On Optimal Strategies for Stealing Cycles*

Sandeep N. Bhatt
Bell Communications Research
Morristown, N. J.
and
Rutgers University
New Brunswick, N. J.

Fan R. K. Chung
University of Pennsylvania
Philadelphia, Penn.

F. Thomson Leighton
MIT
Cambridge, Mass.

Arnold L. Rosenberg
University of Massachusetts
Amherst, Mass.

Abstract. The growing importance of networked workstations as a computing milieu has created a new modality of parallel computing, namely, the possibility of having one workstation *"steal cycles"* from another. In a typical episode of cycle-stealing, the owner of workstation B allows the owner of workstation A to take control of B's processor whenever it is idle, with the promise of relinquishing control immediately upon the demand of the owner of B. Typically, the costs for an episode reside in the overhead required to supply workstation B with work (data and, perhaps, the programs to process the data), coupled with the fact that work in progress when the owner of B reclaims the workstation is lost to the owner of A. The first cost militates toward supplying B with a large amount of work at once; the second cost militates toward repeatedly supplying B with small amounts of work. It is this tension that makes the problem interesting. In this paper, we formulate two models of cycle-stealing. The first model focusses on accomplishing as much work as possible during a single episode, when one knows the probability distribution of the return of B's owner. The second model

---

* A portion of this paper was presented in the minitrack on Partitioning and Scheduling for Parallel and Distributed Computation of the *28th Hawaii Intl. Conf. on System Sciences* (1995).

Authors' Addresses:
Sandeep N. Bhatt: Bell Communications Research, Morristown, NJ 07960;
Fan R. K. Chung: Dept. of Mathematics, Univ. of Pennsylvania, Philadelphia, PA;
F. Thomson Leighton: Dept. of Mathematics and Lab. for Computer Science, MIT, Cambridge, MA 02139;
Arnold L. Rosenberg: Dept. of Computer Science, Univ. of Massachusetts, Amherst, MA 01003

focusses on competing with an omniscient cycle stealer, when one knows how much work the stealer can accomplish. In both models, we study strategies that optimize, under a variety of assumptions, the total amount of work one can expect to garner from an opportunity to steal cycles.

1 Motivation

Research on parallel computing has historically focussed mainly on single machines that are endowed with many processors. The growing importance of networked workstations as a computing milieu has created an alternative modality of parallel computing, namely, the possibility of having one workstation *"steal cycles"* from another; see [1], [3]-[9]. The following scenario defines one version of this paradigm. The owner of workstation A has a massive number of mutually independent tasks that must be computed. In order to expedite the completion of the tasks, the owner of A enters a contract with the owners of (some of) the other workstations in the cluster, that allows A to take control of the processors of these workstations whenever they are idle, with the promise of relinquishing control *immediately* upon the demand of a workstation's owner (say, when the mouse or keyboard is touched). The question motivating the current study is: When workstation B becomes available, how should the owner of workstation A allocate work to B in order to maximize the total amount of work one can expect to garner from a single episode of cycle-stealing? The challenge of this problem resides in the tension created by the varied costs of an episode of cycle-stealing. The first cost resides in the (usually sizable) *fixed* portion of the overheads of supplying work to workstation B and reclaiming the results of that work: in a data-parallel situation, these fixed overheads would reside in "filling the pipe" twice, first to supply input data to B and second to receive output data from B; in the most general situation, these overheads would also include the cost of supplying B with the appropriate programs. We ignore for the moment the second, *variable*, component of the cost of supplying B with work, i.e., the per-datum portion of the cost, for in our model, we absorb this quantity into the cost of B's executing the assigned tasks. The third, and subtlest, component of the cost resides in the fact that the owner of A will lose the results of whatever work is in progress when B's owner reclaims that workstation—due to A's owner's promise to abandon workstation B *immediately* upon demand. The first of these costs would lead the owner of A to supply workstation B with a single large package of tasks, in order to incur the fixed communication cost only once in the episode; the third of these costs would lead the owner of A to supply workstation B with a sequence of small packages of tasks, in order to minimize the potential impact of the return of B's owner and the concomitant loss of all unreported work. Clearly, it is in the interests of the owner of workstation A to seek a strategy that balances the first and third costs in a way that maximizes the expected productive output of each episode of cycle-stealing.

