Proceedings of the 2011 Winter Simulation Conference
S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, eds.

SCHEDULING FIGHTER AIRCRAFT MAINTENANCE WITH REINFORCEMENT LEARNING

Ville Mattila
Kai Virtanen

Systems Analysis Laboratory
Aalto University School of Science
P.O. Box 11100, FIN-00076 Aalto, Finland

ABSTRACT

This paper presents two problem formulations for scheduling the maintenance of a fighter aircraft fleet under conflict operating conditions. In the first formulation, the average availability of aircraft is maximized by choosing when to start the maintenance of each aircraft. In the second formulation, the availability of aircraft is preserved above a specific target level by choosing to either perform or not perform each maintenance activity. Both formulations are cast as semi-Markov decision problems (SMDPs) that are solved using reinforcement learning (RL) techniques. As the solution, maintenance policies dependent on the states of the aircraft are obtained. Numerical experiments imply that RL is a viable approach for considering conflict time maintenance policies. The obtained solutions provide knowledge of efficient maintenance decisions and the level of readiness that can be maintained by the fleet.

1 INTRODUCTION

Fighter aircraft are maintained regularly in order to guarantee that their operating condition meets the performance requirements of their intended use. The maintenance activities of the aircraft are typically time-consuming and the available maintenance resources are limited (Mattila, Virtanen, and Raivio 2008). In normal conditions, the activities are scheduled in advance to preserve sufficient level of readiness in terms of aircraft availability, which is defined as the fraction of mission-capable aircraft that are not being maintained or repaired. Conflict situations, on the other hand, involve increased level of uncertainty through battle damage repairs, for instance. Although some maintenance activities may be momentarily discarded in order to meet the demand for flight missions, others must be performed to keep the aircraft in sufficiently good operating condition. Maintenance scheduling is still needed to preserve the readiness of the fleet but the requirements for scheduling are different.

This paper investigates two formulations for scheduling the maintenance of fighter aircraft during conflict situations. The first formulation involves the sustainment of a high level of readiness such that the fleet can respond to actions undertaken by an opponent with as many aircraft as possible at any given time instant. To reach this goal, the average availability of aircraft is maximized by choosing when to start each maintenance activity. The second formulation involves a situation where a given target level of aircraft availability must be maintained during a limited time period. The requirement can be due to anticipated actions of an opponent or the need to prepare for a future operation. In this formulation, maintenance activities may or may not be performed. The objectives are to reach the target level of availability while maximizing the number of performed maintenance activities so that the operating condition of the aircraft is sacrificed as little as possible. Since the aircraft that are being maintained are considered unavailable these objectives are competing.

The formulations described above are modeled as semi-Markov Decision Problems (SMDPs) (see, e.g., Gosavi 2003). In an SMDP, a system is represented by a Markov chain whose transition probabilities

and transition times depend on an action. The choice of the action that results in a given state transition incurs a reward or a cost depending on the context. The problem is to determine a policy, a mapping from the states of the system to the available actions, that maximizes the obtained reward or minimizes the cost over a given time period. In particular, the suitability of applying reinforcement learning (RL) techniques (Bertsekas and Tsitsiklis 1996, Kaelbling, Littman, and Moore 1996, Sutton and Barto 1998, Gosavi 2003) to solve the above problem formulations is studied. RL entails a variety of techniques for learning optimal behavior through trial and error in dynamic environments, including SMDPs. Here, RL is utilized to learn optimal maintenance policies by presenting actions to a simulation model of aircraft usage and maintenance and by observing the resulting simulated state transitions.

By using RL to solve the maintenance scheduling formulations, state dependent maintenance policies are obtained. In the first formulation, the state of the system involves the current number of aircraft being maintained or repaired and the time until the next maintenance activity must be started. In the second formulation, the state consists of the number of aircraft being maintained or repaired as well as the stage of the time period under consideration. State dependent policies are required in conflict situations where the exact timing of the maintenance activities can not be determined in advance due to high level of uncertainty. The obtained policies offer maintenance decision-makers (DMs) information on efficient maintenance decisions in different scenarios. Moreover, simulations of aircraft usage and maintenance with the optimal policies offer the DMs information on the achievable level of readiness.

Earlier studies on maintenance scheduling of fighter aircraft have mainly considered normal operating conditions where the starting times of the maintenance activities are determined in advance (Mattila and Virtanen 2006, Kozanidis, Liberopoulos, and Pitsilkas 2010). Previously, the possibility of applying RL for the scheduling of condition based maintenance of fighter aircraft has been studied in (Tang et al. 2006) and for the scheduling of periodic maintenance during normal operating conditions in (Mattila 2007). Other applications of RL that involve military aircraft include spare parts management (Simao and Powell 2009) as well as the planning of airlift operations (Wu, Powell, and Whisman 2009). The authors are not, however, aware of earlier studies that would apply RL to the maintenance scheduling problems considered in this paper.

The rest of the paper is organized as follows. In Section 2, a brief introduction to RL as a method for solving SMDPs is given. In Section 3, the two problem formulations are given along with the algorithms used to solve them. Section 4 presents the results of numerical experiments conducted for the formulations. Concluding remarks are given in Section 5.

