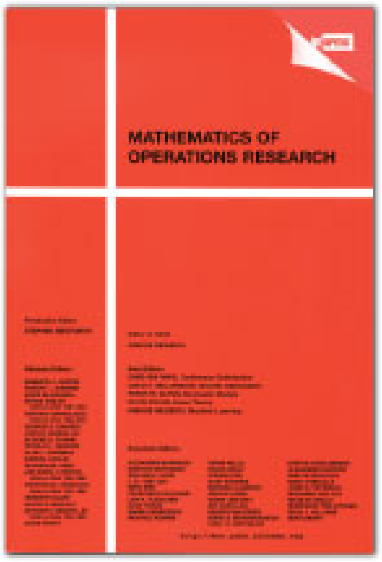

Mathematics of Operations Research

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

Minimizing Service and Operation Costs of Periodic Scheduling

Amotz Bar-Noy, Randeep Bhatia, Joseph (Seffi) Naor, Baruch Schieber,

MATHEMATICS OF OPERATIONS RESEARCH
Vol. 27, No. 3, August 2002, pp. 518–544
Printed in U.S.A.

MINIMIZING SERVICE AND OPERATION COSTS OF PERIODIC SCHEDULING

AMOTZ BAR-NOY, RANDEEP BHATIA, JOSEPH (SEFFI) NAOR, AND BARUCH SCHIEBER

We study the problem of scheduling activities of several types under the constraint that, at most, a fixed number of activities can be scheduled in any single time slot. Any given activity type is associated with a service cost and an operating cost that increases linearly with the number of time slots since the last service of this type. The problem is to find an optimal schedule that minimizes the long-run average cost per time slot. Applications of such a model are the scheduling of maintenance service to machines, multi-item replenishment of stock, and minimizing the mean response time in *Broadcast Disks*. Broadcast Disks recently gained a lot of attention because they were used to model backbone communications in wireless systems, Teletext systems, and Web caching in satellite systems.

The first contribution of this paper is the definition of a general model that combines into one several important previous models. We prove that an optimal *cyclic* schedule for the general problem exists, and we establish the NP-hardness of the problem. Next, we formulate a nonlinear program that relaxes the optimal schedule and serves as a lower bound on the cost of an optimal schedule. We present an efficient algorithm for finding a near-optimal solution to the nonlinear program. We use this solution to obtain several approximation algorithms.

(1) A $9/8$ approximation for a variant of the problem that models the Broadcast Disks application. The algorithm uses some properties of "Fibonacci sequences." Using this sequence, we present a 1.57-approximation algorithm for the general problem.

(2) A simple randomized algorithm and a simple deterministic greedy algorithm for the problem. We prove that both achieve approximation factor of 2. To the best of our knowledge this is the first worst-case analysis of a widely used greedy heuristic for this problem.

1. Introduction. We study a problem of scheduling activities of several types over an infinite number of time slots. We describe the model in terms of a generalized version of the maintenance service scheduling problem studied in Anily et al. (1998). In this formulation, there are m machines $\{1\ldots m\}$ that are to be scheduled for maintenance over an infinite discrete time horizon. In each time slot, at most M machines can be scheduled for maintenance. The cost of operating a machine at any given time slot depends on the number of time slots since the last maintenance of that machine. We assume that each machine i is associated with a constant $a_i > 0$ and the cost of operating the machine in the hth time slot after the last maintenance of that machine is $(h+b)a_i$, for $h \geq 0$ and integer $b \geq 0$. (The value of b is determined by the application.) We assume that the cost associated with the maintenance service of the ith machine is $c_i \geq 0$. The problem is to find an optimal schedule specifying at which time slots to maintain each of the machines to minimize the long-run average cost per time slot.

More formally, a schedule for the Generalized Maintenance Scheduling Problem (GMSP) with m machines is $S = S_1, S_2, \ldots$ where $S_t \subseteq \{1, \ldots, m\}$ and $|S_t| \leq M$ for all $t \geq 1$. Here, $i \in S_t$ means that machine i is scheduled for maintenance at time slot t. The maintenance cost at time slot t is $\sum_{i\in S_t} c_i$. The operating cost $o_t(j)$ of machine j at time slot t is $a_j(t-t'+b)$ for integer $b \geq 0$, where $t' \leq t$ is the largest time slot at which $j \in S_{t'}$.

Received August 8, 1997; revised June 22, 1999, November 2, 2000, and December 7, 2001.
MSC 2000 subject classification. Primary: 90B35, 68W25.
OR/MS subject classification. Primary: Production/scheduling/approximations/heuristic.
Key words. Scheduling, maintenance service, broadcast disk, cyclic schedule, approximation algorithm.

0364-765X/02/2703/0518/$05.00
1526-5471 electronic ISSN, © 2002, INFORMS

(We assume that all machines are maintained at time slot 0 in S.) If $b \neq 0$, a nonzero operating cost is incurred for machine i in every time slot; otherwise, a nonzero operating cost is incurred for machine i in every time slot, except in those time slots t where $i \in S_t$. We want to find a schedule S minimizing

$$\lim_{n\to\infty} \frac{1}{n} \sum_{t=1}^{n} \left(\sum_{i\in S_t} c_i + \sum_{j=1}^{m} o_t(j) \right).$$

Note that this limit may not exist for every schedule. However, in this paper we deal only with those schedules for which the limit exists.

As an example, let $m = 2$, $M = 1$, $b = 0$, and let the schedule for the first four time slots be $(\{1\}, \varnothing, \{2\}, \{1\})$. This schedule incurs cost $a_2 + c_1$ at time slot 1, cost $a_1 + 2a_2$ at time slot 2, cost $2a_1 + c_2$ at time slot 3, and cost $a_2 + c_1$ at time slot 4. The average cost over the first four time slots is $((a_2 + c_1) + (a_1 + 2a_2) + (2a_1 + c_2) + (a_2 + c_1))/4 = (3/4)a_1 + a_2 + (1/2)c_1 + (1/4)c_2$.

Our model has applications in many areas of computer science and operations research, as demonstrated in the following two examples. We remark that the parameter b is used for modeling the different applications. However, an optimal solution of the GMSP is not dependent on the value of b, because its effect is only an addition of a constant to the cost of any feasible solution. In the context of approximate solutions (on which our paper focuses), we provide a performance guarantee for the case where $b = 0$. Increasing the value of b can only improve the performance guarantee.

The Broadcast Disks application. The first application is in the context of Teletext systems (Ammar and Wong 1985, 1987) and the Broadcast Disks (Acharya et al. 1995, Imielinski et al. 1994, Vaidya and Hameed 1997, Acharya et al. 1995), a rapidly growing means of data dissemination in wireless asymmetric communication environments. Recently, Broadcast Disks were used to model efficient caching of Web pages. The common thread in all of these applications is a communication medium that is either unidirectional or fast in one direction but slow in the other. In this application, a database contains m equal length pages denoted by $A_1, \ldots, A_m$. These pages are accessed by clients. At each time, slot M pages are broadcasted. We associate with each page A_i a fraction a_i denoting the probability that a client wishes to access it. Clients arrive in the beginning of slots at a random location of the broadcasting schedule. A client who wishes to access page A_i listens to the broadcast until the end of the time slot in which A_i was broadcasted. The goal is to find an infinite schedule that instructs the disks which pages to broadcast at each time slot. An efficient schedule is one that minimizes the expected time spent by clients wishing to access the pages.

More formally, let $S = S_1, S_2, \ldots,$ be a broadcasting schedule. A client arriving at time slot t wishing to access file A_i has to wait $w_t(i) = t' - t + 1$ time slots where $i \in S_{t'}$ and $i \notin S_j$ for all $t \leq j < t'$. We want to find a schedule S that minimizes $\lim_{n\to\infty}(1/n) \sum_{t=1}^{n} \sum_{i=1}^{m} a_i w_t(i)$. Note that the Broadcast Disks problem is a special case of GMPS where we set $c_i = 0$ and $b = 1$. However, the two problems differ in the way costs are incurred. In the Broadcast Disks problem, the waiting cost for page i decreases with time until the next broadcast of page i. On the other hand, in the GMSP, the operating cost of a machine i increases linearly with time until the next maintenance of machine i.

The Broadcast Disks paradigm can be used to implement Web caching. Because of the speed bottleneck in using telephone lines for Web surfing, the industry has tried to add other means of communications such as cable television and satellite systems to improve the response time. However, such communication media are highly asymmetric; while communication from the base station to the receivers is very fast, reverse communication is slow or even impossible. A way of overcoming this asymmetry that is currently being tested is by limiting the Web surfing via the asymmetric media to a fixed set of "popular" pages that will be transmitted continuously—M pages at each time slot. Then, if we associate a "hit probability" with each page in the set, the goal is to schedule the pages so as to minimize the average waiting time for a requested page.

The multi-item replenishment application. Another application of our model is to the problem of multi-item replenishment of m items that was first considered in Anily et al. (1998). In this problem, at each time slot, the stock of at most M of the items may be replenished. The costs involved are item-specific linear holding costs that are incurred at the end of each time slot and item-specific ordering costs that are incurred in the time slot in which the stock of the item is replenished. Also associated with each item is a demand per time slot that is the rate at which the item is consumed. It is required to have enough inventory of each item to meet the demand before its next replenishment. For item i, let d_i be its demand per time slot, c_i be its ordering cost, and h_i be its unit holding cost per time slot. Let $a_i = d_i h_i$. The holding cost for item i, j time slots before its next replenishment is therefore ja_i. The problem is to find an optimal policy specifying at which time slots to replenish stocks of each of the items so as to minimize the long-run average cost per time slot. This problem is modeled by GMPS with $b = 0$. Many variants of this problem are considered in the literature (Hadley and Whitin 1963).

1.1. Our results. We show that there exists a *cyclic* optimal schedule for the GMSP (§2.1). This result is established by showing that we can restrict our attention to schedules over a finite state space. We establish that the decision version of the GMSP is NP-hard (§2.5) and thus consider some approximation algorithms later. The hardness result is obtained through a chain of reductions. The first is a reduction from a related problem, which we call the "Periodic Maintenance Scheduling Problem," with the GMSP; the second is a reduction from graph k-coloring to this related problem, thus proving the NP-hardness of the GMSP. The NP-hardness is established for a restricted version of the GMSP: even when $M = 1$. However, our NP-hardness result does not apply to the Broadcast Disks problem, for which all $c_i = 0$. The NP-hardness of the Broadcast Disks problem remains an open problem.

Having established the NP-hardness of the GMSP, we turn to approximation algorithms for the GMSP. Here our results broadly correspond to two cases. In the general case, the a_i-s, c_i-s and b may take on any nonnegative values. However, as mentioned earlier, the worst-case bounds for the GMSP are achieved for $b = 0$. Hence, we study the general case under the assumption $b = 0$. The other case is when $c_i = 0$ for all i and $b = 1$. This case corresponds to the Broadcast Disks application and, hence, is interesting by itself.

To establish the NP-hardness result and to design approximation algorithms, we formulate a nonlinear program that relaxes GMSP (§2.2) and is an extension of the nonlinear program in Anily et al. (1998). The solution of this nonlinear program serves as a lower bound for the optimal schedule in the analysis of the approximation ratio of our algorithms, and it is also used as a tool in some of the algorithms. We prove a lower bound on the ratio of the long-run average cost of an optimal schedule to the lower bound provided by the nonlinear relaxation. We show that there are instances of the GMSP with large M for which this ratio is at least 2 and there are instances of the Broadcast Disks for which this ratio is at least $15/14$ even for $M = 3$. These bounds can be viewed as bounds on the "integrality gap" between the fractional solution given by the nonlinear relaxation and the optimal schedule.

We present an efficient algorithm that finds a solution to the nonlinear program that is close (up to an arbitrarily small additive factor ϵ) to the optimal solution (§2.3). Thus, we avoid the usage of general techniques (e.g., interior point algorithms) for solving those nonlinear programs with costs that are prohibitive.

In §§3 and 4, we consider approximate schedules for the case $M = 1$. A ρ-approximate schedule is a schedule with a cost within a factor ρ of the cost of the optimal schedule. A ρ-approximation algorithm is an algorithm that produces a ρ-approximate schedule. In addition to a lower bound, the nonlinear program computes an optimal "fractional" schedule; that is, a schedule that may maintain fractions of machines, as long as the total of these fractions at each time slot adds up to no more than M. Our approximation algorithms can be viewed as a way of rounding this fractional schedule while still bounding the cost of the rounded schedules. In §3, we present a $9/8$-approximation algorithm for the Broadcast Disks problem, and generalize it to a 1.57-approximation algorithm for the general problem (GMSP). (Compare the 1.57 approximation algorithm for the GMSP to the $\sqrt{2}$ integrality gap for our nonlinear program for the GMSP as follows from a result of Anily et al. 1998.) These algorithms use the golden ratio sequence for generating the schedules. Constructing these sequences is not a simple and practical task. Therefore, in §4, using the solution of the nonlinear program for the given instance, we present two simple *online* algorithms that compute 2-approximate schedules for the GMSP. The first algorithm is a randomized algorithm, where the expected long-run average cost of the schedule generated is at most twice the long-run average cost of the optimal schedule. The second is a deterministic greedy algorithm. When applied to the special case of the Broadcast Disks problem, this algorithm is easy to program. Our computational experiments with this greedy algorithm show that it performs very well in practice, and our analysis shows for the first time that this greedy algorithm has a good worst-case performance as well.

