THE SCHEDULING OF MAINTENANCE SERVICE

Shoshana Anily[†] Celia A. Glass[‡] Refael Hassin[§]

Abstract

We study a discrete problem of scheduling activities of several types under the constraint that at most a single activity can be scheduled to any one period. Applications of such a model are the scheduling of maintenance service to machines and multi-item replenishment of stock. In this paper we assume that the cost associated with any given type of activity increases linearly with the number of periods since the last execution of this type. The problem is to find an optimal schedule specifying at which periods to execute each of the activity types in order to minimize the long-run average cost per period.

We investigate properties of an optimal solution and show that there is always a cyclic optimal policy. We propose a greedy algorithm and report on computational comparison with the optimal. We also provide a heuristic, based on regular cycles for all but one activity type, with a guaranteed worse case bound.

Keywords: Scheduling, maintenance.

[†]Faculty of Management, Tel-Aviv University, Tel-Aviv 69978, Israel. anily@taunivm.tau.ac.il

[‡]Faculty of Mathematical Studies, University of Southampton, Southampton SO9 5NH, England. cag@maths.soton.ac.uk

[§]Department of Statistics and Operations Research, Tel Aviv University, Tel Aviv 69978, Israel. hassin@math.tau.ac.il

1 Introduction

We study a problem of scheduling activities of several types. We find it convenient to describe it in terms of scheduling maintenance service to a set of machines.

We consider an infinite horizon discrete time maintenance problem of m machines, $M_1, ..., M_m$. The cost of operating a machine at any given period depends on the number of periods since the last maintenance of that machine. We start with a linear cost structure where each machine i is associated with a constant a_i and the cost of operating the machine in the j-th period after the last maintenance of that machine is ja_i, for $j \geq 0$. We assume that no cost is associated with the maintenance service. Each period service may be given to at most one of the machines. The problem is to find an optimal policy specifying at which periods to service each of the machines in order to minimize the long-run average operating cost per period.

Another application of this model concerns the problem of infinite horizon, discrete time, multi-item replenishment of m items where at each period the stock of at most one of the items may be replenished. The only costs involved are item-specific linear holding cost that are incurred at the end of each period. Let d_i denote the demand per period of item i and let h_i be its unit holding cost per period. Define also $a_i = d_i h_i$. The cost of holding the stock of the ith item j periods prior to the next replenishment of that item is therefore ja_i.

In the maintenance problem the cost related to a machine is increasing up to its next service and in the replenishment problem the cost related to an item is decreasing up to its next reorder point. However, the average long run cost of the systems are of the same structure.

We start by proving that there is an optimal schedule which is cyclic, in Section 2 and proceed in Section 3 to present an algorithm for finding an optimal solution, based on network flow techniques. The two machine case is solved directly in Section 4 and lower bounds of an optimal value presented in Section 5. In Section 6 we present a heuristic with a bounded error guarantee. However, for practical purposes we recommend the simple rule presented in Section 7. It is easily programmed, requires very little computing time, and as demonstated by a numerical study in Section 8, produces near-optimal solutions. We conclude the paper with a list of open problems.

Papers containing analysis of similar type problems are [2, 3, 4, 5, 7, 8, 10, 12]. In particular, in the model treated in [8], there are bounds for each machine on the length of time without maintenance and the problem is to compute a minimum length cyclic policy obeying these bounds. In [12, 10] the exact maintenance intervals for each of the machines are given, and the problem is to minimize the number of servers needed to form a schedule.

2 Existence of an optimal policy

A *policy* P, is a sequence $P = i_1, i_2, ...$ where $i_k \in \{1, ..., m\}$ for $k = 1, 2, ...$ denotes the machine scheduled for service during the k-th period. A policy is *cyclic* if it consists of repetitions of a finite sequence $i_1, ..., i_T$. Such a sequence is said to *generate* the policy. The minimum length of a generating sequence is denoted $T(P)$. For example, 122212221... is cyclic with $T = 4$. Any set of $T(P)$ consecutive periods constitutes a *basic cycle* of P. A cyclic policy P is sometimes identified with its generating sequence S, so that we use $T(S)$ for $T(P)$.

Without loss of generality we assume that $a_1 \geq a_2 \geq ... \geq a_m$. Moreover, we scale the a_i values so that $a_m = 1$. For a policy P, let $C(t, P)$ denote the average cost over periods $1, ..., t$. Clearly, we can restrict ourselves to policies with bounded average costs and therefore we can define for each such policy P the lim sup of its sequence of average costs:

$$C(P) = \overline{\lim}_{t \to \infty} C(t, P).$$

A policy is *optimal* if it minimizes $C(P)$. We let C^* denote the average cost of an optimal policy.

Theorem 2.1 *There exists an optimal cyclic policy for the above defined problem.*

Proof: The proof of Theorem 2.1 follows directly from Lemmas 2.2 and 2.3 below. These lemmas show that it is sufficient to consider cyclic policies with bounded cycle length. Since there are finitely many such policies it follows that there exists an optimal cyclic policy. ■

Lemma 2.2 *For every policy P there exists a policy P^* such that the number of periods between two consecutive maintenance services to M_i is bounded from above by $2m(a_1/a_i + 1)$ for $i = 1, ..., m$ and $C(P^*) \leq C(P)$.*