In this paper, we formulate a mathematical model of the process of cycle-stealing and study strategies that optimize, under a variety of assumptions, the amount of work one can expect to garner from a single episode. After formalizing the framework within which we study the cycle-stealing problem (Section 2), we devote the major portion of the paper (Sections 3-5) to developing three quite distinct methods of analysis to derive optimal cycle-stealing schedules for three different models of the anticipated behavior of the owner of workstation B. We close the paper with some preliminary results on how to perform steal cycles when one does not have the detailed knowledge presupposed by our model (Section 6).

2 A Mathematical Model of Cycle-Stealing

2.1 The Relevant Notions

Lifespans. Clearly, no single strategy can suffice for all possible episodes of cycle-stealing: some episodes may arise because the owner of workstation B is on a multi-week vacation, during which s/he is (almost) certain not to appear; others may involve weekends, where the likelihood of the owner's appearing may depend on his/her current workload; yet other episodes may involve lunch breaks of limited but unpredictable duration; others yet could involve telephone calls, almost certainly of short duration. Accordingly, we consider two *scenarios*, or classes of episodes, that require somewhat different groundrules; and, within these scenarios, we allow different probability distributions on the "risk" of the return of the owner of workstation B. In the *unbounded lifespan* scenario, the owner of workstation A has no *a priori* bound on how long workstation B will remain idle; in the *bounded lifespan* scenario, the owner of workstation A is informed that workstation B will be idle for at most L time units (say, an hour or an night or a week). In both scenarios, the owner of workstation A is given information about the *a priori* probability distribution on the "risk" of the return of the owner of workstation B. In Sections 3, 4, and 5, we assume that s/he has total information about the distribution; in Section 6, we assume that s/he has virtually none, but instead has detailed knowledge about how much work an omniscient schedule can accomplish.

Work schedules. The owner of workstation A partitions the lifespan of workstation B (during the current episode of cycle-stealing) into a *schedule*, i.e., a sequence

$$\mathcal{S} = t_0, t_1, t_2, \ldots$$

of *periods*, the ith of which has finite length $t_i \geq 0$. The intended interpretation is as follows. At time τ_k (to be specified imminently), the kth period begins: the owner of workstation

A supplies workstation B with a *job* containing an amount of work chosen[1] so that t_k time units are sufficient for

- (the owner of) workstation A to send the work to workstation B;
- workstation B to perform the work;
- workstation B to return the results of the work.

If $k = 0$, then $\tau_k = 0$; if $k > 0$, then

$$\tau_k = T_{k-1} \stackrel{\text{def}}{=} t_0 + t_1 + \cdots t_{k-1}.$$

In analyzing cycle-stealing schedules, it is often useful to strip a schedule $\mathcal{S}$ of its first few periods. The following operation achieves this. The *kth tail* of a schedule $\mathcal{S} = t_0, t_1, \ldots$, denoted $\phi_k(\mathcal{S})$, is the schedule $\phi_k(\mathcal{S}) = t_k, t_{k+1} \ldots$; note that $\phi_0(\mathcal{S}) = \mathcal{S}$.

Communication costs. The communications that begin and end each period, wherein the owner of workstation A supplies workstation B with work and workstation B returns the results of that work, incur a fixed *communication overhead* of c time units, independent of the amount of data transmitted in these transactions. This overhead results from some combination of the following actions.

- A shared memory scenario:
  - Workstation A sends workstation B a message "telling it" where to get the data and/or programs it needs to perform its assigned tasks (say, in a shared memory).
  - Workstation B accesses a storage device to get the data and/or programs.
  - Workstation B accesses a storage device to return the results from its assigned tasks.
- A distributed memory scenario:
  - Workstation A sends workstation B the programs and/or data for the allocated tasks, from its local memory.
  - Workstation A "fills the pipe" in the course of transmitting to workstation B the data for the allocated tasks, from its local memory.
  - Workstation B "fills the pipe" in the course of transmitting to workstation A the the results from its assigned tasks, from its local memory.

[1] We assume for the moment that task lengths are known perfectly. Later in the paper, we consider the effect of imperfect knowledge of task lengths on our results.

Of course, the costs of transmitting programs can be avoided in data parallel computations by prestoring a copy of the single program in all workstations on the network.

Typically, the overhead c is quite large compared to the time required to compute a task. Because of this, it is reasonable for the owner of workstation A to try to minimize the number of times that s/he must supply workstation B with work.