Finally, in §5, we show that our algorithms can be generalized to the case in which more than one machine can be maintained concurrently ($M > 1$). We generalize both of our 2-approximation algorithms for the GMSP. However, we were able to generalize the bounds based on the golden ratio sequence only for the Broadcast Disks problem, achieving again a $9/8$ bound. For the GMSP, because we show in §2.2 that for large M the integrality gap can be as large as 2, our approximation bound (for large M) is the best possible using the techniques developed in this paper.

To summarize, our results are as follows:

- We show that the GMSP is NP-hard.
- We formulate a nonlinear program relaxation of the GMSP and develop an efficient algorithm to solve this nonlinear program (almost) optimally.
- We use golden ratio sequences to design a 1.57-approximation algorithm for the GMSP and a $(9/8)$-approximation algorithm for the Broadcast Disks problem, when $M = 1$.
- We design simple randomized and greedy 2-approximation algorithms for the GMSP when $M = 1$.
- We generalize all but the 1.57-approximation algorithm for the case when $M > 1$.

1.2. Related work. Anily et al. (1998) study the maintenance scheduling problem under the assumptions $M = 1$ and $c_i = 0$, $(i = 1 \ldots m)$. They show that there is an optimal schedule that is cyclic, and they present a polynomial time algorithm to solve the two-machine case optimally. For $m > 2$ and $b = 0$, they present a 2.5-approximation algorithm, and for the Broadcast Disks problem ($M = 1$, $b = 1$), they present a 2-approximation algorithm. They also report their computational experience with a simple greedy heuristic that seems to work well in practice. They conjecture that the problem is NP-hard. Anily et al. (1999) study the restricted version of the maintenance scheduling problem ($c_i = 0$, $M = 1$) under the additional assumption that there are only three machines. They are able to solve certain instances of this restricted version of the maintenance scheduling problem optimally, and for the other instances they present a heuristic with worst-case performance ratio of 1.0333.

Ammar and Wong (1985, 1987) consider the problem of minimizing the mean response time in Teletext systems. This problem is the same as minimizing the mean waiting time in Broadcast Disks. They show that an optimal cyclical schedule exists, and their results implicitly contain a randomized 2-approximation algorithm for this problem. This algorithm broadcasts a page of type i with probability proportional to $\sqrt{a_i}$, where a_i is the probability that a customer accesses a page of type i. They also proposed a schedule based on the golden ratio sequence. However, they did not analyze the performance of this schedule. The same problem is considered by Su and Tassiulas (1997) in the context of the Broadcast Disks. They empirically test the performance of a class of greedy heuristics for this problem. They report that the greedy policy that selects the page with largest *Mean Aggregate Delay* for broadcasting has the best performance in most cases. Furthermore, this policy produces schedules with mean response times that are close to a lower bound on the mean response time Ammar and Wong (1985, 1987). The greedy rule that is used by our algorithm is similar to the one in the heuristic of Anily et al. (1998) and is the same as the *Mean Aggregate Delay* rule described in Su and Tassiulas (1997). Additional papers containing analysis of similar types of problems are Chan and Chin (1992, 1993), Glass (1992, 1994), Hassin and Megiddo (1991), Holte et al. (1992), Roundy (1985), Wei and Liu (1983).

Following our work, Kenyon and Schanabel (1999) considered the generalized maintenance scheduling problem in the case where the maintenance times are machine dependent; that is, when different machines require different number of slots for their maintenance. They show that the problem is NP-hard even if maintenance costs are zero, and they provide a randomized 3-approximation algorithm for the case where $M = 1$.

2. Characterizing the optimal solution.

In this section, we prove several properties regarding the optimal solution. We first show that there exists a *cyclic* optimal schedule. Next, we present a nonlinear formulation that relaxes the GMSP. The main result of this section is an efficient algorithm for computing a near optimal solution to the nonlinear program. The algorithm finds a solution that is within some arbitrarily small ϵ (additive) of the optimal value. Next, we consider the integrality gap and prove a lower bound on the ratio of the long-run average cost of an optimal schedule to the lower bound provided by the nonlinear relaxation. We then establish the NP-hardness of the GMSP. As mentioned earlier, this does not establish the NP-hardness of the Broadcast Disks problem.

Without loss of generality, we assume that $c_i > 0$ implies $a_i > 0$, for all machines ($i = 1 \ldots m$). This follows from the fact that a machine i for which $c_i > 0$ and $a_i \leq 0$ is never maintained in an optimal schedule.

2.1. Existence of a cyclic optimal schedule.

A maintenance schedule $S = S_1, S_2, \ldots$ is *cyclic* if there exists a finite time period T such that $S_t = S_{t+T}$, for any time slot t; that is, S is completely determined from $S_1, \ldots S_T$. Such a sequence $S_1 \ldots S_T$ is said to *generate* the schedule. If T is the smallest time slot for which $S_1 \ldots S_T$ generates the schedule S, then $S_1 \ldots S_T$ is called a *basic cycle* of S.

We sketch the proof of the following theorem since a similar proof is given in Anily et al. (1998) for the special case of the GMSP where all $c_i = 0$.

THEOREM 1. *There exists an optimal cyclic schedule for the GMSP.*

PROOF (SKETCH). As in Anily et al. (1998), it can be shown that it is sufficient to consider schedules in which the number of time slots between two consecutive maintenance services of any machine is bounded from above. Hence, the number of distinct states that the schedules can have is finite, where a state corresponds to the m tuple $\{s_1, s_2 \ldots s_m\}$, such that s_i is equal to the number of time slots since the last maintenance of machine i in this state. Therefore, a schedule that is generated by the finite cycle with minimum average cost in the finite state space will be an optimal cyclic schedule. □

2.2. Lower bounds. The following proposition follows from our definition of operating cost.

PROPOSITION 2. *Suppose that machine i is maintained at time slot t' and is not maintained in time slots $t'+1, \ldots, t'+x-1$. Then, the total operating cost incurred by machine i in the x time slots $t', \ldots, t'+x-1$ is given by $(a_i/2)(x+(2b-1))x$.*

PROOF. Machine i incurs an operating cost of $a_i(j+b)$ in time slot $t'+j$ $(j=0\ldots x-1)$. Therefore, the total operating cost incurred by machine i in time slots $t', \ldots, t'+x-1$ is given by

$$\sum_{j=0}^{x-1} a_i(j+b) = \frac{a_i}{2}(x+(2b-1))x. \quad \square$$

We set $p=2b-1$ (hence, $p \geq -1$). Similar to Anily (1998), we consider a nonlinear relaxation of the GMSP. This relaxation provides a lower bound on the long-run average cost of the optimal schedule. Consider the following nonlinear program with variables $\tau_1, \ldots, \tau_m$.

$$\begin{aligned} (1) \qquad \text{minimize} \quad & \frac{1}{2}\sum_{i=1}^{m} a_i(\tau_i+p)+\sum_{i=1}^{m} c_i/\tau_i, \\ \text{subject to} \quad & \sum_{i=1}^{m}(1/\tau_i) \leq M \\ & \tau_i \geq 1 \quad (1 \leq i \leq m). \end{aligned}$$

Intuitively, in the nonlinear program we schedule each of the m machines in fixed periods such that the average cost is minimized. This is a relaxation because these periods may be fractional and may not be achievable simultaneously for all machines.

THEOREM 3. *The optimal solution value of the nonlinear program* (1) *is a lower bound on the long-run average cost of the optimal schedule.*

PROOF. For simplicity, we present the proof for $M=1$. The generalization is straightforward. Consider a relaxation of the problem in which we look for a cyclic schedule and assume that any number of machines can be maintained in a time slot, as long as the total number of maintenances is bounded by the cycle size. The variables in the relaxed problem are integers T and $n_1, \ldots, n_m$, where T is the length of the cycle, and n_i denotes the number of times machine i is maintained during the cycle, for $i=1, \ldots, m$. There is a single constraint that $\sum_{i=1}^{m} n_i \leq T$. The objective is to minimize the average cost per unit time.

Consider machine i. Let $\tau^{(j)}$, for $j=1, \ldots, n_i$, denote the number of time slots between consecutive maintenances of machine i during the cycle. Then, by Proposition 2, the total operating cost of machine i is $(a_i/2)\sum_{j=1}^{n_i}(\tau^{(j)}+p)\tau^{(j)}$. The total maintenance cost of machine i is $n_i \cdot c_i$. We now further relax the problem and allow variables $\tau^{(j)}$ to be fractional.

Because we allow any number of machines to be scheduled for maintenance at a time slot, the maintenance schedule for each machine can be optimized independently, apart from the constraint on the sum of the n_is. This function is optimized by taking equidistant periods $\tau_i = T/n_i$. In this case, the constraint $\sum_{i=1}^{m} n_i \leq T$ translates to $\sum_{i=1}^{m}(1/\tau_i) \leq 1$. The average cost associated with machine i is $(n_i a_i(\tau_i+p)\tau_i)/2T + n_i c_i/T = a_i(\tau_i+p)/2 + c_i/\tau_i$, yielding that the total average cost of the relaxed problem is bounded from below by the solution value of the nonlinear relaxation (1). $\square$

2.3. Computing a near optimal solution of the nonlinear program. We now present the main result of this section, which is to compute a near optimal solution to the nonlinear program (1). We find a solution that is within a small additive constant of the optimal value. Our approach avoids the usage of general techniques (e.g., interior point algorithms) for solving nonlinear programs that are rather expensive.

We first characterize the optimal solution to the nonlinear program (1).

THEOREM 4. *An optimal solution to nonlinear relaxation* (1) *is given by* $\tau_i = \max\{1, \sqrt{2(c_i+\lambda^*)/a_i}\}$ $(i = 1\ldots m)$, *where* $\lambda^* \geq 0$. *If* $\lambda^* > 0$ *then* $\sum_{i=1}^m (1/\tau_i) = M$.

PROOF. We obtain the following nonlinear program by applying Lagrangian relaxation to Equation (1).

$$(2) \qquad \text{minimize } \frac{1}{2}\sum_{i=1}^m a_i(\tau_i+p) + \sum_{i=1}^m \frac{c_i}{\tau_i} - \lambda\left(M - \sum_{i=1}^m \frac{1}{\tau_i}\right) - \sum_{i=1}^m \mu_i(\tau_i - 1).$$

For any fixed $\lambda \geq 0$ and $\mu_i \geq 0$ $(i = 1\ldots m)$, the optimal value of the nonlinear program (2) is a lower bound on the optimal value of the nonlinear program (1), and an optimal solution of (2) is given by $\tau_i = \sqrt{2(c_i+\lambda)/(a_i - 2\mu_i)}$ $(i = 1\ldots m)$, provided that $a_i - 2\mu_i > 0$ (by taking partial derivatives with respect to τ_i).

We now fix $\lambda = 0$, $\mu_i = 0$ $(i = 1\ldots m)$, and set $\tau_i = \max\{1, \sqrt{2c_i/a_i}\}$ $(i = 1\ldots m)$. This assignment may not be a feasible solution for program (1) because it may be the case that $\sum_{i=1}^m (1/\tau_i) > M$. In this case, we increase λ until the latter constraint is satisfied and define $\lambda^* > 0$ to be the value of λ for which

$$\sum_{i=1}^m \frac{1}{\tau_i} = \sum_{i=1}^m \min\left\{1, \sqrt{\frac{a_i}{2(c_i+\lambda^*)}}\right\} = M.$$

In the case $\sum_{i=1}^m (1/\tau_i) \leq M$, we set $\lambda^* = 0$.

For all i such that $\sqrt{2(c_i+\lambda)/a_i}$ is strictly less than 1, we set μ_i to the positive value for which $\sqrt{2(c_i+\lambda)/(a_i-2\mu_i)} = 1$. Note that for this choice of λ and μ_i $(i = 1\ldots m)$, we have that

$$\sqrt{\frac{2(c_i+\lambda)}{a_i - 2\mu_i}} = \max\left\{1, \sqrt{\frac{2(c_i+\lambda)}{a_i}}\right\} \quad (i = 1\ldots m),$$

and for the assignment $\tau_i = \max\{1, \sqrt{2(c_i+\lambda^*)/a_i}\}$ $(i = 1\ldots m)$, we have $\mu_i(\tau_i - 1) = 0$ and $\lambda\big(M - \sum_{i=1}^m (1/\tau_i)\big) = 0$. This implies that $\tau_i = \max\{1, \sqrt{2(c_i+\lambda^*)/a_i}\}$ $(i = 1\ldots m)$ is an optimal solution for both (2) and (1), thus completing the proof. ☐

LEMMA 5. *The term* λ^* *is bounded by* $\lambda^* \leq a_i m^2/(2M^2)$, *for some* $1 \leq i \leq m$.