Proof: Let P be given by an infinite sequence $(i(t))_{t=1}^{\infty}$ where $i(t)$ denotes the machine maintained according to P at period t. Let $t(P)$ be the first period in which, according to policy P, machine with index $i(t(P))$ is maintained at period $t(P)$ but is not maintained during the following $\lfloor 2m(a_1/a_{i(t(P))}+1)\rfloor$ periods. We may assume that there exists a finite $t(P)$ since otherwise there is nothing more to prove. In order to construct a suitable policy P^*, we will define a sequence of policies P_k, for $k = 0, 1, ...$, $P_0 = P$, for which the cost incurred at any period l, $l \geq 1$, by policy P_k does not exceed the cost incurred by policy P_{k-1} at the same period, and $t(P_k) > t(P_{k-1})$. Policy P_k is constructed from policy P_{k-1} as follows: let $i' = i(t(P_{k-1}))$, that is according to policy P_{k-1}, $M_{i'}$ is not maintained for $\lfloor 2m(a_1/a_{i'}+1)\rfloor$ consecutive periods after it was maintained at period $t(P_{k-1})$. Consider period $\tau_1 = t(P_{k-1}) + \lfloor 2ma_1/a_{i'}\rfloor+1$. The cost of operating M_i' at period τ_1 is at least $2ma_1$. During the next $2m-1$ periods after period τ_1 $M_{i'}$ is not maintained, therefore there exists a machine $M_{i''}$ that is maintained during these periods at least three times. Suppose that the second (third) maintenance of $M_{i''}$ during these $2m-1$ periods occurs at period $\tau_2 = \tau_1 + \delta_1$ ($\tau_3 = \tau_1 + \delta_2$) for $2 \leq \delta_1 < \delta_2 \leq 2m-1$. The new policy P_k is identical to P_{k-1} at all periods except at period τ_2; according to policy P_k at period τ_2 $M_{i'}$ is maintained instead of $M_{i''}$. Consequently, $t(P_k) > t(P_{k-1})$.

We now prove that, for any period l, $l \geq 1$, and for any given integer k, the cost incurred by P_k at period l does not exceed the cost incurred by P_{k-1} at the same period. Take some positive integer k. In order to compare the costs of the two policies P_k and P_{k-1} for each period it is sufficient to consider machines $M_{i'}$ and $M_{i''}$ since they are the only ones affected by the above change. Clearly, the cost for each period under both policies at the first τ_2-1 time periods is identical. After period τ_3 the cost associated with $M_{i''}$ is identical for both policies. Under policy P_k, machine i' obtains an additional service prior to period τ_3. Thus from period τ_3 on, the cost incurred at each period by $M_{i'}$ is not larger under policy P_k than the respective cost according to policy P_{k-1}. It remains to compare the cost of these two machines in periods $\tau_2, \tau_2+1, ..., \tau_3-1$: the saving on $M_{i'}$ at each period during this interval is at least $2ma_1$. The additional cost due to $M_{i''}$ during each of these periods is at most $a_{i''}(\tau_3-\tau_1-1) < 2ma_{i''}$ which is bounded from above by $2ma_1$, since $\tau_3-\tau_1-1 < 2m$. Thus, the cost of P_k is no greater than that of P_{k-1} for all periods.

According to the above construction, policies $(P_j)_{j=k}^{\infty}$ coincide on the first $t(P_k)$ periods. As $t(P_k)$ is monotone increasing we conclude that a limiting policy P^* exists. By construction, the cost at each period of P^* is bounded from above by the respective cost of P for all periods, resulting in $C(P^*) \leq C(P)$. ■

As a result of Lemma 2.2 it is sufficient to look at the class of policies $\mathcal{P}$ in which the number of periods between two consecutive maintenance services to each M_i is not greater than $2m(a_1 + 1)$ since we have scaled the a_is to ensure that $1 = a_m \leq a_i$.

Define the *state* of the system at a given period as a vector $s_1, ..., s_m$, where s_i denotes the number of periods since the last maintenance of M_i.

Lemma 2.3 *For each policy $P \in \mathcal{P}$ there exists a cyclic policy $P' \in \mathcal{P}$ for which $C(P') \leq C(P)$.*

Proof: In view of Lemma 2.2, the number of possible states for each M_i for a policy $P \in \mathcal{P}$ is bounded from above by $2m(a_1/a_i + 1) \leq 2m(a_1 + 1)$. Therefore, the total number of possible states, considering the m machines, is bounded by $(2m(a_1 + 1)^m)$. In view of the finiteness of the state space and the stationarity of the model, there exists a policy $P^* \in \mathcal{P}$ that is cyclic and $C(P^*) \leq C(P)$. ■

Remark 2.4 Theorem 2.1 enables us to refer from now on to cyclic policies only. We will do so implicitly in the rest of the paper. Indeed, we will refer to a policy by its defining cycle.

3 A finite algorithm

Let the *state* at a given period be a vector $(s_1, ..., s_m)$ where $s_i \in \{0, 1, ...\}$ specifies the number of periods since the last service to M_i.

An optimal policy can be computed through network flow techniques. Specifically, consider a directed graph with a vertex set corresponding to the states $(s_1, ..., s_m)$ satisfying

1. $s_i \in \{0, ..., u_i\}$ $i = 1, ..., M$, where u_i is an upper bound on s_i;

2. $s_i \neq s_j$ for $i \neq j$;

3. $s_i = 0$ for some $i \in \{1, ..., m\}$.

The arc set consists of arcs from a vertex $(s_1, ..., s_m)$ to the vertices $(s_1 + 1, ..., s_{k-1} + 1, 0, s_{k+1} + 1, ..., s_m + 1)$, for $k = 1, ..., m$. The cost associated with each of these arcs is equal to $\sum_i a_i s_i$. Our task is to compute a minimum average cost cycle in this graph. This can be accomplished in time that is quadratic in the number of nodes ([9]). However, the number of states, and hence the algorithm's complexity, is exponential even when the a values are bounded.