Finally, a note on modeling: the fact that c is fixed, independent of the size of the workload transmitted to workstation B, means that the time our model assesses to computing a task includes the marginal pipeline cost of transmitting that task to workstation B and the marginal pipeline cost of workstation B's returning whatever results it computes in the current episode of cycle-stealing.

Work schedules revisited. At time τ_k, the beginning of period k of schedule $\mathcal{S} = t_0, t_1, \ldots$, the owner of workstation A supplies workstation B (either directly or indirectly, as indicated above) with a job containing[2]

$$w_k \stackrel{\text{def}}{=} t_k \ominus c$$

units of work to perform during this period: If the owner of workstation B *has not returned* by time $T_k = \tau_k + t_k$, then the amount of work done so far during this episode is augmented by w_k; if the owner *has returned* by time T_k, then the episode terminates, with the total amount of work $w_0 + w_1 + \cdots + w_{k-1}$. This termination and work total reflect the fact that the owner of workstation A loses all work that is interrupted by the return of the owner of workstation B. This brief overview points out two facts that influence one's strategy when forming a schedule.

1. Because the fixed communication overhead c is incurred in each period of a schedule, a period of length t produces only $t \ominus c$ work.

2. In the *bounded lifespan* scenario with lifespan L, the risk of being interrupted, hence losing work, may make it desirable to have the lengths of the *productive* periods (i.e., those t_i that exceed c) sum to *less than* L.

We turn now to a discussion of how we formalize and quantify the risk of being interrupted in the midst of a period.

Risks. The *risk* in an episode of cycle-stealing is characterized by two closely related probability functions.

- q is the *risk function* of the episode: for each time t, $q(t)$ is the probability that the owner of workstation B *returns at precisely time* t. Since every episode must end, q

[2]The operator "$\ominus$" denotes *positive subtraction* and is defined by: $x \ominus y \stackrel{\text{def}}{=} \max(0, x - y)$.

satisfies the condition

$$\int_{x \geq 0} q(x)dx = 1.$$

- p is the nonincreasing *life function* of the episode: for each time t, $p(t)$ is the probability that the owner of workstation B *has not returned by time* t.

The functions p and q obey the boundary conditions

$$\begin{aligned} q(0) &= 0 \\ p(0) &= 1 \end{aligned}$$

and, in the *bounded lifespan* scenario with lifespan L,

$p(t) = 0$ for all $t \geq L$.

The functions p and q are related via the *marginal risk function* q^+, which is defined as follows.

$$q^+(t) \stackrel{\text{def}}{=} \int_{x=t\ominus 1}^{t} q(x)dx.$$

First of all, we can define p in terms of q as follows:

$$p(t) = 1 - \int_{x=0}^{t} q(x)dx = 1 - \sum_{i=0}^{t} q^+(i). \tag{2.1}$$

Then, we can define q^+ in terms of p as follows:

$$q^+(t) = p(t) - p(t-1). \tag{2.2}$$

The reader should note that we perform our study in a continuous rather than discrete environment to simplify certain manipulations. The reader should easily be able to extract discrete approximations to our results.

Expected work. Our goal throughout is to maximize the *expected work* in an episode of cycle-stealing. Given any schedule $\mathcal{S} = t_0, t_1, \ldots$ and life function p, this quantity is denoted $E(\mathcal{S}; p)$ and is given by

$$E(\mathcal{S}; p) = \sum_{i \geq 0} (t_i \ominus c) p(T_i) = \sum_{i \geq 0} w_i p(T_i). \tag{2.3}$$

The summation in equation (2.3) must account for every period in schedule $\mathcal{S}$. Accordingly, its upper limit is ∞ in the *unbounded lifespan* scenario and $m-1$ in an m-period *bounded lifespan* scenario.

A cycle-stealing schedule $\mathcal{S} = t_0, t_1, \ldots$ is *optimal* for life function p if $E(\mathcal{S}; p) \geq E(\mathcal{S}'; p)$ for any other schedule $\mathcal{S}'$, i.e., if schedule $\mathcal{S}$ maximizes the expected work.

A closely related study. There is a strong formal similarity between our single-episode version of the cycle-stealing problem and the problem of scheduling saves in a fault-prone computing system. The latter problem is studied in [2], wherein one finds some results that are quite close to some of our results, especially Theorem 4.1. Detailed differences in models and dramatic differences in methodology render the development here and in [2] kindred, but quite distinct.

