)-(10) are clearly satisfied, too. Hence, the current solution together with λ_1, λ_2 and u_i ($i = 1,\ldots, N$) as the dual multipliers satisfies the KKT conditions and is therefore optimal.

Case 4: The arrangement is {L,X,U,D} or {L,X,D,U}.
In this case, the sum of the assigned aircraft flight times in the solution obtained is equal to the second quantity in the arrangement, X. We partition the indices of the decision variables of this solution into 2 sets:
a) Set S_1 contains the indices of the variables x_{it} such that $x_{it} = 0 = X_{ui}$,
b) Set S_2 contains the indices of the variables x_{it} such that $0 < x_{it} = X_{ui}$.

We set $\lambda_1 = \lambda_2 = 0$, $u_i = \max(2(y_{it} - is - x_{it}), 0)$ for $i \in S_1$ and $u_i = 2(y_{it} - is - x_{it})$ for $i \in S_2$. This last quantity is always nonnegative, since set S_2 contains the indices of the variables which were initially swept and later disengaged by the sweeping line because they reached their upper bound; therefore, since the sweeping line has not reached the diagonal yet, their perpendicular distance from the diagonal, $y_{it} - is - x_{it}$, is nonnegative. Constraints (5)-(10) are clearly satisfied for $i \in S_1 \cup S_2$. Hence, the current solution together with λ_1, λ_2 and u_i ($i = 1,\ldots, N$) as the dual multipliers satisfies the KKT conditions and is therefore optimal. □

Proof of Proposition 2

The initialization command in the first line of Step 0 requires time O(|M|). Each command inside the nested for-loop of Step 0 requires time O(1) and is repeated at most $|M|N_{max}$ times. Each of the next two commands outside the nested for-loop requires time $O(N_{max}\log N_{max})$ and is repeated |M| times. Therefore, the total time that Step 0 requires is $O(|M|) + O(|M|(N_{max}+ N_{max}\log N_{max}) = O(|M|N_{max}\log N_{max})$. The initialization command in the first line of Step 1 requires time $O(|M|N_{max})$. The check in the while command of Step 1 requires time O(1), since the existence of grounded aircraft is already known. Finding k requires time O(|M|) and finding l requires time O(1), since the grounded aircraft of each squadron are already sorted. The if-else clause inside the while-loop of Step 1 requires time O(1). Since the while-loop of Step 1 is repeated at most C times (once for each grounded aircraft), Step 1 requires time $O(|M|N_{max}) + O(C|M|)$ in total. The initialization command in the first line of Step 2 requires time $O(|M|N_{max})$. The check in the while command of Step 2 requires time O(1). Finding k requires time O(|M|) and finding l requires time O(1), since the available aircraft of each squadron are already sorted. The if-else clause inside the while-loop of Step 2 requires time O(1). Since the while-loop of Step 2 is repeated at most O(|M|+C) times (when C aircraft that belong to the squadron with the minimum priority index are grounded), Step 2 requires time $O(|M|N_{max}) + O((|M|+C)|M|)$ in total. In Step 3, solving $|M|$ times the problem defined by (1)-(4) requires time $O(|M|N_{max})$ in total. Finally, updating the system status for period t+1 in Step 4 requires time $O(|M|N_{max})$. Since Step 0 is executed once and each of Steps 1-4 is repeated T times, the total time required by AFH1 is $O(|M|N_{max}\log N_{max}) + O(T(|M|N_{max} + C|M| + |M|N_{max} + (|M|+C)|M| + |M|N_{max} + |M|N_{max})) = O(|M|N_{max}\log N_{max}) + O(T \max(|M|N_{max}, C|M|, |M|^2)) = O(|M| \max(N_{max}\log N_{max}, TN_{max}, TC, T|M|))$. □