2 REINFORCEMENT LEARNING

2.1 SMDP

In order to discuss SMDPs, a related problem, namely Markov Decision Problem (MDP), is described (e.g., Gosavi 2003). An MDP involves an underlying Markov chain $X = \{X_n : n \in \mathbb{N}, X_n \in S\}$ where X_n denotes the system state at the nth decision instant, S a finite set of states and $\mathbb{N}$ the set of integers. At each decision instant, an action $a \in A(i)$ is selected where $A(i)$ denotes the set of actions available at state $X_n = i$. Under a, the probability that $X_{n+1} = j$ is denoted with $p(i,a,j)$. The states are assumed Markovian, i.e., the transition probabilities are only dependent on the current state as well as the action and not on the past history of the states or the actions. The reward of moving from state i to j as a result of selecting action a is denoted with $r(i,a,j)$. The objective is to determine the policy that associates an action to each state of the system and maximizes the rewards during some time period.

In MDPs, each state transition results from the selection of an action. Moreover, the time spent in each transition equals unity. In SMDPs, these assumptions are relaxed. Decisions are made at discrete time instants whose intervals follow general probability distributions. The system state can change multiple times between decision instants. In particular, state transition probabilities $p(i,a,j)$ and transition rewards $r(i,a,j)$ are now associated with state transitions from one decision instant to another, not necessarily two

subsequent system states. The same applies to state transition times, denoted with $t(i,a,j)$. It should be noted that here the transition rewards and times are random. These quantities are treated as deterministic in describing the concepts related to RL, whereas realizations of the random values are utilized in solving the maintenance scheduling problems with RL algorithms.

2.2 Reinforcement Learning for SMDPs

MDPs and SMDPs can be solved using dynamic programming (DP) (Bertsekas 1995). DP is based on calculating the optimal value of each state, i.e., reward of choosing the optimal policy in that state and following the optimal policy from then on. It can be shown that these values, denoted with $J^*(i)$, are the solution of a specific system of equations referred to as the Bellman equation. For an SMDP where the average reward per time unit is maximized, the Bellman optimality equation is (Bertsekas and Tsitsiklis 1996)

$$J^*(i) = \max_{a \in A(i)} \sum_{j=1}^{|S|} p(i,a,j) \left[r(i,a,j) - \rho^* t(i,a,j) + J^*(j) \right], \forall i \in S, \tag{1}$$

where ρ^* is the average reward of the optimal policy. The optimal policy is obtained by choosing in each state the greedy action with respect to the value function solved from the Bellman equation.

DP requires the knowledge of the state transition probabilities, rewards, and times corresponding to each state and action for the solution of the value function. RL, in turn, is an approach for the approximate solution of the value function that does not utilize an explicit description of these probabilities, rewards, and times (Gosavi 2003). Instead, the value function is solved, or learned, by repeatedly experimenting different actions in different states. The state transitions are now the result of the logic of the simulation model representing the system under consideration.

RL associates a value, referred to as Q-factor, for each state-action pair instead of each state. The Bellman equation of an average reward SMDP, written in terms of the Q-factors, is (Gosavi 2003)

$$Q(i,a) = \sum_{j=1}^{|S|} p(i,a,j) \left[r(i,a,j) - \rho^* t(i,a,j) + \max_{b \in A(j)} Q(j,b) \right], \forall (i,a).$$

The optimal policy is obtained by selecting in each state the action corresponding to the largest Q-factor of the state.

The Bellman equation is used in determining the optimal values of the Q-factors through iterative updating. The update based on a single state transition can be written as (Gosavi 2003)

$$Q(i,a) \leftarrow Q(i,a) + \alpha \left[r(i,a,j) - \rho t(i,a,j) + \max_{b \in A(j)} Q(b,j) - Q(i,a) \right]. \tag{2}$$

The arrow denotes the assignment of a new value, $\alpha \in [0,1]$ a step size, and ρ an estimate for the average reward of the optimal policy.

In order to determine optimal values of the Q-factors, the step size α must remain sufficiently small. Typically, the step size is decreased as the number of simulated state transitions increases. Different methods for the updating are discussed in (Gosavi 2003). Additionally, each state-action pair must be visited sufficiently often in order to determine the Q-factors accurately (Gosavi 2003). An approach that is often utilized is the ε-greedy strategy (Sutton and Barto 1998) which selects a greedy action, i.e., one that maximizes the Q-factor of the state under consideration, with probability $1-\varepsilon$. Otherwise, a random action that is not greedy is selected. ε is referred to as the exploration probability. It is decreased as more state transitions are simulated such that the selected actions are gradually moved toward the learned policy. The role of step sizes and exploration are discussed further in (Sutton and Barto 1998).