2.2 Productive Schedules: A Simplifying Observation

We begin our study with a technical lemma that simplifies our search for optimal cycle-stealing schedules, by showing that such schedules cannot have many *nonproductive periods*, i.e., periods whose lengths do not exceed the communication overhead c. Specifically, in the *unbounded lifespan* scenario, an optimal schedule cannot have any nonproductive periods, and in the *bounded lifespan* scenario, only the last period of an optimal schedule can be nonproductive.

Call a cycle-stealing schedule $\mathcal{S}$ *productive* for an episode of cycle-stealing with communication overhead c, if the following holds. If $\mathcal{S}$ has infinitely many periods (in the unbounded lifespan scenario), then every period of $\mathcal{S}$ has length $> c$. If $\mathcal{S}$ has m periods (in the bounded lifespan scenario), then every period of $\mathcal{S}$, save possibly the last, has length $> c$.

Lemma 2.1 *For every episode of cycle-stealing, there is an optimal productive schedule.*

Proof. Note first that we lose no generality by assuming that all periods in a schedule have *positive* length. Let us focus, therefore, on a schedule $\mathcal{S} = t_0, t_1, \ldots, t_{k-1}, t_k, t_{k+1}, \ldots$ having a nonproductive period k; i.e., $0 < t_k \leq c$. Construct the schedule $\mathcal{S}^{<k>} = s_0, s_1, \ldots$ from $\mathcal{S}$ as follows. If $\mathcal{S}$ has infinitely many periods, then so also does $\mathcal{S}^{<k>}$; if $\mathcal{S}$ has m periods, then $\mathcal{S}^{<k>}$ has $m-1$ periods; in either case, the periods of $\mathcal{S}^{<k>}$ are defined as follows.

$$s_i = \begin{cases} t_i & \textit{for} \;\; i < k \\ t_k + t_{k+1} & \textit{for} \;\; i = k \\ t_{i+1} & \textit{for} \;\; i > k. \end{cases}$$

We claim that, for all life functions p, $E(\mathcal{S}^{<k>}; p) \geq E(\mathcal{S}; p)$. To wit, by direct calculation.

$$\begin{aligned} E(\mathcal{S}^{<k>}; p) - E(\mathcal{S}; p) &= (t_k + t_{k+1} \ominus c)p(T_k + t_{k+1}) - (t_k \ominus c)p(T_k) - (t_{k+1} \ominus c)p(T_k + t_{k+1}) \\ &= [(t_k + t_{k+1} \ominus c) - (t_{k+1} \ominus c)]p(T_k + t_{k+1}) \\ &\geq 0. \end{aligned}$$

In other words, if a positive-length nonproductive period appears as any but the last period of $\mathcal{S}$, one can never decrease the expected work of $\mathcal{S}$ by combining the nonproductive period with its successor. □

3 The Geometrically Decreasing Lifespan Model

We begin our study with the *geometrically decreasing lifespan* (*GDL*) model, wherein each episode of cycle-stealing has a "half-life;" i.e., the probability that an episode lasts at least $\ell + 1$ time units is roughly half the probability that it lasts at least ℓ time units. This model fits most naturally within the *unbounded lifespan* scenario. For the sake of generality and reality, we replace the parameter $1/2$ in "half-life" by $1/a$ for some *risk parameter* $a > 1$. This adds a bit of realism, in the sense that the "half-life" of an episode need not be measured in the same time units as is work. Note that, with any given risk factor, the conditional distribution of risk in this model looks the same at every moment of time. This fact enters implicitly into our analysis of the model.

Risk functions. Formally, the life function for the GDL model with risk parameter a is given by:

$p_a(t) = ap_a(t+1) = a^{-t}$ for all $t \geq 0$.

Our study of the GDL model focusses on the existence and structure of optimal schedules. In Section 3.1, we present a schedule $\mathcal{S}^{(a)}$ for the GDL model with risk parameter a that is *uniform*, in the sense of having equal-length periods. We prove in Theorem 3.1 that schedule $\mathcal{S}^{(a)}$ is the *unique* optimal schedule for the GDL model with risk parameter a. In Section 3.2, we prove a weak converse to Theorem 3.1, showing that any cycle-stealing episode for which a uniform schedule is optimal honors a weak analog of the GDL model (Proposition 3.1).