PROOF. Consider again the nonlinear program (2) obtained by applying Lagrangian relaxation to (1). It follows from the proof of Theorem 4 that $\lambda^* > 0$ implies that $\sum_{i=1}^m (1/\tau_i) = M$. This implies that there exists some τ_i such that $\tau_i \leq m/M$. However, from Theorem 4, we have $\tau_i = \max\{1, \sqrt{2(c_i+\lambda^*)/a_i}\}$. Therefore, $\lambda^* \leq a_i m^2/(2M^2) - c_i$. Because $c_i \geq 0$, the desired result follows. ☐

Let $y(\lambda)$ denote the objective function of (1) as a function of λ (obtained by substituting for τ_i in terms of λ). We call a λ feasible if $\sum_{i=1}^m \min\{1, \sqrt{a_i/(2(c_i+\lambda))}\} \leq M$. Let $y'(\lambda)$ denote the derivative of $y(\lambda)$ with respect to λ.

LEMMA 6. *For all feasible* λ *for which* $y(\lambda)$ *is differentiable, we have* $0 \leq y'(\lambda) \leq M/2$.

PROOF. Let $\tau_i(\lambda) = \max\{1, \sqrt{2(c_i+\lambda)/a_i}\}$ $(i = 1\ldots m)$. We note that only functions $\tau_i(\lambda)$ for which $\tau_i(\lambda) = \sqrt{2(c_i+\lambda)/a_i}$, $1 \leq i \leq m$, contribute to $y'(\lambda)$. To simplify notation, we assume without loss of generality that this is, indeed, the case for all i, $1 \le i \le m$. A simple calculation shows

$$y'(\lambda) = \sum_{i=1}^{m} \left(\frac{1}{2} a_i - \frac{c_i}{(\tau_i(\lambda))^2} \right) \cdot \frac{d\tau_i}{d\lambda},$$

where $\lambda \le (a_i/2 - c_i)$ implies that $d\tau_i/d\lambda = 0$, and $d\tau_i/d\lambda = 1/(a_i \tau_i(\lambda))$ otherwise. Note that $0 \le \lambda \le (a_i/2 - c_i)$ implies that

$$\left(\frac{1}{2} a_i - \frac{c_i}{(\tau_i(\lambda))^2} \right) = \frac{1}{2} a_i - c_i \ge 0,$$

and $\lambda > (a_i/2 - c_i)$ implies that

$$\left(\frac{1}{2} a_i - \frac{c_i}{(\tau_i(\lambda))^2} \right) = \frac{\lambda}{(\tau_i(\lambda))^2},$$

which shows that $y'(\lambda) \ge 0$ for $\lambda \ge 0$. Also, note that because

$$y'(\lambda) = \sum_{i=1}^{m} \left(\frac{1}{2} a_i - \frac{c_i}{(\tau_i(\lambda))^2} \right) \cdot \frac{d\tau_i}{d\lambda},$$

we have $y'(\lambda) \le (1/2) \sum_{i=1}^{m} a_i (d\tau_i/d\lambda)$. Thus, $y'(\lambda) \le (1/2) \sum_{i=1}^{m} 1/\tau_i(\lambda)$. However, for feasible λ, we have by definition $\sum_{i=1}^{m} 1/\tau_i(\lambda) \le M$. This implies that $0 \le y'(\lambda) \le M/2$ for feasible λ. □

The next corollary follows from Lemma 6 by the continuity of the function $y(\lambda)$.

COROLLARY 7. *For feasible λ_1, λ_2, we have $|y(\lambda_1) - y(\lambda_2)| \le |\lambda_1 - \lambda_2| \cdot M/2$.*

THEOREM 8. *A feasible solution to the nonlinear program* (1) *whose cost is within an arbitrarily small additive factor ϵ of the cost of the optimal solution of the nonlinear program* (1) *can be found in time polynomial in* $\log(1/\epsilon)$, m, $\max_i\{\log(a_i)\}$, *and* $\max_i\{\log(c_i)\}$.

PROOF. Recall from the proof of Theorem 4 that either $\lambda^* > 0$, in which case

$$\sum_{i=1}^{m} \min\left\{1, \sqrt{\frac{a_i}{2(c_i + \lambda^*)}}\right\} = M,$$

or $\lambda^* = 0$, which is the case when

$$\sum_{i=1}^{m} \min\left\{1, \sqrt{\frac{a_i}{2c_i}}\right\} \le M.$$

Note that the function (of λ) $\sum_{i=1}^{m} \min\{1, \sqrt{a_i/(2(c_i + \lambda))}\}$ is a monotonically decreasing function of λ.

The above results imply that either λ^* is 0, or it can be approximated to within an additive ξ by doing a binary search on the range of feasible values of λ (given by Lemma 5). Note that if we find a feasible value for λ such that $\lambda - \xi$ is not feasible, then this implies $\lambda - \xi < \lambda^* \le \lambda$. It follows from Lemma 6 and Corollary 7 that $y(\lambda) - y(\lambda^*) \le M\xi/2$. Recall that $y(\lambda^*)$ is the optimal value of the objective function of (1). Thus, setting $\xi \le 2\epsilon/M$ yields the theorem.

However, there is a minor problem. We cannot exactly answer the question if a given λ is (in)feasible in polynomial time because of the precision required in the square root computations. Note that the square root of a number N can be computed with a precision of δ in time polynomial in $\log N$ and $\log(1/\delta)$.

Let $\lambda_{\max} = am^2/(2M^2)$, where $a = \max_i\{a_i\}$. Define

$$f(\lambda) = \sum_{i=1}^{m} \frac{1}{\tau_i(\lambda)},$$

where $\tau_i(\lambda) = \max\{1, \sqrt{2(c_i+\lambda)/a_i}\}$ $(i = 1 \ldots m)$. Function f can be approximated arbitrarily close by a function g such that for all $\lambda \in [0, \lambda_{\max}]$,

$$|f(\lambda) - g(\lambda)| \leq \delta.$$

Let the error introduced in the computation of λ when using function g in the binary search be denoted by $\zeta = \zeta(\delta)$. The modified binary search proceeds as follows. If $g(\lambda) > M$, then $\lambda^* < \lambda + \zeta$, otherwise $\lambda^* > \lambda - \zeta$. The binary search terminates when we shrink the feasible range of λ to $[\lambda_1, \lambda_2]$ such that, say, $\lambda_2 - \lambda_1 \leq 3\zeta$.

We now elaborate on bounding the error when approximating $f(\lambda)$. A classical theorem (Dahlquist and Björck 1974, p. 240–241) states that the attainable accuracy ζ satisfies

$$\zeta \leq \frac{\delta}{|f'(\lambda)|}.$$

There is a slight technical obstacle in applying this theorem in our case because f is not always differentiable. However, we note that the error in computing $f(\lambda)$ comes only from functions $\tau_i(\lambda)$ for which $\tau_i(\lambda) = \sqrt{2(c_i+\lambda)/a_i}$, $1 \leq i \leq m$. Therefore, we can apply the theorem to f restricted to such functions τ_i. To simplify notation, we assume without loss of generality that indeed for all i, $\tau_i(\lambda) = \sqrt{2(c_i+\lambda)/a_i}$. For $0 \leq \lambda \leq \lambda_{\max}$,

$$f'(\lambda) = \sum_{i=1}^{m} \frac{1}{2(c_i+\lambda)} \cdot \sqrt{\frac{a_i}{2(c_i+\lambda)}} \geq \sum_{i=1}^{m} \frac{1}{2(c_i+\lambda_{\max})} \cdot \sqrt{\frac{a_i}{2(c_i+\lambda_{\max})}}.$$

Thus, the error ζ is polynomially dependent on $\log 1/\delta$ and the binary representation of the numbers in the problem instance. The total error in estimating λ^* can, therefore, be bounded by 3ζ. It follows from Lemma 6 and Corollary 7 that for $\lambda_1 \leq \lambda \leq \lambda_2$, $y(\lambda) - y(\lambda^*) \leq 3M\zeta/2$. Setting $3M\zeta/2$ to be at most ϵ yields the desired bound.

Thus, by doing a modified binary search, in time polynomial in m, $\log(1/\epsilon)$, $\log c$, and $\log a$, we can find a solution to the nonlinear program (1) whose cost is within an additive factor of ϵ of the cost of the optimal solution. The result follows. $\square$

2.4. Integrality gaps. The following lemma bounds the ratio of the long-run average cost of the optimal schedule to the lower bound provided by the nonlinear relaxation in Theorem 3. It shows that there are instances of the GMSP with large M for which this ratio is at least 2, and there are instances of the Broadcast Disks for which this ratio is at least 15/14 even for $M = 3$. These bounds can be viewed as bounds on the "integrality gap" between the fractional solution given by the nonlinear relaxation and the optimal schedule.

LEMMA 9. *There exist instances of the GMSP for which the long-run average cost of the optimal schedule is at least $2M/(M+1)$ times the lower bound in Theorem 3. For the Broadcast Disks, there exist instances for which the long-run average cost of the optimal schedule is at least $2M(M+2)/((2M+1)(M+1))$ times the lower bound in Theorem 3.*

PROOF. Consider a GMSP instance with $c_i = 0, (i = 1 \ldots m)$ in which $a_1 = a_2 = \cdots = a_{M+1} = a$ and $a_{M+2} = a_{M+3} = \cdots = a_m = 1$, where a is chosen to be a very large number. Note that the long-run average cost of the optimal schedule for this instance is at least

$$a(b+1) + \sum_{i=1}^{M} ab = a((M+1)b+1).$$

This follows because at most M of the machines $1, 2, \ldots, M+1$ are scheduled in any time slot. Therefore, in any time slot, at least one of the M machines incurs an operating cost of $a(b+1)$ and the rest of the machines each incurs an operating cost of at least ab. For this instance of the GMSP, the optimal solution of the nonlinear relaxation (1) is given by

$$\tau_j = \frac{\sum_{i=1}^{m} \sqrt{a_i}}{M\sqrt{a_j}}.$$

(see Theorem 4.) By our choice, a is very large compared to m. Hence, by neglecting the low order $o(1)$ terms in the expression for τ_j, we get

$$\tau_1 = \tau_2 = \cdots = \tau_{M+1} = 1 + \frac{1}{M},$$

and

$$\tau_{M+2} = \tau_{M+3} = \cdots = \tau_m = \sqrt{a}\left(1 + \frac{1}{M}\right).$$

It follows that for this instance of the GMSP, the lower bound in Theorem 3 is given by (neglecting low order $o(a)$ terms):

$$\begin{aligned}(M+1)\frac{a}{2}\left(1 + \frac{1}{M} + p\right) &= \frac{(M+1)a}{2}\left(2b + \frac{1}{M}\right)\\ &= (M+1)a\left(b + \frac{1}{2M}\right).\end{aligned}$$

We get that for this instance of the GMSP, the ratio of the long-run average cost of the optimal schedule to the lower bound in Theorem 3 is at least

$$\frac{(M+1)b+1}{(M+1)\left(b+\frac{1}{2M}\right)}.$$

Setting $b = 0$ we get the desired result for the GMSP.

Note that the above GMSP instance can be converted to an instance of the Broadcast Disks problem by scaling down all the a_js such that they sum up to 1 and by setting $b = 1$. The above proof still carries over to this new instance of the GMSP (Broadcast Disks). By setting $b = 1$ in the above expression, for the ratio of the cost of the optimal schedule to the cost of the lower bound, we get the desired result for the Broadcast Disks problem. □

COROLLARY 10. *There exist instances of the GMSP with large M for which the ratio of the long-run average cost of the optimal schedule to the lower bound in Theorem* 3 *is arbitrarily close to* 2, *and for the Broadcast Disks there exist instances even with $M = 3$ for which this ratio is arbitrarily close to* $15/14$.

REMARK 11. A result of Anily et al. (1998) implies a $\sqrt{2}$ gap between the lower bound in Theorem 3 and the cost of the optimal schedule for the GMSP, even for $M = 1$.

2.5. NP-hardness results. To prove the hardness of the GMSP, we define and prove the hardness of a similar problem and then reduce this problem to GMSP. Consider the following Periodic Maintenance Scheduling Problem (PMSP):

> Given m machines and service intervals (integers) $l_1, l_2, \ldots, l_m$ such that $\sum_{i=1}^{m} 1/l_i \leq 1$ does there exist an infinite maintenance service schedule of these machines in which consecutive services of machine i are exactly l_i time slots apart and no more than one machine is serviced in a single time slot?