Proof of Proposition 3

The initialization command in the first line of Step 0 requires time $O(|M|)$. Each command inside the nested for-loop of Step 0 requires time O(1) and is repeated at most $|M|N_{max}$ times. Each of the next two commands outside the nested for-loop requires time $O(N_{max}\log N_{max})$ and is repeated $|M|$ times. The two commands outside the for-loops of Step 0 sort the available and grounded aircraft of the wing and require time $O(N\log|M|)$, since the aircraft of each squadron are already sorted. Therefore, the total time that Step 0 requires is $O(|M|) + O(|M|(N_{max}+ N_{max}\log N_{max})) + O(N\log|M|) = O(|M|N_{max} \max(\log N_{max}, \log|M|))$. The initialization command in the first line of Step 1 requires time $O(|M|N_{max})$. The check in the while command of Step 1 requires time O(1), since the existence of grounded aircraft is already known. Finding k and l requires time O(1), since the grounded aircraft of the wing are already sorted. The if-else clause inside the while-loop of Step 1 requires time O(1). Since the while-loop of Step 1 is repeated at most C times (once for each grounded aircraft), Step 1 requires time $O(|M|N_{max}) + O(C) = O(|M|N_{max})$ in total. The initialization command in the first line of Step 2 requires time $O(|M|N_{max})$. The check in the while command of Step 2 requires time O(1). Finding k and l requires time O(1), since the available aircraft of the wing are already sorted. The if-else clause inside the while-loop of Step 2 requires time O(1). Since the while loop of Step 2 is repeated at most $O(|M|+C)$ times (when C aircraft that belong to the squadron considered last are grounded), Step 2 requires time $O(|M|N_{max}) + O(|M|+C) = O(|M|N_{max})$ in total. In Step 3, solving $|M|$ times the problem defined by (1)-(4), requires time $O(|M|N_{max})$ in total. Finally, updating the system status for period t+1 in Step 4 requires time $O(|M|N_{max})$. Then, sorting the available aircraft of the wing requires time $O(N\log|M|)$, since the available aircraft of each squadron are already sorted. Therefore, the total time required by Step 4 is $O(|M|N_{max}+ N\log|M|) = O(|M|N_{max}\log|M|)$. Since Step 0 is executed once and each of Steps 1-4 is repeated T times, the total time required by AFH2 is $O(|M|N_{max} \max(\log N_{max}, \log|M|)) + O(T(|M|N_{max} + |M|N_{max} + |M|N_{max} + |M|N_{max}\log|M|)) = O(|M|N_{max} \max(\log N_{max}, T\log|M|))$. □

ACKNOWLEDGEMENTS

George Kozanidis is a recipient of a research grant from the State Scholarships Foundation of Greece, which partially supported this work.

REFERENCES

Bazaraa, M.S., Sherali, H.D. and Shetty, C.M. (2006). *Nonlinear Programming: Theory and Algorithms*. Wiley-Interscience, New York, USA.

Brucker, P. (1984). "An O(n) Algorithm for Quadratic Knapsack Problems". *Operations Research Letters*, 3(3), 163-166.

Ehrgott, M. (2000). *Multicriteria Optimization, Lecture Notes in Economics and Mathematical Systems*. Springer-Verlag, Berlin, Germany.

Fourer, R., Gay, D.M. and Kernighan, B.W. (2002). *AMPL: A Modeling Language for Mathematical Programming*. Duxbury Press, Brooks-Cole Publishing Company.

Kozanidis, G. and Skipis A. (2006). "Flight and Maintenance Planning of Military Aircraft for Maximum Fleet Availability: A Biobjective Model". *18th International Conference on Multiple Criteria Decision Making*. Chania, Greece, http://www.dpem.tuc.gr/fel/mcdm2006/Papers/Kozanidis_Skipis.pdf.

Steuer, R.E. (1986). *Multiple Criteria Optimization*. Wiley, New York.

U.S. DoA. (2000). "Field Manual No. 3-04.500: Army Aviation Maintenance (Appendix D: Maintenance Management Tools)", U.S. Department of the Army, Washington, DC. http://www.globalsecurity.org/military/library/policy/army/fm/3-04-500/.