We want to determine low upper bounds on $\{s_i\}$ in an optimal solution. The lower are these bounds the larger are the problems that we can optimally solve. From Lemma 2.2 we have $s_i \leq 2m(a_1/a_i + 1)$ for $i = 1, ..., m$. In particular, $s_1 \leq 4m$. Indeed we conjecture that $s_1 \leq m$ is also correct. We will reduce the other bounds and give the new values in terms of the bound on s_1.

Theorem 3.1 *Let u_1 be an upper bound on the value of s_1 in an optimal policy. Then,*

$$u_i = \sqrt{4\frac{a_1}{a_i}(u_1 + 1)}$$

is an upper bound on s_i, $i = 2, ..., m$.

Proof: We derive the bounds under the assumption that s_i is not bounded by u_1. (They trivially hold in the other case.) Suppose M_i $i \neq 1$ is serviced according to an optimal policy at period t and then is not serviced for $s_i \geq u_1$ consecutive periods till period $t + s_i + 1$. We will compare the cost of adding a service to M_i instead of one of the services to M_1. We search for the period $\tilde{t}$ closest to $t + (s_i + 1)/2$ in which M_1 is serviced; replace this service by a service to M_i. Since, (according to the definition of u_1) M_1 must obtain service during any $u_1 + 1$ consecutive periods, the period $\tilde{t}$ at which this exchange is made must satisfy $|t + (s_i + 1)/2 - \tilde{t}| \leq (u_1 + 1)/2$. Let $c_i(\tau)$ denotes the total cost due to M_i for periods $1, ..., \tau - 1$ if the machine is serviced in period 0 and is not serviced during periods $1, ..., \tau - 1$, i.e., $c_i(\tau) = a_i \tau(\tau - 1)/2$.

The cost incurred by deleting the service to M_1 is at most $c_1(2u_1+2) - 2c_1(u_1+1)$ or $a_1(u_1+1)^2$. The least saving due to M_i is

$$c_i(s_i+1) - c_i(\frac{s_i-u_1}{2}) - c_i(s_i+1-\frac{s_i-u_1}{2}) = a_i\frac{s_i(s_i+2)-u_1(u_1+2)}{4}.$$

Note that the cost of the other machines $M_j,\ j \neq i, j \neq 1$ is not affected by the above exchange. If the maximum additional cost due to M_1 were strictly less than the least saving due to M_i then we could reduce the total cost per cycle by the above exchange, contradicting the optimality of the starting policy. Therefore, by simple algebra, the concavity of the square root function and the fact that $u_1 > 1$, we conclude that $s_i \leq \sqrt{4\frac{a_1}{a_i}}(u_1+1)$. ■

4 Two machine case

We now solve the problem with two machines.

Theorem 4.1 *An optimal solution C^* to the 2-machine problem with cost constants $a_1 \geq 1$ and $a_2 = 1$ is the policy with basic cycle length τ^* containing one service to M_2, where τ^* is the unique integer satisfying*

$$(\tau^*-1)\tau^* \leq 2a_1 < \tau^*(\tau^*+1).$$

Proof: Consider an optimal policy P and suppose that M_2 is maintained at least twice during a cycle. From theorem 2.1 we may assume that P is cyclic. Denote its cycle length by τ. We show first that M_2 is not maintained in any two consecutive periods and then that each interval between two consecutive services of M_2 must have the same cost. From this it follows that there is an optimal policy with a basic cycle containing precisely one service of M_2. We refer to a policy of this type with basic cycle length τ as $P(\tau)$. We then show that the cost function $C(P(\tau))$ is convex in τ and find its minimum value.

Suppose first that there are two or more consecutive periods with service to M_2. Consider the average cost of the solution P' with $T(P') = \tau - 1$ obtained by cancelling one of these services. Then,

$$(\tau-1)C(P') \leq \tau C(P) - 2a_1$$

so that

$$C(P') \leq C(P) + \frac{C(P) - 2a_1}{\tau - 1}.$$

But $C(P) \leq a_1$ since the alternating policy with a period of size 2 has an average cost of $(1 + a_1)/2 \leq a_1$. Hence, $C(P') < C(P)$, a contradiction.

Since P is a cyclic policy we may consider a basic cycle starting at any point in the cycle. We shall consider a basic cycle starting with a service to M_2. We may then partition it into parts, each starting with a service to M_2, terminating in a service to M_1 and otherwise containing only services to M_1, since services to M_2 do not occur consecutively, from above. Then, $C(P)$ is the weighted (in the number of periods) average of the average costs of the parts. We can produce a new policy by repeating a part with the lowest average cost; the average cost of the policy produced is at most $C(P)$ and its basic cycle contains a single service to M_2.

The total cost per cycle due to $P(\tau)$ is $\sum_{i=1}^{\tau-1} i + a_1 = \frac{\tau(\tau-1)}{2} + a_1$. Thus, the average cost

$$C(P(\tau)) = \frac{\tau - 1}{2} + \frac{a_1}{\tau}. \tag{1}$$

Our task of finding an optimal solution C^*, reduces to computing the integer τ^* that minimizes $C(P(\tau)) = \frac{\tau-1}{2} + \frac{a_1}{\tau}$. Since $C(P(\tau))$ is a strictly convex function in τ, this task is achieved by differentiating $C(P(\tau))$ with respect to τ and rounding. If $\sqrt{2a_1}$ is an integer then $\tau^* = \sqrt{2a_1}$. Otherwise, we compare $C(P(\lfloor\sqrt{2a_1}\rfloor))$ with $C(P(\lceil\sqrt{2a_1}\rceil))$ and choose τ^* as the cycle giving a lower average cost. This leads to the expression in the statement of the theorem. ■

5 Lower bounds

In this section we derive lower bounds on the cost of an optimal policy.