Note that a schedule for the PMSP instance is completely determined by specifying, for each machine, one time slot where it is to be scheduled for maintenance. For machine i, this time slot will be denoted by v_i.

LEMMA 12. *A given integer vector $\vec{v}$ determines a feasible schedule for the PMSP instance, if and only if, the following holds for any pair of machines i and j: $v_i - v_j \neq (0 \bmod \gcd(l_i, l_j))$.*

PROOF. Let $v_i - v_j = \gcd(l_i, l_j) \cdot q$ for some integer q. It follows from basic number theory that there exist integers b_i and b_j such that $l_j \cdot b_j - l_i \cdot b_i = \gcd(l_i, l_j) \cdot q$. Therefore, $l_j \cdot b_j + v_j = l_i \cdot b_i + v_i$. Hence, at time slot $l_j \cdot b_j + v_j$, both machines i and j are scheduled for servicing. Hence, $\vec{v}$ does not determine a feasible schedule for the PMSP instance.

If $\vec{v}$ does not specify a feasible schedule for the PMSP instance, then there must exist a pair of machines i and j such that in the schedule determined by $\vec{v}$, both these machines are serviced in a single time slot. Therefore, there exist a pair of integers b_i and b_j such that $l_i \cdot b_i + v_i = l_j \cdot b_j + v_j$. However, then $l_i \cdot b_i - l_j \cdot b_j = v_j - v_i$. Because $l_i \cdot b_i - l_j \cdot b_j = (0 \bmod \gcd(l_i, l_j))$, it follows that $v_i - v_j = (0 \bmod \gcd(l_i, l_j))$. $\square$

THEOREM 13. *The Periodic Maintenance Scheduling problem is NP-complete.*

PROOF. PMSP is obviously in NP. Given the vector $\vec{v}$, we can use Lemma 12 to verify in polynomial time if $\vec{v}$ determines a feasible schedule for the given PMSP instance.

We show that graph coloring can be reduced to PMSP, thus, showing that it is NP-complete. Given a graph G and a number k, we create an instance of PMSP such that the graph is k-colorable if and only if there exists a feasible schedule to the PMSP instance.

Let G have n vertices. By a nonedge, we mean an edge that is present in the complement of G (hence, not present in G). Without loss of generality, we can assume that G does not have any vertex of degree $n-1$.

The reduction works as follows. Every nonedge of G is assigned a weight that is a distinct prime greater than n. The weight of a vertex of G is k times the product of the weights assigned to its incident nonedges. For the ith vertex of G, we create machine i whose value l_i equals the weight of vertex i. Notice that if (i, j) is an edge of G, then $\gcd(l_i, l_j) = k$ and if (i, j) is a nonedge of G, then $\gcd(i, j) = pk$ where p is the prime assigned to nonedge (i, j). Note that each $l_i > n$ and hence $\sum_{i=1}^{n} 1/l_i < 1$.

We first show that if the graph is k-colorable, then there exists a feasible schedule to the PMSP instance. Let us assume that the color classes are numbered $0, 1, \ldots k-1$. Let color class i have n_i vertices. These vertices in color class i are numbered $0, 1, 2, \ldots, n_i - 1$. Note that a vertex is uniquely determined by its number and the number of the color class it belongs to.

We define a vector $\vec{v}$ of length n the components of which are integers lying in the range $[0, nk)$. The component of $\vec{v}$ for the machine corresponding to the jth vertex in the ith color class is $jk + i$. Note that since $j < n$ and $i < k$ for any machine, $jk + i < nk$. Below we show that $\vec{v}$ determines a feasible schedule for the PMSP instance; that is, we show that $v_i - v_j \neq (0 \bmod \gcd(l_i, l_j))$ for any two machines i and j (Lemma 12). There are the following two cases to be considered:

Case 1. i and j correspond to vertices that have no edge between them in G. In this case, $\gcd(l_i, l_j) = pk$ for some prime $p > n$. Therefore, $pk > nk$. But $|v_i - v_j| < nk$. Therefore, $v_i - v_j \neq (0 \bmod \gcd(l_i, l_j))$.

Case 2. i and j correspond to vertices that have an edge between them in G. In this case, $\gcd(l_i, l_j) = k$ and i and j must belong to different color classes, say b_i and b_j, respectively. Hence, $v_i = \alpha \cdot k + b_i$ and $v_j = \beta \cdot k + b_j$, for some integers α and β. Hence, $v_i - v_j = (b_i - b_j \bmod k)$. However, $b_i - b_j \neq (0 \bmod k)$, because $|b_i - b_j| < k$ and $b_i \neq b_j$. Hence, $v_i - v_j \neq (0 \bmod \gcd(l_i, l_j))$.

We now show that if there exists a vector $\vec{v}$ that specifies a feasible schedule to the PSMP instance, then G is k-colorable. We can assume without loss of generality that all components of $\vec{v}$ are nonnegative. We color G by assigning vertex i to color class $v_i \bmod k$. Note that this coloring uses at most k colors. This will be a valid coloring if for any two vertices in a color class, there is no edge between them in G. To obtain a contradiction, let the vertices corresponding to machines i and j, which are assigned to the same color class have an edge between them in G. Because they are assigned to the same color class, $v_i \bmod k = v_j \bmod k$, and hence $v_i - v_j = 0 \bmod k$. Note that $\gcd(l_i, l_j) = k$ because vertices i and j have an edge between them in G. However, then by Lemma 12, $\vec{v}$ does not specify a feasible schedule for the PMSP instance, a contradiction. Hence, G is k-colorable. □

We now return to the NP-hardness proof of GMSP.

THEOREM 14. *Given an instance of the GMSP and a bound C, it is NP-hard to decide whether there exists a solution to this instance with a long-run average cost at most C. The hardness result holds even when $M = 1$.*

PROOF. To prove the hardness of the GMSP, we reduce the PMSP to the GMSP as follows. Given an instance of the PMSP, we define an instance of the decision version of the GMSP with $M = 1$ by setting $a_i = 2$, $c_i = l_i^2$, for $i = 1 \ldots m$, and $C = \sum_{i=1}^{m}(l_i + p) + \sum_{i=1}^{m} l_i$ where $p = 2b - 1$. In the proof of Theorem 3, we showed that the average cost incurred by machine i in any feasible schedule is at least $(1/2)a_i(\tau_i + p) + c_i/\tau_i$ for some $\tau_i \geq 1$. The function $(1/2)a_i(\tau_i + p) + c_i/\tau_i$ has a unique minimum. This minimum is attained when $\tau_i = \sqrt{2c_i/a_i} = l_i$. These facts imply that the long-run average cost of this GMSP instance is at least C and that in any feasible schedule of this GMSP instance of long-run average cost C, the average cost incurred for machine i is exactly $(l_i + p) + l_i$. We show that this GMSP instance has average cost C if and only if the instance of PMSP has a feasible schedule with the service intervals $l_1, \ldots, l_m$.

If the PMSP instance has a feasible schedule S, then the same schedule is also feasible for the GMSP instance and its long-run average cost is exactly C. In the other direction, suppose there exists a schedule S for the GMSP instance with cost, at most, C. Following §2.1, we may assume that S is cyclic. To show that S is also feasible for the PMSP instance, we show below that any two consecutive services of machine i in S are exactly l_i apart.

Let T be a basic cycle of S; hence T has finite length. The long-run average cost of machine i in S, and hence the average cost of machine i in T, is exactly $(l_i + p) + l_i$. Let $\alpha_1, \alpha_2 \ldots \alpha_{n_i}$ be the number of time slots between adjacent services of machine i in T. Note that $\sum_{j=1}^{n_i} \alpha_j = |T|$. The total cost incurred for machine i in the basic cycle T is (Proposition 2)

$$\frac{1}{2}\sum_{j=1}^{n_i} a_i(\alpha_j + p)\alpha_j + n_i c_i = \frac{1}{2}a_i \sum_{j=1}^{n_i} \alpha_j^2 + \frac{1}{2}a_i p|T| + n_i c_i.$$

Let $x_i = |T|/n_i$ and let $\alpha_j = x_i + \delta_j$. Note that $\sum_{j=1}^{n_i} \delta_j = 0$. We first show that $\delta_j = 0$, $(j = 1 \ldots n_i)$. If some $\delta_j \neq 0$, then the total cost incurred for machine i in the basic cycle T

$$\begin{aligned}\frac{1}{2}a_i \sum_{j=1}^{n_i}(x_i + \delta_j)^2 + \frac{1}{2}a_i p|T| + n_i c_i &= \frac{1}{2}a_i \sum_{j=1}^{n_i}\big(x_i^2 + \delta_j^2\big) + \frac{1}{2}a_i p|T| + n_i c_i \\ &> \frac{1}{2}a_i n_i x_i^2 + \frac{1}{2}a_i p|T| + n_i c_i.\end{aligned}$$

(The first equality follows from $\sum_{j=1}^{n_i} \delta_j = 0$.) Hence, if some $\delta_j \neq 0$, then the average cost for machine i in the basic cycle T is strictly greater than $(1/2)a_i(x_i + p) + c_i/x_i$, which leads to a contradiction since for this instance of the GMSP ($a_i = 2$ and $c_i = l_i^2$, for $i = 1 \dots m$)

$$\frac{1}{2}a_i(x_i + p) + \frac{c_i}{x_i} \geq (l_i + p) + l_i.$$

It follows that $\alpha_j = x_i$, ($j = 1 \dots m$). Therefore, the average cost incurred by machine i in schedule S is $\frac{1}{2}a_i(x_i + p) + c_i/x_i$, which is equal to $(l_i + p) + l_i$ if and only if $x_i = l_i$. Therefore, any two consecutive services of machine i in S are exactly l_i apart. □

3. Approximation algorithms based on golden ratio sequences. In this section, we provide nontrivial approximation algorithms for the case where $M = 1$. In §5, we generalize the algorithm to the case where more than one machine can be served concurrently ($M > 1$) only for the Broadcast Disks problem ($b = 1$ and $c_i = 0$).

Fix $M = 1$. For the Broadcast Disks problem ($b = 1$ and $c_i = 0$), we present a 9/8-approximation algorithm. For the GMSP ($b = 1$ and $c_i \geq 0$), we generalize this bound to a 1.57-approximation algorithm. The algorithms construct a schedule in which the service frequencies, denoted by $f_i = 1/\tau_i$, are those given by the optimal solution to the nonlinear program (1). However, unlike the lower bound, the gaps between two consecutive appearances of the same machine in the schedule may have several values. The construction is based on the golden ratio sequence. In this sequence, there are at most three values for the gaps between two consecutive appearances of the same machine i, and all three are close to the optimal periods (τ_i). The golden ratio sequence was developed in Hofri and Rosberg (1987) and Itai and Rosberg (1984) and is based on an open address hashing method introduced in Knuth (1973).

The golden ratio sequence is based on the Fibonacci numbers (Definition 1). In our proofs, we use some well-known properties about Fibonacci numbers (Lemma 15) and some identities and inequalities on Fibonacci numbers (Lemma 16). Let $\phi = (1 + \sqrt{5})/2$ be the golden ratio number ($\phi \approx 1.61803$). See Graham et al. (1989, Chapter 6.6) for more details on the Fibonacci numbers.

Definition 1. The Fibonacci numbers are defined as follows: $F_0 = 0$, $F_1 = 1$ and, recursively, $F_k = F_{k-1} + F_{k-2}$ for $k \geq 2$. (The sequence is 0, 1, 1, 2, 3, 5, 8, 13, ...)

Lemma 15. *The value of the kth Fibonacci number is* $F_k = (\phi^k - (1-\phi)^k)/\sqrt{5}$. *Moreover*:
- *The ratio* F_{2i+1}/F_{2i} *is a monotonically decreasing function on* i, *the limit of which is* ϕ.
- *The ratio* F_{2i}/F_{2i-1} *is a monotonically increasing function on* i, *the limit of which is* ϕ.

Lemma 16. *For* $k \geq 2$ *and* $1 \leq j \leq k-2$:
(1) $F_j^2 + F_{j+1}^2 - F_{j+2}^2 = -2F_jF_{j+1}$.
(2) $F_k = F_{j+1}F_{k-j} + F_jF_{k-j-1} = F_{j+2}F_{k-j-1} + F_{j+1}F_{k-j-2}$.
(3) $F_{j+1}^2F_{k-j-2} + F_{j+2}^2F_{k-j-1} = F_{j+1}F_k + F_jF_{j+2}F_{k-j-1}$.