One additional aspect of RL algorithms involves the concept of temporal difference (TD) which is defined for the average reward SMDP as (Bertsekas and Tsitsiklis 1996)

$$\delta = r(i,a,j) - \rho t(i,a,j) + \max_{b \in A(j)} Q(b,j) - Q(i,a), \tag{3}$$

i.e., the term in the square brackets in Equation (2). The temporal difference is the difference between the current estimate of a Q-factor and a new observation made on it. It can be seen that the update in Equation (2) involves a single state transition, its associated reward, and time as well as the next state. A method based on Equation (2) could therefore be referred to as a one-step TD method. The range of the observed transitions could, however, also span a single transition to infinity. Multi-step TD methods are typically parametrized with the parameter $\lambda \in [0,1]$ that indicates how far in the future the state transitions keep affecting the updated Q-factor. Equation (2) with the consideration of several future transitions is (Bertsekas and Tsitsiklis 1996)

$$Q(i_k,a_k) \leftarrow Q(i_k,a_k) + \alpha \sum_{q=k}^{\infty} \lambda^{q-k} \delta_k, \tag{4}$$

where the value of each variable at a given decision instant is indicated with the subscript k. More weight is given to nearby transitions and less to ones that are far ahead. Several transitions need to be considered in the update if the rewards of choosing a given action in a given state are experienced only several state transitions later. Using update of the form presented in Equation (4) with $\lambda > 0$ can improve the convergence of the RL algorithm in such cases. Both algorithms that are utilized to solve the problem formulations presented in this paper are multi-step TD methods.

It should be noted that the introduction given here is for an SMDP in which the average reward per time unit is maximized. The basic concepts are similar in the case of maximizing the total reward over a time period although the equations for updating Q-factors are slightly different (Gosavi 2003).

3 PROBLEM FORMULATIONS

This section describes the problem formulations for maintenance scheduling of a fleet of fighter aircraft operating under conflict conditions. The common features of the formulations are first discussed after which the unique features of each one are described. Following the description of each formulation is the description of an RL algorithm that is used in its solution.

3.1 Description of the Problem

The maintenance of fighter aircraft is performed at regular intervals measured in flight hours. A certain tolerance is additionally allowed, i.e., there is a feasible window of elapsed flight hours from the previous completion of the maintenance during which the maintenance must be started. The maintenance scheduling problem is to decide when to start the maintenance activities of the aircraft that have reached the feasible maintenance window.

The states of the aircraft evolve through the following events:

- An aircraft starts or completes a flight mission.
- Failure or battle damage repair of an aircraft is started.
- Maintenance of an aircraft is started.
- The repair or maintenance of an aircraft is completed.

It is assumed that the available aircraft form a first-in-first-out queue from which individual aircraft are drawn to perform flight missions. Failures and battle damage occur during the flight missions at exponentially distributed time intervals. If a failure or a battle damage is sustained, the aircraft is taken to

repair after the mission. Maintenance needs are also assessed at the end of the mission. The maintenance facility that performs the maintenance activities and repairs has a limited capacity. If the number of aircraft requiring maintenance or repair exceeds this capacity, a first-in-first-out queue is formed.

The decision instants of the maintenance scheduling problem correspond to the time instants when an aircraft whose elapsed flight hours have reached the feasible maintenance window completes a flight mission. Moreover, the decision is only made for the aircraft with the highest number of elapsed flight hours since this aircraft requires maintenance most urgently. The available actions at each state of the system are to *start* or *delay* the maintenance.

An important question in applying RL to the problem setting described above is the representation of the system state that is utilized. Ideally, the representation includes all information needed to determine the next state transition as well as the associated reward and is said to be Markovian. However, a representation that allows the prediction of the future state and the reward under different actions with sufficient accuracy and allows the RL algorithm to learn the optimal policy can be used (Sutton and Barto 1998). In the maintenance scheduling problem formulations, state representations are identified that include as few variables as possible but still allow optimal decisions to be made. The significance of the state representation is discussed at length in (Sutton and Barto 1998).

3.2 Formulation 1

3.2.1 Maximization of Average Availability

The first formulation involves a situation where the readiness of the aircraft fleet needs to be kept at a high level for an extended time period. This situation is faced when some activity from an opponent is anticipated but there is no prior information of when such activity takes place. The objective of maintenance scheduling is then to maximize the average availability of aircraft. Since there is no natural limit for the time period under consideration, the corresponding SMDP becomes that of maximizing average availability over infinite horizon.

It is assumed that the maintenance of aircraft is performed within feasible maintenance windows. Thus, if the elapsed flight hours of an aircraft exceed the feasible window, the aircraft is automatically taken to maintenance. This is the primary distinction to the second formulation of the scheduling problem to be described later. In addition, the demand for flight missions, the failure rate of aircraft, mission durations as well as maintenance capacity are assumed to remain unchanged over time.

The system state is represented by $s = (m,h)$, where m is the number of aircraft being currently maintained, repaired, or queued in the maintenance facility and h the number of flight hours until the aircraft with the maximum number of elapsed flight hours exceeds the feasible maintenance window. The value of h is appropriately discretized.

The reward of moving from state s_k at the kth decision instant to state s_{k+1} under action a is defined as

$$r(s_k, a, s_{k+1}) = \int_{\tau_k}^{\tau_{k+1}} 1 - \frac{m(t)}{M} dt,$$

where $m(t)$ denotes the number of aircraft in the maintenance facility at time t, τ_k the time of the kth decision instant and M the total number of aircraft in the fleet.