3.1 Optimal Schedules for the GDL Model

For each risk parameter $a > 1$, consider the uniform schedule $\mathcal{S}^{(a)} \stackrel{\text{def}}{=} t^{(a)}, t^{(a)}, t^{(a)}, \ldots$ whose periods have common length $t^{(a)}$ defined implicitly by the equation[3]

$$t^{(a)} \ln a + a^{-t^{(a)}} = 1 + c \ln a. \tag{3.1}$$

Direct calculation shows that $\mathcal{S}^{(a)}$ has expected work

$$E(\mathcal{S}^{(a)}; p_a) = \frac{a^{-t^{(a)}}}{\ln a}. \tag{3.2}$$

Theorem 3.1 *$\mathcal{S}^{(a)}$ is the unique optimal schedule for the GDL model with risk parameter a.*

[3] $\ln a$ denotes the natural logarithm of a.

Proof. We address three issues in turn. First, we prove that there indeed exists an optimal schedule for the GDL model with risk parameter a.[4] Next, we prove that the schedule $\mathcal{S}^{(a)}$ is one such optimal schedule. Finally, we prove the uniqueness of schedule $\mathcal{S}^{(a)}$.

There exist optimal schedules. The existence of schedules that are optimal for the GDL model with risk parameter a follows from the Least Upper Bound Principle. To wit, let $\mathcal{S} = t_0, t_1, t_2, \ldots$ be a schedule all of whose periods have length $> c$. (By Lemma 2.1, if there exists an optimal schedule, then there exists such a productive one.) By definition, then,

$$E(\mathcal{S}; p_a) = \sum_{i=0}^{\infty}(t_i - c)a^{-T_i} \leq \sum_{i=0}^{\infty}(t_i - c)a^{-(t_i+T_{i-1})} \leq \sum_{i=0}^{\infty}(t_i - c)a^{-(t_i+(i-1)c)}.$$

Since the form xa^{-x} is bounded above by a constant, it follows that $E(\mathcal{S}; p_a)$ is also bounded above by a constant, whence the claim.

Schedule $\mathcal{S}^{(a)}$ is optimal. Let $\mathcal{S} = t_0, t_1, t_2, \ldots$ be any optimal schedule, and consider the tail $\phi_1(\mathcal{S}) = t_1, t_2 \ldots$ of $\mathcal{S}$. Since schedule $\mathcal{S}$ is optimal, we have $E(\mathcal{S}; p_a) \geq E(\phi_1(\mathcal{S}); p_a)$, so that

$$E(\mathcal{S}; p_a) = (t_0 - c)a^{-t_0} + a^{-t_0}E(\phi_1(\mathcal{S}); p_a) \leq (t_0 - c)a^{-t_0} + a^{-t_0}E(\mathcal{S}; p_a). \quad (3.3)$$

This recurrence yields the bound.

$$E(\mathcal{S}; p_a) \leq \frac{t_0 - c}{a^{t_0} - 1}. \quad (3.4)$$

By direct calculation, one sees that, for all $t > c$, the uniform schedule $\mathcal{S}_t \stackrel{\text{def}}{=} t, t, t, \ldots$ has expected work

$$E(\mathcal{S}_t; p_a) = \frac{t - c}{a^t - 1}, \quad (3.5)$$

so that the expected work of schedule $\mathcal{S}_{t_0}$ matches the bound of inequality (3.4). Three conclusions follow.

1. The bound in (3.4) is, in fact, an equality.
2. The inequality in (3.3) is an equality. (This will be useful in our argument for uniqueness.)
3. There is a uniform schedule, call it $\mathcal{S}^{(a)}$, that is optimal for the GDL model with risk parameter a.

[4]This is not clear *a priori*; for instance, the life function $p(t) = 1/(t+1)$ does not admit an optimal schedule.

Having demonstrated the existence of schedule $\mathcal{S}^{(a)}$, we can identify it definitively by determining the value $t^{(a)}$ of t that maximizes expression (3.5). This determination is a straightforward calculus exercise: expression (3.5) has a unique maximum that occurs when t assumes the value $t^{(a)}$ defined implicitly in equation (3.1). The resulting expression (3.2) for $E(\mathcal{S}^{(a)}; p_a)$ is immediate.

Schedule $\mathcal{S}^{(a)}$ is uniquely optimal. We proceed by revisiting the optimal schedule $\mathcal{S} = t_0, t_1, \ldots$ inductively, period by period.