Proof. (1) The first identity follows by squaring the recursive equation $F_{j+2} = F_j + F_{j+1}$.
(2) The second identity can be proven by repeatedly applying the recursive definition.
(3) To prove the third identity, we expand one of the F_{j+2} in the term $F_{j+2}^2F_{k-j-1}$ to get

$$\begin{aligned} F_{j+1}^2F_{k-j-2} + F_{j+2}^2F_{k-j-1} &= F_{j+1}F_{j+1}F_{k-j-2} + F_{j+1}F_{j+2}F_{k-j-1} + F_jF_{j+2}F_{k-j-1} \\ &= F_{j+1}(F_{j+1}F_{k-j-2} + F_{j+2}F_{k-j-1}) + F_jF_{j+2}F_{k-j-1}. \end{aligned}$$

The claim follows using identity 2. □

The following definition is needed for the next theorem that characterizes the golden ratio sequence. Let $i_1, i_2, \ldots, i_t$ be a sequence of the numbers $1, \ldots, m$. Suppose that i appears in positions $r < s$ in the sequence and does not appear in positions $r+1, \ldots, s-1$. We say that the *gap* between these two appearances is $s-r$. For i that appears N_i times in the sequence, there are N_i gaps. The last gap is computed assuming that the sequence is cyclic.

THEOREM 17 (HOFRI AND ROSBERG 1987). *Let $g_1, \ldots, g_m$ be frequencies such that $\sum_{i=1}^m g_i = 1$, and let F_k be a Fibonacci number such that $g_i = N_i/F_k = (F_{k-j_i} + S_i)/F_k$ for some $1 \le j_i \le k-2$ and $0 \le S_i < F_{k-j_i-1}$ ($\sum_{i=1}^m N_i = F_k$ and for some integers N_i). Then, there exists a sequence of length F_k of the numbers $1, \ldots, m$ such that i appears N_i times in the sequence. Furthermore, for each $1 \le i \le m$, there are at most three values for the gaps between two consecutive appearances of i in the sequence*:

- *S_i gaps of value F_{j_i},*
- *$F_{k-j_i-2} + S_i$ gaps of value F_{j_i+1},*
- *$F_{k-j_i-1} - S_i$ gaps of value F_{j_i+2}.*

Note that there are $S_i + (F_{k-j_i-2} + S_i) + (F_{k-j_i-1} - S_i) = F_{k-j_i} + S_i = N_i$ gaps. Note also that the sum of all gaps is the length of the sequence $F_{j_i} S_i + F_{j_i+1}(F_{k-j_i-2} + S_i) + F_{j_i+2}(F_{k-j_i-1} - S_i) = F_k$ (by Identity 2 of Lemma 16).

The golden ratio sequence.

Input: $f_1, \ldots, f_m$ such that $\sum_{i=1}^m f_i = 1$. (Recall that in our case, $f_i = 1/\tau_i$, where $\tau_1, \ldots, \tau_m$ is the optimal solution of (1).)

Construction: Let F_k be a Fibonacci number. Define $N_i = \lfloor f_i F_k \rfloor$ or $N_i = \lceil f_i F_k \rceil$ such that $\sum_{i=1}^m N_i = F_k$. Then $(N_i - 1)/F_k \le f_i \le (N_i + 1)/F_k$. Let $g_i = N_i/F_k$. Then, $N_i/(N_i+1) \le g_i/f_i \le N_i/(N_i-1)$. For given $f_1, \ldots, f_m$, as k tends to infinity the N_i values become larger. Hence, the ratio g_i/f_i tends to one. Consequently, for any given $\varepsilon > 0$, we can find a large enough F_k such that

$$1 - \varepsilon \le \frac{g_i}{f_i} \le 1 + \varepsilon. \tag{3}$$

The value of ε will be fixed later.

Output: For these frequencies $g_1, \ldots, g_m$ determined by ε and F_k, we construct the golden ratio sequence of length F_k as defined in Theorem 17.

3.1. A 9/8-approximation algorithm for the Broadcast Disks problem.

THEOREM 18. *The golden ratio sequence yields a 9/8-approximation schedule for the Broadcast Disks problem.*

PROOF. Consider an instance of the Broadcast Disks problem with pages $A_1, \ldots, A_m$ and access probabilities $a_1, \ldots, a_m$. Let $\tau_1, \ldots, \tau_m$ be the optimal solution to (1). Let $f_1 = 1/\tau_1, \ldots, f_m = 1/\tau_m$ that we denote as optimal frequencies. As shown in Theorem 3, the lower bound for the average operating cost for an instance of the Broadcast Disks problem is $(1/2)\sum_{i=1}^m (\tau_i + 1)a_i$. For the golden ratio sequence schedule, we show the existence of a sequence $\beta_1, \ldots, \beta_m$ with the following two properties:

- The upper bound on the expected operating cost is $(1/2)\sum_{i=1}^m (\beta_i + 1)a_i$.
- $(\beta_i + 1)/(\tau_i + 1) \le 9/8$ for $1 \le i \le m$.

Therefore, the following ratio is an upper bound for the approximation of the golden ratio schedule:

$$\frac{\frac{1}{2}\sum_{i=1}^m (\beta_i + 1)a_i}{\frac{1}{2}\sum_{i=1}^m (\tau_i + 1)a_i} \le \frac{9}{8}. \tag{4}$$

To prove the bound $(\beta_i+1)/(\tau_i+1) \le 9/8$, we do the following. By definition and by inequality (3), $\tau_i = 1/f_i \ge (1-\varepsilon)/g_i$. Denote $\psi_i = 1/g_i$. By choosing a large enough F_k, we can make ε small enough such that

$$\frac{\beta_i+1}{\tau_i+1} \le \frac{\beta_i+1}{1/g_i - \varepsilon/g_i + 1} \le \frac{\beta_i}{\psi_i}.$$

We complete the proof by showing in Lemma 7 that already $\beta_i/\psi_i \le 9/8$. □

For ease of presentation and because the following applies to all i, we omit the subscript i.

LEMMA 19. *Let $r = \beta/\psi$, then $r \le 9/8$.*

PROOF. We first compute ψ. Because $g = (F_{k-j}+S)/F_k$, it follows that

$$\psi = \frac{1}{g} = \frac{F_k}{(F_{k-j}+S)}. \tag{5}$$

By definition $j \le k-2$. First, consider the case $j = k-2$. Because $S = 0$ when $j = k-2$, it follows that in this case, i appears exactly once in the sequence and contributes $F_k(F_k+1)/F_k = F_k+1$ to the coefficient of $(1/2)a$ in the numerator of inequality (4) while computing the operating cost. Therefore, $\beta = F_k = \psi$ and, in this case, the ratio $r = 1$. From now on, we assume that $j \le k-3$.

We now bound β. In the periodic schedule of length F_k, a gap of size x contributes $x(x+1)/2$ to the coefficient of a/F_k in calculating the average operating cost. Hence, by the golden ratio sequence properties (Theorem 17), we have

$$\beta+1 \le \frac{S\cdot F_j(F_j+1) + (F_{k-j-2}+S)F_{j+1}(F_{j+1}+1) + (F_{k-j-1}-S)F_{j+2}(F_{j+2}+1)}{F_k}.$$

Note that the $\frac{1}{2}$ factor disappears because it exists in both the upper bound and the lower bound in inequality (4). Rearranging the terms, we get

$$\beta+1 \le \frac{\left(F_j^2+F_{j+1}^2-F_{j+2}^2\right)S + \left(F_{j+1}^2F_{k-j-2}+F_{j+2}^2F_{k-j-1}\right)}{F_k} + \frac{S\cdot F_j + (F_{k-j-2}+S)F_{j+1} + (F_{k-j-1}-S)F_{j+2}}{F_k}.$$

The second term in the right-hand side equals 1 because the coefficients of S cancel each other and due to the second identity of Lemma 16. Applying the first and the third identities of Lemma 16 on the first term we get,

$$\beta \le \frac{F_{j+1}F_k + F_jF_{j+2}F_{k-j-1} - 2F_jF_{j+1}S}{F_k}.$$

Substituting the above upper bound on β and the value of ψ from Equation (5), we have

$$r \le \frac{(F_{k-j}+S)(F_{j+1}F_k + F_jF_{j+2}F_{k-j-1} - 2F_jF_{j+1}S)}{F_k^2}. \tag{6}$$

Rearranging the terms, we get

$$r \le \frac{F_{j+1}F_{k-j}F_k + F_jF_{j+2}F_{k-j-1}F_{k-j}}{F_k^2} + \frac{F_{j+1}F_k + F_jF_{j+2}F_{k-j-1} - 2F_jF_{j+1}F_{k-j}}{F_k^2}S - \frac{2F_jF_{j+1}}{F_k^2}S^2.$$

We look at r as a function of S and find the maximum value of r by solving the equation $r'(S) = 0$. We get

$$F_{j+1}F_k + F_jF_{j+2}F_{k-j-1} - 2F_jF_{j+1}F_{k-j} - 4F_jF_{j+1}S = 0.$$

It follows that the maximum value of r occurs at $S = S_{\max}$, where

$$S_{\max} = \frac{F_{j+1}F_k + F_jF_{j+2}F_{k-j-1} - 2F_jF_{j+1}F_{k-j}}{4F_jF_{j+1}}.$$

Substituting the value of $S_{\max}$ in the upper bound for r and rearranging the terms, we get the following upper bound.

CLAIM 20:

$$r \le \frac{(F_{j+1}F_k + F_jF_{j+2}F_{k-j-1} + 2F_jF_{j+1}F_{k-j})^2}{8F_jF_{j+1}F_k^2}.$$

PROOF. Let $a = F_j$, $b = F_{j+1}$ $c = F_{j+2}$, $x = F_{k-j-1}$, $y = F_{k-j}$ and $z = F_k$. Then,
(1) $S_{\max} = (bz + acx - 2aby)/(4ab)$,
(2) $-2abS_{\max} = (2aby - bz - acx)/2$,
(3) $y + S_{\max} = (2aby + bz + acx)/(4ab)$,
(4) $bz + acx - 2abS_{\max} = (2aby + bz + acx)/2$.
Hence, by inequality (6)

$$r \le \frac{(y+S)(bz + acx - 2abS_{\max})}{z^2} = \frac{(bz + acx + 2aby)^2}{8abz^2}.$$

Putting the Fibonacci terms back in the inequality completes the proof. □
Using the second identity of Lemma 16, we get that

$$2F_jF_{j+1}F_{k-j} = 2F_j(F_k - F_jF_{k-j-1}) = 2F_jF_k - 2F_j^2F_{k-j-1}.$$

Therefore, the term inside the parentheses in the upper bound for r (Claim 20) can be rewritten as

$$\begin{aligned}
F_{j+1}F_k + F_jF_{j+2}F_{k-j-1} + 2F_jF_{j+1}F_{k-j} &= (F_{j+1} + 2F_j)F_k + (F_{j+2} - 2F_j)F_jF_{k-j-1}\\
&= (F_{j+2} + F_j)F_k + F_{j-1}F_jF_{k-j-1}\\
&= (F_{j+2} + F_j)F_k + F_{j-1}(F_k - F_{j+1}F_{k-j})\\
&= (F_{j+2} + F_j + F_{j-1})F_k - F_{j-1}F_{j+1}F_{k-j}\\
&= F_{j+3}F_k - F_{j-1}F_{j+1}F_{k-j}.
\end{aligned}$$

It remains to prove that for $1 \le j \le k-3$

$$r \le \frac{(F_{j+3}F_k - F_{j-1}F_{j+1}F_{k-j})^2}{8F_jF_{j+1}F_k^2} \le \frac{9}{8}.$$

We prove this in the next lemma for $k \ge 8$. □

LEMMA 21. *For $k \ge 8$ and $1 \le j \le k-3$,*

$$(F_{j+3}F_k - F_{j-1}F_{j+1}F_{k-j})^2 \le 9F_jF_{j+1}F_k^2.$$

PROOF. We prove the inequality directly for $j = 1$ and $j = 2$. For $j = 1$,

$$(F_{j+3}F_k - F_{j-1}F_{j+1}F_{k-j})^2 = (F_4F_k)^2 = 9F_k^2 = 9F_jF_{j+1}F_k^2.$$

For $j = 2$,

$$(F_{j+3}F_k - F_{j-1}F_{j+1}F_{k-j})^2 = (5F_k - 2F_{k-2})^2 = (3F_k + 2F_{k-1})^2$$

and

$$9F_jF_{j+1}F_k^2 = 18F_k^2.$$

Therefore, we need to show that $3F_k + 2F_{k-1} \le 3\sqrt{2}F_k$. By rearranging the terms, we need to show that

$$\frac{F_k}{F_{k-1}} \ge \frac{2}{3\sqrt{2}-3} \approx 1.60948.$$

By Lemma 15 and because $\phi \ge 1.60948$, it follows that the above inequality is true for odd k. The above inequality is true for $k = 8$ and, therefore, true for even k by Lemma 15.