3.2.2 RL Algorithm for Formulation 1

The algorithm used for solving Formulation 1 is called λ-SMART (Gosavi, Bandla, and Das 2002). The algorithm learns the optimal average reward appearing in the Bellman optimality equation (1) at the same time while learning the optimal Q-factors. Its steps are as follows:

0. Choose a value for λ in the interval $[0,1]$. Set Q-factors $Q(s,a) = 0$ as well as traces $G(s,a) = 0$ and visit factors $V(s,a) = 0$ for all (s,a). Set the cumulative reward $c_t = 0$ and the total time $t_t = 0$.

Choose values for the initial step size α and the exploration rate ε. Set the estimate of the optimal average reward $\rho = 0$. Set the index of the decision making instant $k = 0$. Choose a very large total simulation time T. Choose an arbitrary initial state s_1.

1. Let $a_k^{\max}$ be the greedy action with respect to the Q-factors of the current state , i.e., $a_k^{\max} = \arg\max_{a \in A} Q(s_k, a)$. Set $a_k = a_k^{\max}$ with probability $1 - \varepsilon$. Else, choose an exploratory action as a_k.
2. Simulate the system until the next decision instant. Denote the associated transition time with $t(s_k, a_k, s_{k+1})$ and the reward with $r(s_k, a_k, s_{k+1})$. Set $V(s_k, a_k, s_{k+1}) \leftarrow V(s_k, a_k, s_{k+1}) + 1$.
3. If a_k is a greedy action, then set
   $c_t \leftarrow c_t + r(s_k, a_k, s_{k+1})$,
   $t_t \leftarrow t_t + t(s_k, a_k, s_{k+1})$,
   $\rho \leftarrow c_t / t_t$.
4. Calculate the temporal difference δ_k associated to the transition according to Equation (3). Set the trace value of the current state-action pair $G(s_k, a_k) = 1$. Then, update all elements of Q and G:
   $Q(s,a) \leftarrow Q(s,a) + (\alpha / V(s,a)) \delta_k G(s,a)$,
   $G(s,a) \leftarrow \lambda G(s,a)$.
5. If $\sum_{i=1}^{k} t(s_k, a_k, s_{k+1}) \leq T$, set $k \leftarrow k+1$ and return to Step 1. Else return $\pi(s) = \arg\max_{a \in A} Q(s,a)$ as the generated optimal policy.

The traces $G(s,a)$ of the algorithm are simply a way to accomplish incrementally the updating of the Q-factors when several temporal differences are involved. The Q-factors can thus be updated immediately after each state transition without waiting until all of the transitions affecting the update are simulated.

While the implementation of the λ-SMART algorithm is otherwise standard in this paper, a noteworthy exception is the strategy for exploring the state-action pairs. It turns out that guaranteeing continued exploration, i.e., application of alternative actions in each state, can not be accomplished satisfactorily using a basic ε-greedy strategy. Consider that the value of the state variable $h = \tilde{h}$. Before the aircraft under consideration is forced to undergo maintenance, the action *delay* has to be chosen $\tilde{h}$ times. If the exploratory action is always selected randomly, the probability of observing a forced start of maintenance may remain very small during the early stages of learning. The Q-factors corresponding to the states where h has a small value may thus be based on insufficient number of observations and remain inaccurate.

To overcome the difficulty of exploration, the following approach is taken. In the place of exploratory actions in the ε-greedy strategy, a randomly chosen threshold policy is used. In this policy, whenever the value of the state variable h is less than a random integer drawn uniformly from the set $\{1, 2, ..., H+1\}$, where H is the maximum value of h, *start* is selected. The threshold policy is kept constant for several transitions and then updated. This approach makes it possible to repeatedly visit the different state-action pairs during exploratory actions. The exploration probability is slowly decreased in order to guarantee the convergence of the algorithm.

3.3 Formulation 2

3.3.1 Maintaining Availability above Target Level

Formulation 2 involves a situation where a specific target level of aircraft availability needs to be maintained. The need may be due to an operation to be carried out by the fleet or due to anticipated activity of an opponent. In contrast to the first formulation, maintenance of aircraft can be delayed beyond the feasible maintenance window. Then, the objectives are to maintain aircraft availability above the target level and to perform as many maintenance activities as possible such that the operating condition of the aircraft is not adversely affected. As emphasized before, the aircraft being maintained are not considered available and, thus, the above objectives are competing.

The time period under consideration is assumed finite. The demand for flight missions, the failure and battle damage rates of aircraft, mission durations as well as maintenance capacity may change over time,

whereas they were assumed constant in Formulation 1. Therefore, time is included in the system state (Gosavi 2003). The feasible time window for maintenance does not play a role anymore so the elapsed flight hours of the aircraft are not included. The system state is thus represented by $s = (m,d)$, where d denotes the day of the time period under consideration and m is defined as before.