Theorem 5.1 *A lower bound on the cost of an optimal solution is given by*

$$\sum_{i=2}^{m}(i-1)a_i.$$

Proof: At each period there must be at least one machine that has not been maintained during the last $m-1$ periods, another one that has not been maintained for at least $m-2$ periods, and so on. A lower bound is obtained when we assume that the machines that have not been maintained for a longer time are those with lower costs. ■

This bound is strengthened by the following theorem.

Theorem 5.2

$$LB1 = \sum_{j=1}^{m}\sum_{i<j}^{m}\sqrt{a_i a_j}$$

is a lower bound on the cost of an optimal policy.

Proof: Consider a relaxation of the problem in which we assume that any number of services may be performed in a single period. The variables in the relaxed problem are integers T and $n_1, ..., n_m$ where T is the length of the basic cycle and n_i represents the number of times M_i is maintained during the basic cycle, for $i = 1, \ldots, m$. There is a single constraint that $\sum_{i=1}^{m} n_i = T$. The objective is to minimize the average cost per unit time. Consider a machine M_i. Let $\tau^{(j)}$, for $j = 1, \ldots, n_i$, denote the integer number of periods in the intervals in a basic cycle between services to M_i. Then the total cost of servicing M_i is $\frac{a_i}{2}\sum_{j=1}^{n_i}(\tau^{(j)}-1)\tau^{(j)}$. While optimizing this expression we further relax the constraints and allow the variables $\tau^{(j)}$ to be continuous. Thus, the intervals between services are no longer restricted to be integer and, since we allow services to overlap, the service times to each of the machines may be optimized independently, apart from the constraint on the sum of the n_i's.

This function is optimized by taking equi-distance intervals $\tau_i = T/n_i$. In this case, the average cost associated with M_i is just $(n_i a_i(\tau_i - 1)\tau_i)/2T = a_i(\tau_i - 1)/2$, and the total average cost of the relaxed problem is bounded from below by the solution value to the following problem:

$$\begin{aligned} &\text{minimize} && \tfrac{1}{2}\textstyle\sum_{i=1}^{m} a_i(\tau_i - 1), \\ &\text{subject to:} && \\ & && \textstyle\sum_{i=1}^{m}(1/\tau_i) = 1. \end{aligned}$$

By deleting constant factors in the objective function and applying Lagrangian relaxation we obtain an equivalent form of the problem:

$$\text{minimize} \quad \sum_{i=1}^{m}(a_i\tau_i) - \lambda(1-\sum_{i=1}^{m}(1/\tau_i)).$$

The solution to this problem satisfies the following system of equations

$$\begin{array}{ll} \lambda = a_i\tau_i^2 & \text{for } i=1,...,m, \\ \sum_{i=1}^{m} 1/\tau_i = 1, & \end{array}$$

and is therefore given by $\lambda = (\sum_{j=1}^{m}\sqrt{a_j})^2$ and $\tau_i = \tau_i^R$ where

$$\tau_i^R = \sum_{j=1}^{m}\sqrt{a_j}/\sqrt{a_i}.$$

The cost of the corresponding solution is

$$\frac{1}{2}\sum_{i=1}^{m} a_i(\tau_i^R - 1) = \frac{1}{2}(\sum_{i=1}^{m}\sqrt{a_i})^2 - \frac{1}{2}(\sum_{i=1}^{m} a_i) = \sum_{i<j}\sqrt{a_i a_j} = LB1, \qquad (2)$$

giving a lower bound on the optimal cost of the original problem. ■

Remark 5.3 Observe that the bound of Theorem 5.1 is a weaker bound than LB1 since, according to our assumption, for $i < j$, $a_i \geq a_j$, thus $\sqrt{a_i a_j} \geq a_j$.

Remark 5.4 The lower bound $LB1$ may be far from optimum when τ_1^R is close to 1, i.e. $a_1 >> \sum_{i=2}^{m} a_i$. The actual average cost due to M_1 in the optimal policy will be much greater than M_1's contribution to $LB1$. In order to see this gap, consider again the 2-machine problem with $a_2 = 1$ and M_2 is served exactly once in a cycle. The continuous relaxation of the average cost function (1) provides a lower bound of $\sqrt{2a_1} - 0.5$ on the optimal average cost. Then,

$$\lim_{a_1\to\infty}\frac{C(P(\tau^*))}{LB1} \geq \lim_{a_1\to\infty}\frac{\sqrt{2a_1}-0.5}{\sqrt{a_1}} = \sqrt{2}, \qquad (3)$$

where τ^* denotes the optimal basic cylce length.

When a_1 is large relative to the other costs, the optimal solution will include consecutive services to M_1 and the quality of $LB1$ will be poor. We now present a lower bound that will perform well exactly in these cases. This observation is validated by computational results presented in Section 8.

Let C_{1i} denote the solution value, i.e. the minimum average cost, of the two machine problem consisting of M_1 and M_i.

Theorem 5.5 *$LB2 = \sum_{i=2}^{m} C_{1i}$ is a lower bound on the cost of an optimal policy.*

Proof:

Consider a relaxation of the problem in which we assume that machines M_i, for $i = 2, ..., m$, can be serviced simultaneously. The condition $\sum_{i=1}^{m} n_i = T$ ensures that for each additional machine service in a given time period there is a corresponding empty time period with no service elsewhere in the basic cycle.

A lower bound on the cost of a solution to the relaxed problem is given by costing maintenance to M_1 at a_1 for each time period in which it is not serviced and to $M_2, ..., M_m$ in the usual way. This is equivalent to accruing a cost due to M_1 of a_1 for each of the services to $M_2, ..., M_m$.