We now claim that $F_{j+1}F_{k-j} \ge 0.7F_k$ for $2 \le j \le k-3$ and $k \ge 8$. For $j = 2$ and $j = 3$, this inequality could be verified using Lemma 15 as was done before. For $j > 3$, we prove this claim by a double induction on k and j using the recursive definition of the Fibonacci numbers. Therefore, to complete the proof we need to show that

$$(F_{j+3}F_k - 0.7F_{j-1}F_k)^2 \le 9F_jF_{j+1}F_k^2.$$

Dividing the above inequality by F_k^2, it remains to show that

$$(F_{j+3} - 0.7F_{j-1})^2 \le 9F_jF_{j+1}.$$

Applying the recursive definition, we get $F_{j+3} - 0.7F_{j-1} = 1.3F_{j+1} + 1.7F_j$. Thus, we want to prove that

$$(1.3F_{j+1} + 1.7F_j)^2 \le (1.3 + 1.7)^2F_jF_{j+1}.$$

By rearranging the above inequality as a function of F_{j+1}/F_j, it follows that the above inequality holds if $F_{j+1}/F_j \le 2.89/1.69$. Again, because $\phi < 2.89/1.69$ and because this inequality is true for $j = 4$, we can use Lemma 15 to complete the proof. □

3.2. A 1.57-approximation algorithm for the GMSP.

THEOREM 22. *The golden ratio sequence yields a 1.57-approximation schedule for the GMSP.*

PROOF. We assume that $p = -1$ $(b = 0)$. This is because b only appears as an additive term in the objective function of the nonlinear program relaxation (1) with a minimum value of $b = 0$. Hence, for $b > 0$, the performance guarantee of the algorithm can only be better. Without loss of generality, let $\tau_1 \le \tau_2 \le \cdots \le \tau_m$ be the (almost) optimal periods computed for the given GMPS instance as shown in Theorem 8. Let r be the approximation factor of our schedule. Depending on the value of τ_1, we distinguish among the following three cases:

Case 1. $\tau_1 \ge 1.283$: In this case, no machine requires frequency greater than $1/1.283$. Here, we use the golden ratio sequence algorithm directly to find the schedule. It can be shown that the cost of the schedule thus obtained is

$$\frac{1}{2}\sum_{i=1}^{m} a_i(x_i - 1) + \sum_{i=1}^{m} \frac{c_i}{\tau_i},$$

and $x_i \le (9/8)\tau_i$ (see Lemma 19). On the other hand, as shown earlier, the optimal value of the nonlinear program relaxation (1) is a lower bound on the cost of the optimal solution that is within a arbitrarily small additive constant (ϵ) of the following term. Note that for computing approximation ratios, we can ignore additive constants, hence we will assume that the lower bound is given by

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

Because the coefficients of c_i are the same in both our schedule and the lower bound, the ratio of the coefficients of a_i in our schedule to these coefficients in the lower bound serves as an upper bound on the approximation factor. Thus,

$$r \le \frac{x_i - 1}{\tau_i - 1} \le \frac{\frac{9}{8}(\tau_i - 1) + \frac{1}{8}}{\tau_i - 1} \le \frac{9}{8} + \frac{\frac{1}{8}}{0.283} < 1.57.$$

For the other two cases, we first choose a number $x \ge 2$ (depending on the given τ_1 value) and then in adjacent blocks of x time slots, we schedule machine 1 in the first $x-1$ time slots. The other machines are scheduled in the remaining time slots by using the golden ratio sequence algorithm. Note that in the new schedule, we have only $1/x$ fraction of time slots left over to schedule machines $2, \ldots, m$, while in the optimal schedule, they may require $1 - 1/\tau_1$ fraction of the slots. So, we have to scale $\tau_2, \ldots, \tau_m$ by the factor $\max\{1, (\tau_1 - 1)x/\tau_1\}$. We now use the golden ratio sequence algorithm to find a schedule for machines $2, \ldots, m$ in the $1/x$ fraction of empty slots.

We define the following three parameters:

- Let α denote the ratio of the operating cost of machine 1 in this schedule to the operating cost of machine 1 in the optimal schedule. It follows that because $b = 0$ $(p = -1)$

$$\alpha = \frac{1/x}{(\tau_1 - 1)/2} = \frac{2}{x(\tau_1 - 1)}.$$

- Let γ denote the ratio of the maintenance cost of machine 1 in this schedule to the maintenance cost of machine 1 in the optimal schedule. It follows that

$$\gamma = \frac{(x-1)/x}{1/\tau_1} = \frac{(x-1)\tau_1}{x}.$$

- Let $\beta = (9/8)\max\{1, (\tau_1 - 1)x/\tau_1\}$.

Lemma 23. *The cost of the schedule obtained by this algorithm is at most*

$$\max\left\{\alpha, \gamma, \beta + \frac{\beta - 1}{3.533}, 1\right\}$$

times the cost of the optimal schedule.

Proof. The four terms among which we choose the maximum represent all possible bounds on the approximation factor of the algorithm. By definition, α and γ bound the ratio of the operating cost and the maintenance cost of machine 1 in the schedule to the corresponding costs of this machine in the optimal solution. Lemma 24 shows that the ratio of the operating cost of machines $2, \ldots, m$ in the schedule to the cost of these machines in the optimal solution is, at most, $\beta + (\beta - 1)/3.533$. Finally, the same lemma shows that the maintenance cost of machines $2, \ldots, m$ is no more than their maintenance cost in the optimal schedule. ☐

Case 2. $1.00085 \le \tau_1 < 1.283$: In this case, we verified using a computer program that for every value of τ_1, there exists a number $2 \le x \le 1{,}500$ such that

$$r \le \max\left\{\alpha, \gamma, \beta + \frac{\beta - 1}{3.533}\right\} < 1.57.$$

Our program computes, for every integer value of x in the range $2 \le x \le 1{,}500$, the range of values for τ_1 for which each of α, γ, and $\beta + (\beta - 1)/3.533$ is less than 1.57. Let $\tau_1(x)$ denote the range of τ_1 values for x. We used the program to ensure that $\cup_{x=2}^{1,500} \tau_1(x)$ contains the range $1.00085 \le \tau_1 < 1.283$.

Case 3. $1 < \tau_1 < 1.00085$: In this case, we set $x = \lfloor x^* \rfloor$ where

$$x^* = \frac{2}{1.566} \cdot \frac{\sqrt{\tau_1}}{\tau_1 - 1}.$$

Let

$$\alpha^* = \frac{2}{x^*(\tau_1 - 1)} = \frac{1.566}{\sqrt{\tau_1}}.$$

Note that $1.565 \le \alpha^* \le 1.566$. Substituting α^* we get

$$\alpha = \frac{2}{x(\tau_1 - 1)} = \frac{x^*}{x} \cdot \frac{2}{x^*(\tau_1 - 1)} = \frac{x^*}{x}\alpha^* \le \left(1 + \frac{1}{x^* - 1}\right)\alpha^*.$$

Note that

$$\beta = \frac{9}{8} \max\left\{1, \frac{(\tau_1 - 1)x}{\tau_1}\right\} \le \frac{9}{8} \max\left\{1, \frac{(\tau_1 - 1)x^*}{\tau_1}\right\} = \frac{9}{8}\frac{2}{1.566\sqrt{\tau_1}} \le 1.437.$$

Thus, we have

$$\beta + \frac{\beta - 1}{3.533} \le 1.437 + \frac{0.437}{3.533} \le 1.561 < \alpha^*.$$

Also note that

$$\gamma = \frac{(x-1)\tau_1}{x} \le \tau_1 \le 1.00085 \le \alpha^*.$$

Hence,

$$r \le \max\left\{\alpha, \gamma, \beta + \frac{\beta - 1}{3.533}\right\} \le \left(1 + \frac{1}{x^* - 1}\right)\alpha^* \le \alpha^* + \frac{\alpha^*}{x^* - 1}.$$

Because $x^* \ge 1501$, we get that $r \le \alpha^* + 0.002 \le 1.57$. $\square$

LEMMA 24. *The operating cost of machines* $2, \ldots, m$ *in the schedule obtained by the golden ratio sequence algorithm for GMPS is, at most,* $\beta + (\beta - 1)/3.533$ *times their operating cost in the optimal schedule. Furthermore, their maintenance cost in the algorithm is no more than their maintenance cost in the optimal schedule.*

PROOF. Recall that $\beta = (9/8)\max\{1, (\tau_1 - 1)x/\tau_1\}$. Because we used the golden ratio sequence to schedule the machines $2, \ldots, m$ in $1/x$ fraction of the time slots left over after scheduling machine 1, and because for $2 \le i \le m$, we scaled all the τ_i by a factor $\max\{1, (\tau_1 - 1)x/\tau_1\}$, the cost of the schedule for machines $2, \ldots, m$ is

$$\frac{1}{2}\sum_{i=2}^{m} a_i(\tau_i' - 1) + \sum_{i=2}^{m} \frac{c_i}{\tau_i''},$$

where $\tau_i \le \tau_i' \le \tau_i\beta$ and $\tau_i'' \ge \tau_i$. Hence, the maintenance cost of machines $2, \ldots, m$ is no more than their maintenance cost in the optimal schedule, proving the second part of the lemma. To prove the first part, we use the following bound on the operating cost of machines $2, \ldots, m$.

$$\text{(7)} \qquad \frac{1}{2}\sum_{i=2}^{m} a_i(\tau_i' - 1) \le \frac{1}{2}\sum_{i=2}^{m} a_i(\beta\tau_i - 1) = \frac{\beta}{2}\sum_{i=2}^{m} a_i(\tau_i - 1) + \frac{\beta - 1}{2}\sum_{i=2}^{m} a_i.$$

Because $\sum_{i=1}^{m} 1/\tau_i \le 1$ and $\tau_1 < 1.283$, it follows that $\tau_i > 4.533$ for $2 \le i \le m$. Substituting these values of τ_i in the lower bound to the optimal solution given by the nonlinear program (1) (as before, ignoring the small additive constant ϵ), we get that the cost of the optimal solution is at least $(3.533/2)\sum_{i=2}^{m} a_i$. Clearly, the cost of the optimal solution is at least $(1/2)\sum_{i=2}^{m} a_i(\tau_i - 1)$. Therefore, the cost of the optimal solution is at least

$$\max\left\{\frac{1}{2}\sum_{i=2}^{m} a_i(\tau_i - 1),\ \frac{3.533}{2}\sum_{i=2}^{m} a_i\right\}. \tag{8}$$

By dividing the coefficient of each machine in the first summand in (7) by the corresponding coefficient in the first operand of the max operation in (8), we get that the first summand in (7) is at most β times the optimal solution. By dividing the coefficient of each machine in the second summand in (7) by the corresponding coefficient in the second operand of the max operation in (8), we get that the second summand in (7) is at most $(\beta - 1)/3.533$ times the optimal solution. Summing these two factors yields the proof. □

4. Practical 2-Approximation Algorithms for the GMSP. In this section, we present simple approximation algorithms for the GMSP. Recall that the approximation algorithms presented in §3 use golden ratio sequences, the construction of which is not a simple and practical task. All the algorithms presented in this section use a simple rule to decide which machine to schedule in the current time slot. Similar algorithms have been shown to perform well in practice (Ammar and Wong 1985, Anily et al. 1998, Su and Tassiulas 1997, Vaidya and Hameed 1997). We show for the first time that these algorithms also have a bounded worst-case performance. The approximation algorithms presented in this section are for the GMSP with $M = 1$. We show in §5 that these algorithms can be generalized to the case in which more than one machine can be maintained concurrently ($M > 1$), with the same approximation bounds.

4.1. A randomized approximation algorithm. Let OPT denote the long-run average cost of the optimal schedule. As shown in Theorem 8, we can find, in polynomial time $\tau_1, \ldots, \tau_m$ such that $(1/2)\sum_{i=1}^{m} a_i(\tau_i + p) + \sum_{i=1}^{m} c_i/\tau_i \le OPT + \epsilon$, for some arbitrarily small ϵ, and such that $\sum_{i=1}^{m} 1/\tau_i \le 1$. The randomized algorithm is as follows:

Randomized algorithm:

At time slot t, $(t = 1, 2 \ldots)$:
Schedule machine i for maintenance with probability $1/\tau_i$;
(Note: For a given time slot, the selection of machines for maintenance is
done in a mutually exclusive way.)