Two types of rewards are associated to the state-action pairs in order to accommodate the twofold objectives of the problem. First, a positive immediate reward of r_m units is obtained, if the maintenance of the aircraft under consideration is started. Second, a negative reward of r_a units per one unit of time and availability incurs whenever the availability of the aircraft decreases below the target level. The target level of availability is denoted with β. The reward of selecting action a in state s_k and moving to state s_{k+1} as a result is expressed as

$$r(s_k,a,s_{k+1}) = r_m \mathbf{1}(a = \{start\}) + r_a \int_{\tau_k}^{\tau_{k+1}} \max\left[0, \beta - \left(1 - \frac{m(t)}{M}\right)\right] dt,$$

where $\mathbf{1}(\cdot)$ is an indicator function and other notation are defined as before. The optimization criterion is the sum of the above rewards over a finite time horizon. It is worth noting the meaning of using the above described reward for learning the optimal policy. The relative magnitudes of the rewards r_m and r_a determine how many units of unavailability is tolerated to perform one maintenance activity. The rewards thus implicitly weigh the objectives of the problem and the values of r_m and r_a should reflect the desired trade-off between maintaining aircraft and maintaining high availability.

3.3.2 RL Algorithm for Formulation 2

The algorithm for solving Formulation 2 is only briefly discussed, since several of its steps are similar to λ-SMART. The algorithm that is used here is called SARSA(λ) (Sutton and Barto 1998). It uses the following definition for temporal differences

$$\delta = r(i,a,j) + Q(b,j) - Q(i,a),$$

where b denotes the action that is selected in the next state j. Thus, the updating of the Q-factors is different compared λ-SMART (see Equation (3)). Step 3 of λ-SMART in which the average reward is estimated is not needed. A basic ε-greedy strategy for exploring state-action pairs is utilized in SARSA(λ) and several independent simulation replications are performed instead of a single long run. Otherwise, the steps of the algorithm are similar to λ-SMART.

4 NUMERICAL EXPERIMENTS

In this section, the results of numerical experiments conducted with the two problem formulations are presented. The inputs of the simulation model of aircraft usage and maintenance, utilized for learning optimal maintenance policies, are the same for both formulations (Table 1). The inputs are illustrative and based on the operation of training aircraft during normal conditions (Mattila, Virtanen, and Raivio 2008).

Maintenance durations follow the Gamma distribution with shape parameter 2 and scale parameter 40. Repair durations are also Gamma distributed. Times between flight missions as well as failures are exponentially distributed with means of 0.75 hours and 12 flight hours. Flight durations follow the normal distribution with a mean of 0.75 hours and a standard deviation of 0.25. Battle damage repairs are not included since they would be modeled in the same way as the failure repairs. Maintenance capacity indicates the maximum number of aircraft that can be maintained or repaired simultaneously.

4.1 Results for Formulation 1

The values for the parameters of the λ-SMART algorithm in Formulation 1 are presented in Table 2. These values were found to be effective through several test runs.

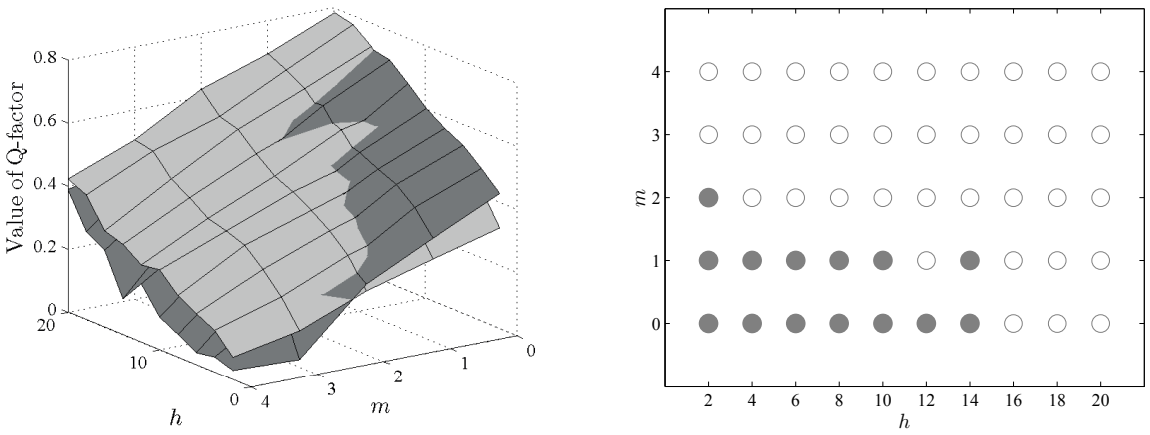
Table 1: The inputs of the simulation model of aircraft usage and maintenance.

Number of aircraft	16
Feasible maintenance window (flight hours)	$[40,60]$
Duration of maintenance (hours)	Gamma(2,40)
Time between flight missions (hours)	Expo(0.75)
Duration of a flight mission (hours)	Norm(0.75,0.25)
Time between failures (flight hours)	Expo(12)
Duration of a failure repair (hours)	Gamma(2,4.5)
Maintenance capacity (number of aircraft)	3

Table 2: The parameter values of the λ-SMART algorithm.