As we just noted, the expected work of schedule $\mathcal{S}$ is dually given by expression (3.4), via direct derivation, and by expression (3.2), via the preceding analysis. Since $t^{(a)}$ is the *unique* value of t that maximizes expression (3.5), it follows that $t_0 = t^{(a)}$.

If we combine the expression (3.2) for $E(\mathcal{S}; p_a)$ with the expression in (3.3) that relates $E(\phi_1(\mathcal{S}); p_a)$ and $E(\mathcal{S}; p_a)$, we find that $E(\phi_1(\mathcal{S}); p_a) = E(\mathcal{S}; p_a)$. Therefore, if we repeat the calculation that led to (3.3), but using schedule $\phi_1(\mathcal{S})$ in place of schedule $\mathcal{S}$, we discover that $t_1 = t^{(a)}$.

Continuing inductively through the successive tails of schedule $\mathcal{S}$, we find that each period of $\mathcal{S}$ has length $t^{(a)}$. We conclude that schedule $\mathcal{S}^{(a)}$ is the unique optimal schedule for the GDL problem with risk parameter a, which completes the proof. □

3.2 A Weak Converse

It is easier to describe the weak converse of Theorem 3.1 that we expose in this section if we reword the Theorem as follows.

If $p_a(t) = ap_a(t+1)$, then $w_i \equiv w^{(a)} \stackrel{\text{def}}{=} 1/(a-1)$.

Our weak converse of this fact takes the following informal form.

Any cycle-stealing episode for which a uniform schedule is optimal "almost" honors the GDL model.

To make this claim precise, say that the uniform schedule $\mathcal{S}$ is optimal for a given cycle-stealing episode, and say that all periods of $\mathcal{S}$ are identically $w + c$ for some parameter w. On the one hand, we have

$$E(\mathcal{S}) - E(\mathcal{S}^{+k}) \geq 0$$

so that

$$w \sum_{i=k}^{\infty} q(T_i) \geq p(T_k). \tag{3.6}$$

On the other hand, we have

$$p(T_k - 1) = 1 - \sum_{i=1}^{T_k - 1} q(i) = \sum_{i=T_k}^{\infty} q(i) \geq \sum_{i=k}^{\infty} q(T_i). \tag{3.7}$$

Combining Facts (3.6) and (3.7), we find the desired weak converse:

Proposition 3.1 *Say that the uniform schedule $\mathcal{S} = w + c, w + c, \ldots$ is optimal for a given cycle-stealing episode. Then for infinitely many t, including all $t \equiv 0 \bmod (w + c)$,*

$$p(t) \leq wp(t-1).$$

4 The Uniform Risk Model

In this section, we consider the *uniform risk* (*UR*) model, wherein the probability of the return of the owner of workstation B is the same at every moment. Of course, this assumption makes sense only in the *bounded lifespan* scenario. The intention is to model a situation wherein one can predict with some confidence that the owner of workstation B is likely to be absent for precisely L time units, but wherein there is a slowly growing probability that s/he will return early, so that the probability of retaining control of the workstation at any particular time decreases at a fixed constant rate.

Risk functions. The risk functions for the UR model with lifespan L are given explicitly by:

- $q_L(t) = 1/L$ for $1 \leq t \leq L$.
- $p_L(t) = 1 - t/L$ for $0 \leq t \leq L$.

Characterizing optimal schedules. We now demonstrate that the unique optimal schedule for the UR model partitions the anticipated lifespan into periods whose lengths form a decreasing arithmetic sequence whose common difference is the communication overhead c; moreover, these periods are maximal in number, given this rate of decrease.

For each lifespan L, consider the m-period schedule $\mathcal{S}^{(L)} = t_0^{(L)}, t_1^{(L)}, \ldots, t_{m-1}^{(L)}$, where

$$m = \left\lfloor \sqrt{\frac{2L}{c} + \frac{1}{4}} - \frac{1}{2} \right\rfloor, \tag{4.1}$$

and, for $0 \le i < m$,

$$t_i^{(L)} = \frac{L}{m+1} + \frac{cm}{2} - ci. \tag{4.2}$$

Direct calculation shows that $\mathcal{S}^{(L)}$ has expected work

$$E(\mathcal{S}^{(L)}; p_L) = \frac{L}{2} - mc - \frac{L}{2(m+1)} + \frac{m(m+1)(m+2)c^2}{24L}. \tag{4.3}$$

Theorem 4.1 *$\mathcal{S}^{(L)}$ is the unique optimal schedule for the UR model with lifespan L.*