Thus solutions to this relaxed problem, with lower bound costs, correspond to an amalgamation of $m-1$ 2-machine problems for M_i and M_1, $i \geq 2$. Therefore, the least cost of the latter $\sum_{i \geq 2}, C_{1i} = LB2$, provides a lower bound to the former and hence to the original problem. ∎

6 Bounded error heuristic

In this section we develop a simple policy and show that its worst case ratio error is bounded by 2.5. According to the proposed policy the machines, except possibly M_1, are maintained in equi-distant time intervals which are machine dependent, where the time intervals are given as integer power of two. Before proceeding with the algorithm we need to describe some properties of a policy in which machines are maintained at frequencies which are integer powers of two.

Lemma 6.1 *If $\sum_{i=1}^{m} 2^{w_i} = 2^W$ for some integers W, and $w_1 \leq \ldots \leq w_m$, and $m > 1$, then there is a partition of the set $\{w_1, \ldots, w_m\}$ at some integer ℓ for which*

$$\sum_{i=1}^{\ell} 2^{w_i} = \sum_{i=\ell+1}^{m} 2^{w_i} = 2^{W-1}.$$

Proof: See, Lemma 1 in [6]. ■

Lemma 6.2 *Given $\tau_1, \tau_2, ..., \tau_m$ such that*
(i) $\sum_{i=1}^{m} \tau_i^{-1} \leq 1$ *and*
(ii) $\tau_i = 2^{\ell_i}$, $\ell_i \in I\!N$ *for* $i = 1, ..., m$,
there exists a policy with a basic cycle length of $T = \max\{\tau_i\}$*, in which* M_i *is serviced at equi-distant intervals of length* τ_i *each. This policy can be constructed in* $O(m \log m)$ *time.*

Proof: Suppose that the indices are ordered so that $\tau_1 \leq \tau_2 \leq ... \leq \tau_m$. According to such a policy M_i must be serviced n_i times during a cycle where

$$n_i = \frac{T}{\tau_i} = 2^{w_i}, \quad w_i = \ell_m - \ell_i \in I\!N.$$

Thus, the number of services to each machine during a cycle is an integer powers of two and $n_1 \geq n_2 \geq ... \geq n_m$.

Without loss of generality we assume that $\sum \tau_i{}^{-1} = 1$, or equivalently, that $\sum n_i = T = 2^{\ell_m}$. Otherwise, we may introduce dummy machines $M_{m+1}, ..., M_{m+k}$, with $\tau_i = 2^{\ell_m} = T$ for $i = m+1, \ldots, m+k$. Each of these machines will be serviced once during the cycle to fill the k gap periods with services.

The required policy is fully specified by the periodicity τ_i and the t_i, the first period in the basic cycle in which M_i is serviced for $i = 1, ..., m$. It can be constructed by repeated applications of Lemma 6.1 as demonstrated below. In the initial step we start with the set, $A = \{M_1, ..., M_m\}$ and we allocate all of the T periods in a basic cycle to the machines in A without specifying the assignment of machines to periods. We set $\tau(A) = t(A) = 1$ meaning that machines from A are serviced each $\tau(A)$ periods starting from $t(A)$. (Only $t(A)$ is required for the algorithm, we define $\tau(A)$ just for the sake of the description.) In the second step of the procedure we partition A into two subsets, B and C, as follows. Let ℓ be as in Lemma 6.1 with $\sum_{i=1}^{\ell} n_i = \sum_{i=\ell+1}^{m} n_i$, then take B to be $\{M_1, ..., M_\ell\}$ and C to be $\{M_{\ell+1}, ..., M_m\}$. We allocate the periods to B and C in an alternating fashion: $BCBCBC...BC$. We set $\tau(B) = \tau(C) = 2$, $t(B) = 1$, and $t(C) = 2$. We repeat this procedure for the machines within each set. In the general step, if a set F is partitioned into sets G and H, then we allocate the periods assigned to F in order $GHGH\dots$ and set $\tau(G) = \tau(H) = 2\tau(F)$, $t(G) = t(F)$, and $t(H) = t(F) + \tau(F)$. The process is repeated as long as there are sets consisting of more than one machine. It ends with sets $\{M_i\}$ such that $\tau(\{M_i\}) = \tau_i$. Setting $t_i = t(\{M_i\})$ we obtain the required poicy.

To implement the algorithm we first compute the partial sums $N_j = n_1 + ... + n_j$ for $j = 1, ..., m$. This takes linear time. Then, each application of Lemma 6.1 requires a binary search in the relevant range of the values N_j and takes $O(\log m)$ time, while defining the t_j values takes constant time. In total there are $m - 1$ such iterations. The complexity of ordering the indices and constructing the policy is therefore $O(m \log m)$. ■

We now proceed with the description of a *power of two heuristic*. We start with a generally infeasible solution which is known to have low cost, namely the solution to the relaxed problem induced by $LB1$ described in Section 5. From it we construct a schedule with basic cycle length which is an integer power of two, in which the frequency of maintenance service to any of $M_2, ..., M_m$ is reduced by at most a factor of 2 to an integer power of two. This schedule is completed by providing the most expensive machine, M_1, with services at all periods in which none of $M_2, ..., M_m$ is serviced. Thus all machines other than M_1 are serviced at regular intervals which are powers of 2, while M_1 is possibly not. Recall that $a_1 \geq a_2 \geq \dots a_m$. The trivial case $m = 1$ is excluded from consideration.

Power-of-two heuristic:

For $i = 1, ..., m$:
  $\tau_i^R \leftarrow \sum_{j=1}^{m} \sqrt{a_j}/\sqrt{a_i}$;
  $\ell_i \leftarrow$ the integer satisfying $2^{\ell_i - 1} < \tau_i^R \leq 2^{\ell_i}$;
  $\tilde{\tau}_i \leftarrow 2^{\ell_i}$.

Construct a schedule with services at regular intervals
$\tilde{\tau}_i = 2^{\ell_i}$ for $i = 1, \dots, m$,
as specified a in proof of Lemma 6.2.