Parameter	Description	Value
λ	Discount factor of temporal differences	0.9
α	Initial step size	0.5
ε	Initial exploration rate	1
T	Total length of simulation (days)	$3 \cdot 10^4$

The Q-factors obtained by running the λ-SMART algorithm and the corresponding optimal policy are shown in Figure 1. In Figure 1(a), the Q-factors of all state-action pairs are shown. The optimal policy that selects the action with the maximum Q-factor at each state is depicted in Figure 1(b). It should be noted that Q-factors are only determined for $m = \{0,1,2,3,4\}$, since the capacity of the maintenance facility is 3. The Q-factors for $m = 4$ aggregate the Q-factors for system states in which there are 4 or more aircraft in maintenance.



(a) Q-factors

(b) Optimal policy

Figure 1: The learned Q-factors and the corresponding optimal policy. The values of the Q-factors for action *start* are depicted with dark gray and those for action *delay* with light gray. In the optimal policy, a filled circle indicates that action *start* is selected whereas a circle with no fill indicates that *delay* is selected.

In the optimal policy, maintenance is started when there are no aircraft in the maintenance facility or when a single aircraft is in maintenance and there are still a number of flight hours until the feasible maintenance window is exceeded. This policy appears valid as maintenance resources are reserved for failure repairs. With higher levels of congestion, maintenance is not started until the feasible maintenance

window is exceeded. The reason is that delaying the maintenance makes it possible to complete ongoing maintenance activities or repairs and, thus, the workload of the maintenance facility may be kept even.

When running the λ-SMART algorithm repeatedly, the Q-factors appear to converge consistently to a policy that is similar to the one shown in Figure 1(b). The Q-factors of some states are, however, very close to one another. Some variability in the obtained policy therefore appears from one run of the RL algorithm to another. This explains, for instance, that in Figure 1(b) the optimal action with $(m,h) = (1,12)$ is *delay* although the optimal actions in the neighboring states $(m,h) = (1,10)$ and $(m,h) = (1,14)$ are *start*.

It should be noted that the optimal policy is quite close to a policy in which the decisions to start maintenance depend solely on the current level of congestion in the maintenance facility. It is straightforward to simulate each such policy and select the most efficient one. Figure 2 compares the average availability produced by the best of such policies and the optimal policy solved with RL. The results are based on 1000 independent simulation replications, each warmed up to the steady state. The average availability is higher for the optimal policy. For a maintenance DM, however, the consideration of the worst rather than the average case is more significant. Figure 2 therefore also compares the lower percentiles of observed availabilities, where differences are more noticeable. The 5th percentiles indicate the level of availability that is reached at an arbitrary time instant with probability of 0.95. The difference in the 5th percentiles is slightly over 0.05 in favor of the optimal policy produced with RL for a large part of the observed time period. The optimal policy therefore performs better in preserving the readiness of the aircraft fleet.

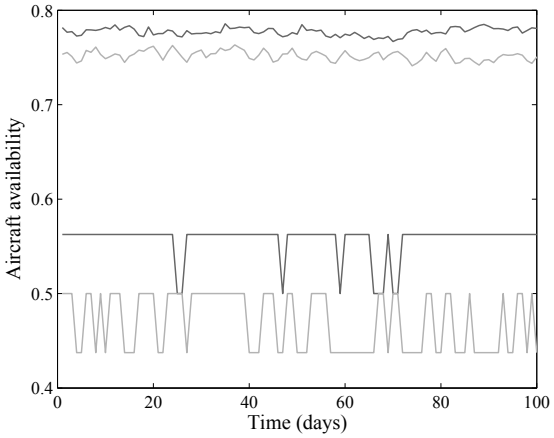


Figure 2: The availability of aircraft obtained with the optimal policy and the heuristic reference policy. Dark gray denotes the optimal policy and light gray the heuristic policy. The average timely availability as well as the 5th percentile of the availability are shown.

4.2 Results for Formulation 2

In Formulation 2, the inputs for the simulation model of aircraft usage and maintenance described in Table 1 are used with two additions, namely the target level of aircraft availability denoted with β and the duration of the scenario T. To emphasize that Formulation 2 models primarily a situation where major changes in operations are likely to occur, the target availability level is time-dependent as follows: $\beta(t) = 0.5, t \in [0, T/2]$ and $\beta(t) = 0.75, t \in [T/2, T]$. T is chosen to be 30 days.

Essentially the same parameters are used for the SARSA(λ) algorithm as with λ-SMART in Formulation 1 (see Table 2). 1000 independent simulation replications, each 30 days long, are performed to learn the optimal policy. Additionally, rewards $r_a = -1$ and $r_m = 0.2$ are utilized. Figure 3 presents the learned policy. In the beginning, i.e., with low values of d, maintenance activities are started at higher congestion

level than in the middle and closing stages of the time period. The RL algorithm thus performs as expected.