THEOREM 25. *The expected long-run average cost of schedule S generated by the randomized algorithm is, at most, $2 \cdot OPT + \epsilon$ for some arbitrarily small ϵ.*

PROOF. Let n be any large positive number. Let X_i be a random variable, which denotes the total cost incurred by machine i over the first n time slots in schedule S. Let $X = \sum_{i=1}^{m} X_i$. Then, X is the total cost of schedule S over the first n time slots. Note that by linearity of expectations, the expected cost is $E(X) = \sum_{i=1}^{m} E(X_i)$. Let T_i be a random variable that is one more than the number of maintenances of machine i in schedule S in the first n time slots. Let the random variables $Y_i(j)$ $(j = 1, \ldots, T_i)$ denote the periods between maintenances of machine i; that is, $Y_i(j) = r_j - r_{j-1}$ where r_j is the time slot of the jth maintenance of machine i in schedule S (by our assumption, the 0th maintenance of any machine in schedule S occurs at time slot 0). The random variables $Y_i(j)$ are identically distributed independent random variables for machine i. The value $A(Y_i(j))$ represents the operating cost incurred by machine i between its $(j-1)$th and jth maintenances and is given by $A(Y_i(j)) = a_i Y_i(j)(Y_i(j)+p)/2$. The total maintenance cost of machine i in the first n time slots of schedule S is $(T_i - 1)c_i < T_i c_i$. Define a random variable

$$Z_i = \sum_{j=1}^{T_i} A(Y_i(j)) + T_i c_i.$$

Note that $Z_i \geq X_i$ and, therefore, $E(Z_i) \geq E(X_i)$. In Lemma 26, we prove that

$$E(Z_i) = na_i\left(\tau_i + \frac{p-1}{2}\right) + nc_i/\tau_i + a_i\tau_i\left(\tau_i + \frac{p-1}{2}\right) + c_i.$$

Therefore, the expected long-run average cost of S over first n time slots, for large n is

$$\lim_{n\to\infty} E(X/n) = \lim_{n\to\infty} \sum_{i=1}^{m} E(X_i)/n \leq \lim_{n\to\infty} \sum_{i=1}^{m} E(Z_i)/n = \sum_{i=1}^{m} a_i\left(\tau_i + \frac{p-1}{2}\right) + \sum_{i=1}^{m} c_i/\tau_i,$$

which is at most $2\cdot(OPT+\epsilon) = 2\cdot OPT + \epsilon'$ because $p \geq -1$, for some arbitrarily small ϵ'. $\square$

The proof of the next lemma is similar to the proof of Wald's identity (Feller 1967).

LEMMA 26. $E(Z_i) = na_i(\tau_i + (p-1)/2) + nc_i/\tau_i + a_i\tau_i(\tau_i + (p-1)/2) + c_i$.

PROOF. Let us define an infinite set of random variables:

$$I_j = \begin{cases} 1 & \text{if } j = 1. \\ 1 & j \geq 2 \text{ and the } (j-1)\text{th maintenance of machine } i \text{ in } S \\ & \quad \text{occurs at or before time slot } n. \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, we can write

$$Z_i = \sum_{j=1}^{T_i} A(Y_i(j)) + T_i c_i = \sum_{j=1}^{\infty} I_j A(Y_i(j)) + c_i \sum_{j=1}^{\infty} I_j.$$

Note that I_j is completely independent of $Y_i(j), Y_i(j+1), \ldots$.

$$E(Z_i) = E\left(\sum_{j=1}^{\infty} I_j A(Y_i(j))\right) + c_i E\left(\sum_{j=1}^{\infty} I_j\right) = \sum_{j=1}^{\infty} E(I_j A(Y_i(j))) + c_i E\left(\sum_{j=1}^{\infty} I_j\right).$$

The last equality follows from the fact that $\sum_{j=1}^{\infty} I_j A(Y_i(j))$ is finite and by the linearity of expectation. But I_j is independent of $Y_i(j)$. Therefore, $E(Z_i) = \sum_{j=1}^{\infty} E(I_j)E(A(Y_i(j))) + c_i E(\sum_{j=1}^{\infty} I_j)$. Note that the $Y_i(j)$ are identically distributed random variables (geometrically distributed with parameter $1/\tau_i$). Hence, $E(A(Y_i(j))) = \mu_i$, for some yet to be determined value μ_i. Hence,

$$E(Z_i) = \mu_i \sum_{j=1}^{\infty} E(I_j) + c_i E\left(\sum_{j=1}^{\infty} I_j\right),$$

but $\sum_{j=1}^{\infty} I_j = T_i = T_i' + 1$, where T_i' is the number of maintenances of machine i in schedule S in the first n time slots. T_i' is binomially distributed with parameter $1/\tau_i$. Hence, $E(T_i') = n/\tau_i$ and $E(T_i) = n/\tau_i + 1$. Now we compute μ_i. Let Y be a random variable, geometrically distributed with parameter $1/\tau_i$.

$$\begin{aligned} E(A(Y)) &= E\left(a_i \frac{Y(Y+p)}{2}\right) \\ &= \frac{a_i}{2} \sum_{j=1}^{\infty} j(j-1)\left(1 - \frac{1}{\tau_i}\right)^{(j-1)} \frac{1}{\tau_i} + a_i\tau_i \frac{p+1}{2}. \end{aligned}$$

Therefore,

$$\mu_i = \frac{a_i}{2}\frac{2(1-1/\tau_i)}{(1/\tau_i)^2} + \frac{a_i\tau_i(p+1)}{2} = a_i\tau_i\left(\tau_i + \frac{p-1}{2}\right).$$

Hence,

$$\begin{aligned} E(Z_i) &= (n/\tau_i+1)a_i\tau_i\left(\tau_i+\frac{p-1}{2}\right)+\left(\frac{n}{\tau_i}+1\right)c_i \\ &= na_i\left(\tau_i+\frac{p-1}{2}\right)+n\frac{c_i}{\tau_i}+a_i\tau_i\left(\tau_i+\frac{p-1}{2}\right)+c_i. \quad \square \end{aligned}$$

4.2. Deterministic greedy algorithms. Let $\vec{s} = s_1, s_2 \dots s_m$ be an integer vector denoting the state of the system, where $s_i = k$ $(i = 1 \dots m)$ means that machine i has not been maintained for k time slots since its last maintenance. Initially, $s_i = 0$ $(i = 1 \dots m)$. Let $\tau_1, \dots, \tau_m$ be a feasible almost optimal solution of (1) computed as described in Theorem 8.

Define machine 0 as a dummy machine with $a_0 = c_0 = 0$ and $1/\tau_0 = 1 - \sum_{i=1}^m 1/\tau_i$. The dummy machine is used to ensure that in every time slot, exactly one machine is scheduled. This is achieved by scheduling dummy machine 0 in every time slot where none of the m machines $1 \dots m$ are scheduled. Note that this does not change the cost of the solution. The dummy machine is introduced for ease of presentation of the algorithm and the proofs.

The greedy heuristic works as follows.

Greedy rule—GR1:

Let $\vec{s}$ be the state at time slot $t-1$, $(t = 1, 2 \dots)$:
If $\sum_{i=1}^m 1/\tau_i < 1$
  Let machine $i \in \{0 \dots m\}$ minimize $(c_j - (s_j+1)a_j\tau_j)$ (ties broken by a consistent rule)
    Schedule machine i for maintenance at time slot t;
else
  Let machine $i \in \{1 \dots m\}$ minimize $(c_j - (s_j+1)a_j\tau_j)$ (ties broken by a consistent rule)
    Schedule machine i for maintenance at time slot t.

For the special case of GMPS, when $c_i = 0$, $(i = 1 \dots m)$, we show in Lemma 27 that the greedy rule **GR1** is equivalent to

Greedy rule—GR2:

Let $\vec{s}$ be the state at time slot $t-1$, $(t = 1, 2 \dots)$:
Let machine i maximize $a_j(s_j+1)^2$ (ties are broken by some consistent rule)
  Schedule machine i for maintenance at time slot t.

This greedy heuristic **GR2** is easy to program because we do not need to calculate $\tau_1, \dots, \tau_m$. Note that in this special case of GMPS, the total cost incurred by machine i, in case it is not maintained for s_i+1 time slots, is proportional to $(1/2)a_i(s_i+1)^2$. This is our motivation for the greedy rule **GR2**. The greedy rule of Anily et al. (1998) is based on the same motivation, and nothing was known about the worst-case performance of this heuristic. The computational results of Anily et al. (1998) and our experiments with **GR2** show that these greedy heuristics perform within 7% of the optimal. Note that the greedy rule **GR2** is the same as the “maximize Mean Aggregate Delay” rule that seems to perform well in practice as reported in Su and Tassiulas (1997). We show in Theorem 28 that the heuristics based on the greedy rule **GR1** and hence **GR2** have a bounded worst-case performance guarantee as well.

LEMMA 27. *For the special case when $c_i = 0$ $(i = 1 \dots m)$, the greedy rule* **GR1** *is equivalent to the greedy rule* **GR2**.

PROOF. For the special case when $c_i = 0$ $(i = 1 \dots m)$ and $M = 1$, the optimal solution $\tau_1, \dots, \tau_m$ of the nonlinear program relaxation (1), as follows from the result of Anily et al. (1998) and by Theorem 4, is given by $\tau_i^2 = 2\lambda/a_i$ for $\lambda > 0$. Hence, $(a_i\tau_i)^2 = 2\lambda a_i$. Therefore, $a_i\tau_i$ is directly proportional to $\sqrt{a_i}$. Hence, $(s_i+1)a_i\tau_i$ is directly proportional to $\sqrt{a_i}(s_i+1)$. Hence, we get that

$$(s_i+1)a_i\tau_i \geq (s_j+1)a_j\tau_j \Longleftrightarrow \sqrt{a_i}(s_i+1) \geq \sqrt{a_j}(s_j+1) \Longleftrightarrow a_i(s_i+1)^2 \geq a_j(s_j+1)^2. \quad \square$$

THEOREM 28. *Let $\tau_1, \dots, \tau_m$ be a feasible, almost optimal solution of* (1) *computed as described in Theorem* 8. *The long-run average cost of the schedule S generated by the greedy heuristic based on greedy rule* **GR1** *is, at most,* $\sum_{i=1}^m a_i(\tau_i + (p-1)/2) + \sum_{i=1}^m (c_i/\tau_i)$ *that by Theorem* 8 *is at most $2 \cdot OPT + \epsilon$, for some arbitrarily small ϵ.*

PROOF. We show that the total cost of the schedule S generated by the greedy heuristic over the first n time slots is at most

$$(n+1)\sum_{i=1}^m a_i\left(\tau_i + \frac{p-1}{2}\right) + (n+1)\sum_{i=1}^m \frac{c_i}{\tau_i}.$$

The proof follows by taking n arbitrarily large so that $(n+1)/n$ approaches 1.

Let the vector $\vec{s^t} = s_1^t, s_2^t \dots s_m^t$ denote the state of the system at time slot t. Define $C(t, S)$ to be the sum of the operating cost incurred in time slot $(t-1)$ and the maintenance cost in the tth time slot of schedule S. We assume that no cost is incurred in the 0th time slot. For $t \geq 0$, define $P(t, S) = \sum_{i=1}^m a_i s_i^t \tau_i$. We use $P(t, S)$ as a potential function. Note that $P(0, S) = 0$ because $s_i^0 = 0$ $(i = 1 \dots m)$. Let

$$Avg = \sum_{i=1}^m a_i\left(\tau_i + \frac{p-1}{2}\right) + \sum_{i=1}^m \frac{c_i}{\tau_i},$$

and let

$$W(t, S) = Avg - C(t, S) + P(t-1, S) - P(t, S).$$

Below, we show that $W(t, S) \geq 0$ for $t \geq 1$. By adding the inequalities $W(t, S) \geq 0$ for $1 \leq t \leq n+1$, we get

$$(n+1) \cdot Avg - \sum_{t=1}^{n+1} C(t, S) + P(0, S) - P(n+1, S) \geq 0.$$

Note that $\sum_{t=1}^{n+1} C(t, S)$ is an upper bound on the cost incurred in the first n time slots of the schedule S. Also note that $P(n+1, S) \geq 0$. Hence, the cost incurred in the first n time slots of the schedule S is at most $(n+1) \cdot Avg$, which is what we set out to prove.

We now show that $W(t, S) \geq 0$ for $t \geq 1$. Recall that machine 0 is a dummy machine with $a_0 = c_0 = 0$ and $1/\tau_0 = 1 - \sum_{i=1}^m 1/\tau_i$. For $0 \leq j \leq m$, Let S_j be the schedule that schedules the same machine as S in time slots $1, \dots, t-1$ and schedules machine j in time slot t. Hence,

$$W(t, S_j) = Avg - \sum_{i=1}^m a_i\left(s_i^{t-1} + \frac{p+1}{2}\right) - c_j + \sum_{i=1}^m a_i s_i^{t-1}\tau_i - \sum_{i \neq j,\, 1 \leq i \leq m} a_i\left(s_i^{t-1} + 1\right)\tau_i.$$

It can be verified that

$$\sum_{j=0}^m \frac{1}{\tau_j} W(t, S_j) = 0.$$

Hence, there is a machine $j \in \{0 \dots m\}$ for which $W(t, S_j) \geq 0$. Also, in case $\sum_{i=1}^m 1/\tau_i = 1$, there exists such a machine $j \in \{1 \dots m\}$.

Note that we can write

$$W(t, S_j) = W(t, S_0) - \big(c_j - a_j(s_j^{t-1}+1)\tau_j\big).$$

Recall that in case $\sum_{i=1}^{m} 1/\tau_i = 1$, the greedy rule **GR1** schedules the machine $j \in \{1\dots m\}$ that minimizes $c_k - a_k(s_k^{t-1}+1)\tau_k$. It follows that this machine j also maximizes $W(t, S_k)$ for $k \in \{1\dots m\}$ and, therefore, $W(t, S_j) \geq 0$.