Complete the schedule by using all "gap" periods, where none of $M_1, \dots, M_m$ is serviced, for additional services to M_1.

Theorem 6.3 *Let C be the average cost of the solution produced by the Power-of-Two heuristic. Let C^* be the average cost of an optimal solution. Then, $C \leq 2.5C^*$.*

Proof:

In the power of two solution, M_i for $i = 2, \ldots, m$ is serviced every $\tilde{\tau}_i$ periods and therefore has average cost $a_i(\tilde{\tau}_i - 1)/2$. The total cost due to M_1 over any $\tilde{\tau}_1$ periods is at most $a_1\tilde{\tau}_1(\tilde{\tau}_1 - 1)/2$, since M_1 is serviced at least every $\tilde{\tau}_1$ periods. As the cost due to M_1 is only accrued in periods in which M_1 is not serviced, the average cost in such periods is at most $a_1\tilde{\tau}_1/2$. Now, by construction, the proportion of periods without service to M_1 during a basic cycle of length T is $\sum_{i\geq 2} 1/\tilde{\tau}_i$. Therefore,

$$C \leq 0.5\sum_{i=2}^{m} a_i(\tilde{\tau}_i - 1) + 0.5a_1\tilde{\tau}_1(\sum_{i=2}^{m} \tilde{\tau}_i^{-1}).$$

Since $\tilde{\tau}_i \leq 2\tau_i^R$, and $\sum_{i=2}^{m}(\tilde{\tau}_i)^{-1} \leq \sum_{i=2}^{m}(\tau_i^R)^{-1} = 1 - (\tau_1^R)^{-1}$,

$$C \leq 0.5\sum_{i=2}^{m} a_i + \sum_{i=2}^{m} a_i(\tau_i^R - 1) + a_1\tau_1^R(1 - (\tau_1^R)^{-1}) = 0.5\sum_{i=2}^{m} a_i + \sum_{i=1}^{m} a_i(\tau_i^R - 1).$$

Now, from equation (2), $\sum_{i=1}^{m} a_i(\tau_i^R - 1) = 2LB1$ and applying Theorem 5.1 and Remark 5.3 gives $0.5\sum_{i=2}^{m} a_i \leq 0.5LB1$. Thus, we obtain the inequalities, $C \leq 2.5LB1 \leq 2.5C^*$, which completes the proof. ■

Remark 6.4 In this paper we assume that the operating cost of a machine is linearly dependent of the time since its last service, starting with zero cost at the period a service is given. Alternatively, we could assume that the cost at that period is already a_i. The average cost associated with any solution differs between the two versions by a constant $\sum_i a_i$, and therefore they are equivalent with respect to optimal solutions. However, the version we treat is harder to approximate with respect to the error ratio since both the optimal and approximate solutions are smaller and hence their ratio increases. When the cost functions starts at a_i,

$$LB1 = \sum_{i<j}\sqrt{a_i a_j} + \sum_{i=1}^{m} a_i$$

while the τ_i^R are exactly as in the other case. Analyzing the same power-of-two heuristic, even without completing the schedule with additional services to M_1 at the last step of the heuristic, results in a worst case ratio of 2. The analysis is much simpler for this version of the problem, since the average cost due to M_i over any $\tilde{\tau}_i$ consecutive periods is $a_i(\tilde{\tau}_i+1)/2$ for $i = 1, ..., m$. Thus we obtain the inequalities

$$C \leq 0.5 \sum_{i=1}^{m} a_i(2\tau_i^R + 1) =$$

$$\sum_{i=1}^{m} a_i(\tau_i^R + 1) - 0.5 \sum_{i=1}^{m} a_i = 2LB1 - 0.5 \sum_{i=1}^{m} a_i < 2LB1.$$

The heuristic therefore has a worst-case bound of at most 2.

7 Greedy heuristics

In this section we propose a greedy heuristic that enables us to approximately solve problems which are too large to be optimally solved by the algorithm of Section 3. We give some intuitive motivation for its design. The proposed greedy heuristic is tested computationally and results are reported in Section 8.

We compare the total cost incurred for each of the machines since the last time they were serviced, assuming that they are not serviced in the next period, and select the one with the largest such total cost for service in the next period.

Greedy rule - GR:

Take $(s_1, ..., s_m)$ at period t;
take an element $\hat{i}$ in $argmax\{a_i(s_i+2)(s_i+1) : 1 \leq i \leq m\}$;
service $M_{\hat{i}}$ in period $t+1$.

The incentive for this heuristic, arises from the lower bound $LB1$ obtained from the continuous relaxation of the problem described in Section

5. In this relaxation, when the time is continuous, the cost incurred by M_i grows linearly at a rate a_i with intercept 0. Thus, the total cost incurred between two consecutive maintenance services to M_i given at distance of τ time units one from the other is $a_i\tau^2/2$ (the area of the respective triangle). By substituting τ_i^R for τ for $i = 1, ..., m$, we find that the total cost incurred between two consecutive services to M_i is constant.

Remark 7.1 For the 2-machine case the policy produced by the greedy algorithm is an optimal policy. To see this observe that in the greedy algorithm we service M_2 after $\tau - 1$ consecutive services to M_1 for the smallest value of τ that satisfies $a_2(\tau + 1)\tau > 2a_1$. But this value of τ is the optimal basic cycle length for the 2-machine case, from Theorem 4.1.

Remark 7.2 One might have thought that the marginal cost would be a better criteria than total cost of a partial interval, i.e. using $a_i s_i$ in place of $\sqrt{a_i} s_i$ in the algorithm. But it can be shown that the resulting algorithm has an unbounded worse case ratio even in the two machine case.