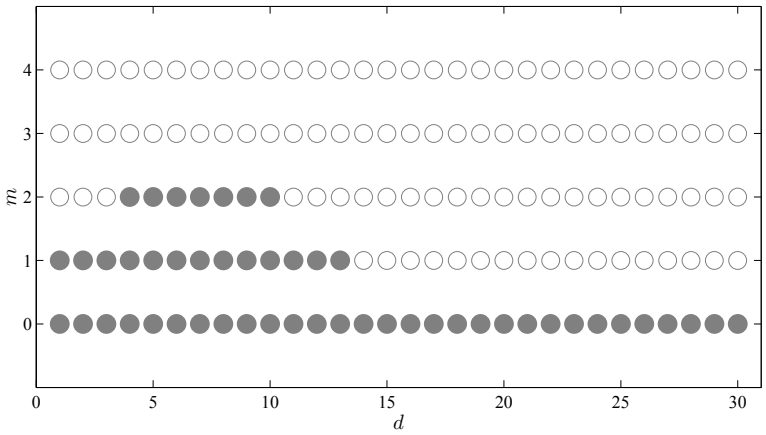


Figure 3: The optimal policy. A filled circle in a given state indicates that the optimal action is *start* and a circle with no fill that the optimal action is *delay*.

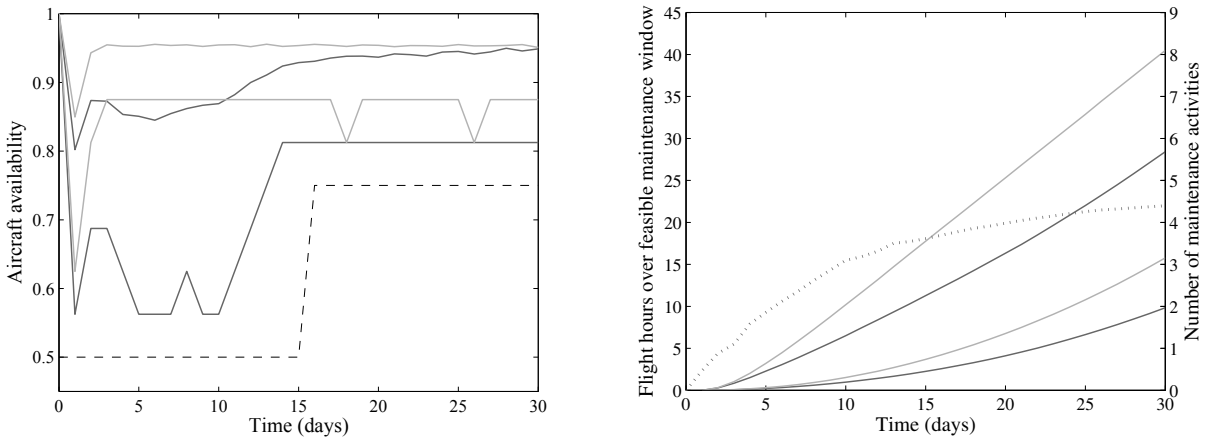
Figure 4(a) presents the development of the average aircraft availability as well as the 5th percentile of the availability with the optimal policy. The availability target is exceeded during the entire time period. As a reference, the usage of aircraft is simulated such that no maintenance is performed besides failure repairs. The results for this reference policy, which also exceeds the target level of availability, are also depicted in Figure 4(a). The cumulation of the elapsed flight hours of the aircraft and the number of performed maintenance activities with the optimal policy and the case with no maintenance are depicted in Figure 4(b). All results are based on 1000 independent simulation replications.

The benefit of rewarding maintenance in the optimal policy is that the elapsed flight hours are kept at a lower level. The aircraft are thus in better operating condition at the end of the period and the fleet is in a better position with regard to future operations after the time period. Whereas the optimal policy provides information on efficient maintenance decisions, the simulation of the policy provides knowledge of achievable level of readiness in terms of aircraft availability as well as the condition of the aircraft as described above. Moreover, the simulation of the optimal policy provides a way to assess the prospected amount and timing of maintenance activities which can be utilized for the planning of conflict operations.

5 CONCLUSIONS

In this paper, the problem of scheduling the maintenance of fighter aircraft under conflict operating conditions is considered. Two SMDPs are formulated and solved with RL techniques. The first formulation involves the maximization of the average aircraft availability. The second formulation is concerned with maintaining a specific target level of availability and simultaneously performing a maximum number of maintenance activities. RL algorithms are able to find efficient maintenance policies in both cases. These policies also provide improved performance in terms of the aircraft availability and the number of performed maintenance activities compared to heuristic reference policies.

In applying the RL techniques in this paper, a small subset from a large number of available variables are used to represent system state. The selected variables are identified as sufficient for learning purposes through several tests. Thus, a look-up table presentation (e.g., Gosavi 2003), in which the Q-factors are stored explicitly for each state-action pair, is computationally feasible for solving the problem formulations and the learned policies are relatively simple. It should be noted that the policies are conditional on the inputs of the simulation. The capability to use the policies in actual decision-making requires the solution



(a) Aircraft availability

(b) Elapsed flight hours and number of maintenance activities

Figure 4: The figure on the left depicts the development of aircraft availability and the 5th quantile of the availability. Dark gray denotes the optimal policy and light gray the case where no maintenance is performed. The dashed line depicts the target level of availability. The figure on the right depicts the maximum and average number of flight hours of aircraft over the feasible maintenance window. Dark gray again denotes the optimal policy and light gray the case with no maintenance. The number of started maintenance activities for the optimal policy is presented with a dotted line.

of several problem instances corresponding to different operating conditions as well as the capability to identify the situation that is actually at hand. In practice, the policies are therefore primarily used to infer overall guidelines for decision-making instead of exact rules as well as to assess what kind of state information is necessary in the decision-making.