In the other case when $\sum_{i=1}^{m} 1/\tau_i < 1$, the greedy rule **GR1** schedules the machine $j \in \{0\dots m\}$ that minimizes $c_k - a_k(s_k^{t-1}+1)\tau_k$. It follows that this machine j also maximizes $W(t, S_k)$ for $k \in \{0\dots m\}$ and, therefore, $W(t, S_j) \geq 0$. Thus, it follows that if S is the schedule generated by the greedy heuristic, then $W(t, S) \geq 0$ for $t \geq 1$. □

5. Generalization to $M > 1$. In this section, we show that our algorithms can be generalized to the case where more than one machine can be maintained concurrently ($M > 1$). We generalize both of our 2-approximation algorithms for the GMSP. Note that Lemma 9 shows that this is the best possible approximation for the GMSP (for large M) by using the lower bound in Theorem 3. Therefore, we are able to generalize the bounds based on the golden ratio sequence only for the Broadcast Disks problem.

5.1. The Broadcast Disks problem. To get the 9/8 bound for the case $M > 1$, we generalize Theorem 17 as follows. Let $g_1, \dots, g_m$ be frequencies such that $\sum_{i=1}^{m} g_i = M$ and $g_i < 1$ for $1 \leq i \leq m$. Let F_k be a Fibonacci number such that $g_i = N_i/F_k = (F_{k-j_i} + S_i)/F_k$ for some $1 \leq j_i \leq k-2$ and $0 \leq S_i < F_{k-j_i-1}$. Then, $\sum_{i=1}^{m} N_i = MF_k$. The following theorem defines the *generalized golden ratio sequence*. This theorem is not cited in Itai and Rosberg (1984) and Hofri and Rosberg (1987), but can be inferred straightforwardly by the way the original golden ratio sequence is constructed.

THEOREM 29. *There exists a sequence of length $M \cdot F_k$ of the numbers $1, \dots, m$ such that i appears N_i times in the sequence. Furthermore, for each $1 \leq i \leq m$, there are, at most, three values for the gaps between two consecutive appearances of i in the sequence*
- *S_i gaps of value F_{j_i},*
- *$F_{k-j_i-2} + S_i$ gaps of value F_{j_i+1},*
- *$F_{k-j_i-1} - S_i$ gaps of value F_{j_i+2}.*

Consider an instance of the Broadcast Disks problem with pages $A_1, \dots, A_m$ and access probabilities $a_1, \dots, a_m$. Let $f_1 = 1/\tau_1, \dots, f_m = 1/\tau_m$ be the (almost) optimal frequencies computed using the binary search technique in Theorem 8. Without loss of generality, we may assume that $f_i < 1$ for $1 \leq i \leq m$. Otherwise, if $f_i = 1$, page A_i would be scheduled in all time slots, thus achieving an optimal waiting time.

From here on, we proceed as in the proof of Theorem 18 by applying the generalized golden ratio sequence instead of the golden ratio sequence.

THEOREM 30. *The golden ratio sequence yields a 9/8-approximation algorithm for the Broadcast Disks problem for $M \geq 1$.*

5.2. Practical 2-approximation algorithms for the GMSP.

5.2.1. Randomized approximation algorithm. To show that the randomized approximation algorithm in §4.1 can be generalized to the case $M > 1$, we demonstrate the existence of a scheduling policy in which machine i is scheduled with probability $1/\tau_i$ in any given time slot and, at most, M machines are scheduled in any given time slot. We show in Lemma 31 how to construct one such scheduling policy. The generalized randomized approximation algorithm uses this scheduling policy to schedule machine i with probability $1/\tau_i$ in time slot t, $(t = 1, 2, \ldots)$. The expected long-run average cost of the schedule generated by the generalized randomized approximation algorithm is at most twice the long-run average cost of the optimal policy, because the proof of Theorem 25 requires only that machine i be scheduled with probability $1/\tau_i$ in any given time slot.

LEMMA 31. *Given m values $\tau_i \geq 1$ $(i = 1 \ldots m)$ for which $\sum_{i=1}^{m} 1/\tau_i \leq M$, we can define an event of scheduling, at most, M out of m machines such that machine i is scheduled with probability $1/\tau_i$.*

PROOF. Without loss of generality, we can assume $\tau_i > 1$, $(i = 1 \ldots m)$. This can be achieved by scheduling the machines in the set $\{i | \tau_i = 1\}$ in all time slots.

Let $f_i = 1/\tau_i$, and let $S_i = \sum_{j=1}^{i} f_i$. Let us assign interval $I_i = [S_{i-1}, S_i)$ to machine i. In any given time slot, select a number α uniformly in $(0, 1)$, and schedule all the machines i for which $(\alpha + k) \in I_i$, for some integer k, in the given time slot. Note that at most M machines are scheduled in any given time slot. This follows because $S_m \leq M$. Also note that no machine is selected twice in any given time slot since s and $s+1$ cannot simultaneously be in I_i (since $f_i \leq 1$). Finally, note that in a given time slot, machine i is scheduled with probability f_i. $\square$

5.2.2. Deterministic greedy algorithms. The greedy rule **GR1** is generalized to the rule of scheduling at most M machines with the least $c_j - (s_j + 1)a_j\tau_j$ values for which $c_j - (s_j + 1)a_j\tau_j$ is at most 0. Similarly, the greedy algorithm **GR2** is generalized to the rule of scheduling M machines with the largest $a_j(s_j + 1)^2$ value. It follows from the proof of Lemma 27 that the generalized greedy rule **GR1** is equivalent to the generalized greedy rule **GR2**, when $c_i = 0$ $(i = 1 \ldots m)$. The proof that the heuristic based on greedy rule **GR1** is a 2-approximation essentially follows from the proof of Theorem 28 with the following modifications. Let S be the schedule constructed by this greedy heuristic. To show that $W(t, S) \geq 0$ for $t \geq 1$, we define the set F of all possible schedules that schedule the same machine as S in time slots $1, \ldots, t-1$ and schedule at most M machines in time slot t. It can be verified that

$$\sum_{S' \in F} p(S')W(t, S') = 0,$$

where $p(S')$ is any probability distribution on the schedules of set F, such that machine i is scheduled with probability $1/\tau_i$ in time slot t. That is, if $F^i \subset F$ is the set of schedules in which machine i is scheduled in time slot t, then

$$\sum_{S' \in F^i} p(S') = 1/\tau_i.$$

Such a probability distribution exists as shown in the proof of Lemma 31. Note that if $S^0 \in F$ does not schedule any machine at time slot t and $S' \in F$ schedules machines in set R at time slot t, then

$$W(t, S') = W(t, S^0) - \sum_{j \in R} \left(c_j - a_j(s_j^{t-1} + 1)\tau_j\right).$$

Hence, as shown in the proof of Theorem 28, $W(t, S) \geq 0$ for $t \geq 1$.

6. Open problems. The following issues offer directions for future research.

- Our NP-hardness proof does not apply to the case when all $c_i = 0$. In particular, the NP-hardness for the Broadcast Disks problem is not known.

- We believe that it might be possible to improve the 9/8-approximation bound for the Broadcast Disks by first determining the schedule for machine 1 based on a_1, and then scheduling all the other machines using the golden ratio sequence.
- Our analysis of the greedy algorithm establishes an approximation ratio of 2. In practice, the schedules generated by the greedy algorithm are close to the optimal schedule. Can we prove a better bound than 2 on the performance of the greedy algorithm?

Acknowledgments. The authors thank Samir Khuller for helpful discussions of minimizing service and operation costs of periodic scheduling. We also thank Rafi Hassin for providing the papers of Anily et al. (1998, 1999). We are grateful to the anonymous referees for many helpful suggestions that improved the exposition. A preliminary version of this paper appeared in the proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco, CA, 1998, pp. 11–20. This work was done while Amotz Bar-Noy was a member of the Electrical Engineering Department, Tel Aviv University, Tel Aviv 69978 Israel and was supported by the MOST Consortium administered by the Israeli Ministry of Commerce and Industry. The work of Joseph (Seffi) Naor was supported by the MOST Consortium administered by the Israeli Ministry of Commerce and Industry.

References

Acharya, S., R. Alonso, M. J. Franklin, S. Zdonik. 1995. Broadcast disks: Data management for asymmetric communications environment. *ACM SIGMOD Internat. Conf. on Management of Data (SIGMOD 95)*. San Jose, CA, 199–210.

———, M. J. Franklin, S. Zdonik. 1995. Dissemination-based data delivery using broadcast disks. *IEEE Personal Comm.* **2**(6) 50–60.

Ammar, M. H., J. W. Wong. 1985. The design of teletext broadcast cycles. *Performance Eval.* **5**(4) 235–242.

———, ———. 1987. On the optimality of cyclic transmission in Teletext systems. *IEEE Trans. Comm.* **COM-35**(1) 68–73.

Anily, S., C. A. Glass, R. Hassin. 1998. The scheduling of maintenance service. *Discrete Appl. Math.* **82** 27–42.

———, ———, ———. 1999. Scheduling of maintenance services to three machines. *Ann. Oper. Res.* **86** 375–391.

Chan, M. Y., F. Chin. 1992. General schedulers for the pinwheel problem based on double-integer reduction. *IEEE Trans. Comput.* **41**(6) 755–768.

———, ———. 1993. Schedulers for larger classes of pinwheel instances. *Algorithmica* **9** 425–462.

Dahlquist, G., Å. Björck. 1974. *Numerical Methods*. Prentice-Hall, Inc. Englewood Cliffs, NJ.

Feller, W. 1967. *An Introduction to the Theory of Probability and Its Applications*. John Wiley and Sons, New York.

Glass, C. A. 1992. Feasibility of scheduling lot sizes of three products on one machine. *Management Sci.* **38** 1482–1494.

———. 1994. Feasibility of scheduling lot sizes of two frequencies on one machine. *Eur. J. Oper. Res.* **75** 354–364.

Graham, R. L., D. E. Knuth, O. Patashnik. 1989. *Concrete Mathematics*. Addison-Wesley, Reading, MA.

Hadley, G., T. M. Whitin. 1963. *Analysis of Inventory Systems*. Prentice-Hall, Englewood Cliffs, NJ.

Hassin, R., N. Megiddo. 1991. Exact computation of optimal inventory policies over an unbounded horizon. *Math. Oper. Res.* **16** 534–546.

Hofri, M., Z. Rosberg. 1987. Packet delay under the golden ratio weighted TDM policy in a multiple-access channel. *IEEE Trans. Inform. Theory* **11–33** 341–349.

Holte, R., L. Rosier, I. Tulchinsky, D. Varvel. 1992. Pinwheel scheduling with two distinct numbers. *Theoret. Comput. Sci.* **100** 105–135.

Imielinski, T., S. Viswanathan, B. Badrinath. 1994. Energy efficient indexing on air. *ACM SIGMOD Int. Conf. on Management of Data* (SIGMOD 94), Minneapolis, MN, 25–36.

Itai, A., Z. Rosberg. 1984. A golden ratio control policy for a multiple-access channel. *IEEE Trans. Automat. Contr.*, **AC-29** 712–718.

Kenyon, C., N. Schabanel. 1999. The data broadcast problem with non-uniform transmission times. *Proc. 10th Annual ACM-SIAM Sympos. on Discrete Algorithms* (SODA 99), Baltimore, MD, 547–556.

Knuth, D. E. 1973. *The Art of Computer Programming*, vol. 1. Addison-Wesley, Reading, MA.

Roundy, R. 1985. 98%-effective integer-ratio lot-sizing for one-warehouse multi-retailer systems. *Management Sci.* **31** 1416–1430.

Su, C. J., L. Tassiulas. 1997. Broadcast scheduling for information distribution. *Proc. IEEE Infocom* (Infocom 97), Kobe, Japan, 109–117.

Wei, W., C. Liu. 1983. On a periodic maintenance problem. *Oper. Res. Lett.* **2** 90–93.

Vaidya, N., S. Hameed. 1997. Log time algorithms for scheduling single and multiple channel data broadcast. *Proc. 3rd Ann. ACM/IEEE Internat. Conf. on Mobile Comput. and Networking* (Mobicom 97), Budapest, Hungary, 90–99.

A. Bar-Noy: Computer and Information Science Department, Brooklyn College, 2900 Bedford Avenue, Brooklyn, New York 11210; e-mail: amotz@sci.brooklyn.cuny.edu

R. Bhatia: Bell Labs, Lucent Technologies, Murray Hill, New Jersey 07974; e-mail: randeep@research.bell-labs.com

J. Naor: Computer Science Department, Technion, Haifa 32000, Israel; e-mail: naor@cs.technion.ac.il

B. Schieber: IBM T. J. Watson Research Center, PO Box 218, Yorktown Heights, New York 10598; e-mail: sbar@watson.ibm.com