8 Computational results

In this section we test the performance of the greedy heuristic GR proposed in Section 7 and the effectiveness of the lower bounds $LB1$ and $LB2$ derived in Section 5. We applied the greedy algorithm with the following tie-break rule: when $\hat{i}$ is not uniquely defined take the largest index among the candidates for selection. The initial state was arbitrarily chosen to have $s_i = i - 1$ $i = 1, 2, ..., m$. For small size problems, i.e. $m = 3$ and $m = 4$, we compute the optimal solution, denoted by OPT, according to the algorithm proposed in Section 3.

For the three-machine problem we also include the basic cycle length T for each of the two schedules, OPT and GR. We use LB to denote $\max\{LB1, LB2\}$. In order to facilitate the comparison we use bold letters for LB. The effectiveness of the lower bounds and of the heuristic is measured by the ratios OPT/LB and GR/OPT for $m = 3$ and $m = 4$, and by the ratio GR/LB in all other cases.

Results of our computational experiments, for a selection of instances with 3,4,5 and 10 machines, are presented in Tables 1 to 4.

a_1	a_2	$LB1$	$LB2$	OPT	GR	T_O	T_G	$\frac{OPT}{LB}$	$\frac{GR}{OPT}$
1	1	**3.00**	2.00	3.00	3.00	3	3	1.000	1.000
2	1	**3.83**	3.00	4.00	4.00	3	4	1.044	1.000
2	2	**4.83**	3.50	5.00	5.00	3	8	1.035	1.000
5	1	**5.47**	5.33	5.50	5.50	4	4	1.005	1.000
5	2	**6.18**	6.17	7.00	7.00	4	4	1.133	1.000
5	5	**9.47**	7.67	10.00	10.00	5	12	1.056	1.000
10	1	7.32	**8.00**	8.00	8.00	4	5	1.000	1.000
10	2	9.05	**9.33**	9.50	10.00	4	5	1.018	1.000
10	5	**12.47**	11.50	13.33	13.33	6	6	1.069	1.000
10	10	**16.32**	14.00	17.25	17.25	16	16	1.057	1.000
30	1	11.95	**14.50**	14.50	14.50	8	8	1.000	1.000
30	2	14.64	**17.25**	17.29	17.39	17	18	1.002	1.006
30	5	19.96	**22.25**	22.25	22.25	8	8	1.000	1.000
30	10	25.96	**27.25**	28.44	28.50	9	10	1.044	1.002
30	30	**40.95**	37.25	42.92	42.92	13	13	1.048	1.000
50	1	15.14	**19.00**	19.00	19.00	10	10	1.000	1.000
50	2	18.49	**22.64**	22.67	22.67	21	21	1.001	1.000
50	5	25.12	**29.50**	29.50	29.50	10	10	1.000	1.000
50	10	32.59	**36.17**	36.50	36.50	10	10	1.009	1.000
50	30	**51.28**	49.50	55.00	55.23	15	13	1.073	1.004
50	50	**64.14**	59.50	66.82	66.82	17	17	1.042	1.000

Table 1: Results for examples with 3 machines ($a_3 = 1$).
$T_O \equiv T(OPT)$, $T_G \equiv T(GR)$

a_1	a_2	a_3	$LB1$	$LB2$	OPT	GR	OPT/LB	GR/OPT
1	1	1	**6.00**	3.00	6.00	6.00	1.000	1.000
2	1	1	**7.24**	4.50	7.33	7.33	1.012	1.000
2	2	1	**8.66**	5.00	8.80	8.80	1.016	1.000
2	2	2	**10.24**	5.50	10.40	10.40	1.016	1.000
5	1	1	**9.71**	8.00	10.00	10.00	1.030	1.000
5	2	1	**11.46**	8.83	11.75	11.75	1.025	1.000
5	2	2	**13.39**	9.67	13.73	13.73	1.025	1.000
5	5	1	**14.94**	10.33	15.00	15.00	1.004	1.000
5	5	2	**17.21**	11.17	17.50	17.50	1.017	1.000
5	5	5	**21.71**	12.67	22.25	22.25	1.025	1.000
10	1	1	**12.49**	12.00	12.50	12.50	1.001	1.000
10	2	1	**14.65**	13.33	15.00	15.00	1.024	1.000
10	2	2	**16.94**	14.67	17.50	17.50	1.033	1.000
10	5	1	**18.87**	15.50	19.50	19.50	1.033	1.000
10	5	2	**21.52**	16.83	22.50	22.57	1.046	1.000
10	5	5	**26.78**	19.00	27.87	27.87	1.041	1.000
10	10	1	**23.65**	18.00	24.50	24.50	1.036	1.000
10	10	2	**26.68**	19.33	27.50	28.00	1.031	1.000
10	10	5	**32.70**	21.50	34.00	34.00	1.040	1.000
10	10	10	**39.49**	24.00	40.45	40.45	1.024	1.000
30	1	1	19.43	**21.75**	21.75	21.75	1.000	1.000
30	5	1	28.67	**29.50**	29.50	29.50	1.000	1.000
30	5	5	**39.44**	37.25	40.50	40.50	1.027	1.000
30	10	1	**35.60**	34.50	37.00	37.00	1.039	1.000
30	10	5	**47.51**	42.25	49.67	51.33	1.045	1.020
30	10	10	**56.44**	47.25	58.42	58.64	1.035	1.004
30	30	1	**52.91**	44.50	55.84	55.85	1.055	1.000
30	30	5	**67.69**	52.25	70.50	70.62	1.042	1.002
30	30	10	**78.76**	57.25	81.50	81.50	1.035	1.000
30	30	30	**106.43**	67.25	108.47	108.47	1.019	1.000

Table 2: Results for examples with 4 machines ($a_4 = 1$).