Another way in which the investigation of the maintenance scheduling problems benefits military DMs is that it offers information on the achievable level of readiness. This information can be used as a basis for planning conflict operations or contingency plans. For instance, the simulation results for the policy learned in Formulation 1 indicate the number of aircraft that are at best immediately available for a flight mission with a given degree of confidence. Alternatively, the DMs could assess what arrangements are required in terms of maintenance policy or resources to complete specific conflict operations by investigating problem instances similar to Formulation 2.

Future consideration of conflict time maintenance of aircraft with RL may include other problems such as selecting the airbase for aircraft that are returning from a mission and require maintenance. In addition, Formulation 2 is clearly a multi-objective problem. The future research should further examine RL approaches that are able to handle multiple objectives.

REFERENCES

Bertsekas, D. P. 1995. *Dynamic Programming and Optimal Control*. Belmont, Massachusetts: Athena Scientific.

Bertsekas, D. P., and J. N. Tsitsiklis. 1996. *Neuro-Dynamic Programming*. Belmont, Massachusetts: Athena Scientific.

Gosavi, A. 2003. *Simulation-based Optimization - Parametric Optimization Techniques and Reinforcement Learning*. Boston, Massachusetts: Kluwer Acadmic Publishers.

Gosavi, A., N. Bandla, and T. K. Das. 2002. "A Reinforcement Learning Approach to a Single Leg Airline Revenue Management Problem with Multiple Fare Classes and Overbooking". *IIE Transactions* 34 (9): 729–742.

Kaelbling, L. P., M. L. Littman, and A. W. Moore. 1996. "Reinforcement Learning: A Survey". *Journal of Artificial Intelligence Research* 4:237–285.

Kozanidis, G., G. Liberopoulos, and C. Pitsilkas. 2010. "Flight and Maintenance Planning of Military Aircraft for Maximum Fleet Availability". *Military Operations Research* 15 (1): 53–73.

Mattila, V. 2007. "Flight Time Allocation for a Fleet of Aircraft through Reinforcement Learning". Poster presented at the 2007 Winter Simulation Conference, Washington, DC. Available via http://www.sal.tkk.fi/publications/ppt-files/cmat07.ppt [accessed September 12, 2011].

Mattila, V., and K. Virtanen. 2006. "Scheduling Periodic Maintenence of Aircraft through Simulation-Based Optimization". In *Proceedings of the 47th Conference on Simulation and Modelling*, edited by E. Juuso, 38–43. Helsinki, Finland: Finnish Society of Automation.

Mattila, V., K. Virtanen, and T. Raivio. 2008. "Improving Maintenance Decision-Making in the Finnish Air Force through Simulation". *Interfaces* 38 (3): 187–201.

Simao, H., and W. B. Powell. 2009. "Approximate Dynamic Programming for Management of High-Value Spare Parts". *Journal of Manufacturing Technology Management* 20 (2): 147–160.

Sutton, R. S., and A. G. Barto. 1998. *Reinforcement Learning: An Introduction*. Cambridge, Massachusetts: MIT Press.

Tang, L., G. Kacprzynski, J. Bock, and M. Begin. 2006. "An Intelligent Agent-Based Self-Evolving Maintenance and Operations Reasoning System". Paper presented at the 2006 IEEE Aerospace Conference, Big Sky, MT. doi:10.1109/AERO.2006.1656106.

Wu, T. T., W. B. Powell, and A. Whisman. 2009. "The Optimizing-Simulator: An Illustration Using the Military Airlift Problem". *ACM Transactions on Modeling and Computer Simulation* 19 (3): 14:1–14:31.

AUTHOR BIOGRAPHIES

VILLE MATTILA received the M.Sc. degree in Industrial Engineering and Management with a minor in Systems and Operations Research from Helsinki University of Technology, Espoo, Finland, in 2002. He is currently a researcher at the Systems Analysis Laboratory, Aalto University School of Science and is working toward the Ph.D. degree. His research interests include discrete-event simulation and simulation-based optimization with applications in aircraft maintenance and airbase logistics systems. His e-mail address is Ville.A.Mattila@tkk.fi.

KAI VIRTANEN received the M.Sc. and Dr.Tech. degrees in systems and operations research from the Helsinki University of Technology, Espoo, Finland, in 1996 and 2005, respectively. He is currently Adjunct Professor at the Systems Analysis Laboratory in the Aalto University School of Science, Espoo, Finland. His research interests include optimization, decision and game theory with particular attention to aerospace applications as well as discrete-event simulation. He is the author of about 40 publications in scientific journals and conferences on these fields. His e-mail address is Kai.Virtanen@tkk.fi.