a_1	a_2	a_3	a_4	$LB1$	$LB2$	GR	GR/LB
5	1	1	1	**14.94**	10.67	15.00	1.004
5	5	1	1	**21.42**	13.00	21.75	1.015
5	5	5	1	**29.42**	15.33	29.50	1.003
5	5	5	5	**38.94**	17.67	39.50	1.014
10	5	1	1	**26.27**	19.50	26.75	1.018
10	10	5	1	**42.26**	25.50	43.00	1.018
30	10	5	1	**59.39**	49.50	61.83	1.041
30	30	1	1	**65.86**	51.75	68.77	1.044
30	30	30	1	**123.86**	74.50	126.95	1.025
30	30	30	30	**201.91**	97.25	204.00	1.010
100	1	1	1	46.00	**54.57**	54.57	1.000
100	30	5	1	**125.81**	119.79	127.50	1.013
100	30	30	5	**209.59**	169.48	217.15	1.036
100	100	30	5	**294.23**	206.14	304.67	1.035

Table 3: Results for examples with 5 machines ($a_5 = 1$).

a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	$LB1$	$LB2$	GR	GR/LB
10	1	1	1	1	1	1	1	1	**64.46**	36.00	64.50	1.001
10	9	8	7	6	5	4	3	2	**224.91**	65.17	229.55	1.021
10	10	10	3	3	3	1	1	1	**153.03**	55.00	153.50	1.003
10	10	10	10	10	1	1	1	1	**189.06**	60.00	190.00	1.005
10	10	10	10	10	10	10	10	10	**388.46**	84.00	389.50	1.003
100	1	1	1	1	1	1	1	1	**126.00**	122.79	126.50	1.004
10^3	1	1	1	1	1	1	1	1	320.00	**398.00**	398.00	1.000
10^3	10^3	1	1	1	1	1	1	1	**1534.0**	1353.8	1615.7	1.053

Table 4: Results for examples with 10 machines ($a_{10} = 1$).

In addition, for the case of $m = 20$ and $a_i = 21 - i$ for $i = 1, ..., 20$ we obtained the following results: $LB1 = 1796.35$, $LB2 = 272.83$ and $GR = 1833.69$ and hence, $GR/LB = 1.021$.

The results confirm that lower bounds $LB1$ and $LB2$ are both useful. Bound $LB1$ performs better most of the time, while $LB2$ consistently does better for cases when a_1 is large compared with the other a_i values; the larger m the larger should be the relative size of a_1 for $LB2$ to outperform $LB1$. So for $m = 3$ $LB2$ frequently outperforms $LB1$ whereas it rarely does so for larger m.

The lower bound, LB, gives values within 8% and 6% of the optimum for 3 and 4 machines respectively in our experiments. Moreover, for larger problems the GR solution and hence the optimal solution are within 6% of the lower bound.

All the evidence is that GR gives a very good approximation to the optimal solution, especially for large values of m. It performs within 2% of optimality for our examples with $m = 3$ and 4 and within 6% of LB for larger m.

9 Conclusions

In this paper we address a scheduling problem which may appear simple at first sight. We present a simple rule that seems to give satisfactory approximate results. However, our theoretical analysis is not complete. We described a finite algorithm of exponential complexity. We suspect that the problem is NP-hard when the number of machines is part of the problem's input, but have so far not succeeded in proving it.

Even the three machine problem is difficult to solve analytically or by a polynomial time algorithm. In preliminary work presented in [1] our approach has been to classify cases and solve them to optimality. For the remaining cases we present a heuristic with a guaranteed worse case bound of 5%. Solving the three machine case to optimality is the subject of current research.

We assumed that the cost functions are linear. However, the results of this paper might be generalized to any convex function.

References

[1] S. Anily, C.A. Glass and R. Hassin, "Scheduling of maintenance services to three machines", OR Preprint no. OR58, Faculty of Mathematical Sciences, Southampton University, 1994.

[2] R. Chandrasekaran, A. Mukhopadhyay and P. Yang, "Optimal ordering policies for a capacity constrained two-item inventory system", Working Paper, The University of Texas at Dallas, 1992.

[3] R. Chandrasekaran, A. Mukhopadhyay and P. Yang, "Optimal ordering policies for a capacity constrained multi-item inventory system", Working Paper, The University of Texas at Dallas, 1992.

[4] C.A. Glass, "Feasibility of scheduling lot sizes of three products on one machine", *Management Science* **38**, 1992, 1482-1494.

[5] C.A. Glass, "Feasibility of scheduling lot sizes of two frequencies on one machine ", *European Journal of Operational Research* **75**, 1994, 354-364.

[6] R. Hassin, "A dichotomous search for a geometric random variable", *Operations Research* **32**, 1984, 423-439.

[7] R. Hassin and N. Megiddo, "Exact computation of optimal inventory policies over an unbounded horizon", *Mathematics of Operations Research* **16**, 1991, 534-546.

[8] R. Holte, L. Rosier, I. Tulchinsky and D. Varvel, "Pinwheel scheduling with two distinct numbers", *Theoretical Computer Science* **100** 1992, 105-135.

[9] R. M. Karp, "A characterization of the minimum cycle mean in a digraph", *Discrete Mathematics* **23**, 1978, 309-312.

[10] A. Mok, L. Rosier, I. Tulchinsky and D. Varvel, "Algorithms and complexity of the periodic maintenance problem", *Microprocessing and Microprogramming* **27** 1989, 657-664.

[11] R. Roundy, "98%-Effective Integer-Ratio Lot-Sizing for One-Warehouse Multi-Retailer Systems", *Management Science* **31**, 1985, 1416-1430.

[12] W. Wei and C. Liu, "On a periodic maintenance problem", *Operations Research letters* **2** 1983, 90-93.