Proof. Let us represent by $\mathcal{S} = t_0, t_1, \ldots, t_{m-1}$ a generic m-period schedule for the UR model with lifespan L. Note that the number of periods m is an unknown here, as well as the period lengths $t_0, t_1, \ldots, t_{m-1}$. When equation (2.3) is instantiated with schedule $\mathcal{S}$ and risk function p_L, it can be written in the form

$$E(\mathcal{S}; p_L) = \sum_{i=0}^{m-1} (t_i \ominus c) \left(1 - \frac{1}{L} T_i\right). \tag{4.4}$$

We uncover the optimal cycle-stealing schedule for the UR model with lifespan L by deriving an assignment of values to the variables $m, t_0, t_1, \ldots, t_{m-1}$, that maximizes expression (4.4) subject to the constraints:

- $c > 0$, $L > 0$, and $m > 0$;
- $t_i \ge 0$ for all $0 \le i < m$;
- $\sum_{i=0}^{m-1} t_i = L$.

Two simplifications facilitate our search for a maximizing assignment. Firstly, Lemma 2.1 assures us that if we substitute ordinary subtraction for positive subtraction in all terms of expression (4.4) save the last, we do not change the maximum value of the expression within the space of interest. We lose no generality, therefore, by focussing on the expression

$$\sum_{i=0}^{m-2} (t_i - c) \left(1 - \frac{1}{L} T_i\right) + (t_{m-1} \ominus c) \left(1 - \frac{1}{L} T_{m-1}\right) \tag{4.5}$$

rather than on expression (4.4). Secondly—and less obviously—we shall see that we lose no generality if we use ordinary subtraction rather than positive subtraction in the last term of expression (4.5) also, yielding the expression

$$\sum_{i=0}^{m-1} (t_i - c) \left(1 - \frac{1}{L} T_i\right). \tag{4.6}$$

Although this change broadens the space over which we are searching for a maximizing assignment, we shall see that the unique maximizing assignment for expression (4.6) over the broader space actually lies in the space of sought maximizing assignments for expression (4.5), hence also for expression (4.4). The ability to use ordinary subtraction throughout significantly simplifies our analysis.

We begin our search for a maximizing assignment by rewriting expression (4.6) to better expose its fundamental structure. By direct manipulation plus the fact that $L^2 = (t_0 + t_1 + \cdots + t_{m-1})^2$, we find that

$$\begin{aligned} L \cdot \sum_{i=0}^{m-1} (t_i - c)(1 - T_i/L) &= \sum_{i=0}^{m-1} (t_i - c)(t_{i+1} + t_{i+2} + \cdots + t_{m-1}) \\ &= \frac{1}{2}\left(L^2 - \sum_{i=0}^{m-1} t_i^2\right) - c\sum_{i=0}^{m-1} it_i \\ &= \frac{1}{2}L^2 - \frac{1}{2}\sum_{i=0}^{m-1}(t_i^2 + 2cit_i) \\ &= \frac{1}{2}L^2 - \frac{1}{2}\sum_{i=0}^{m-1}\left[(t_i + ci)^2 - c^2 i^2\right], \end{aligned}$$

the last expression resulting from completing the square. Finally, we let $u_i = t_i + ci$ for $0 \le i < m$, and we divide by L (to compensate for having factored out L earlier), to arrive at the expression for $E(\mathcal{S}^{(L)})$ that we shall try to maximize.

$$E(\mathcal{S}^{(L)}) = \frac{1}{2}L + \frac{c^2}{2L}\sum_{i=0}^{m-1} i^2 - \frac{1}{2L}\sum_{i=0}^{m-1} u_i^2 = \frac{1}{2}L + \frac{c^2 m(m-1)(2m-1)}{12L} - \frac{1}{2L}\sum_{i=0}^{m-1} u_i^2. \tag{4.7}$$

Our goal now is to maximize expression (4.7) subject to the following constraints.

1. Since each $t_i \ge 0$, we must have each $u_i \ge ci$.
2. Since $\sum_{i=0}^{m-1} t_i = L$, we must have

$$\sum_{i=0}^{m-1} u_i = \sum_{i=0}^{m-1} t_i + ci = L + c\frac{m(m-1)}{2}.$$

Easily, we shall have achieved this goal once we have minimized the sum

$$u_0^2 + u_1^2 + \cdots + u_{m-1}^2 \tag{4.8}$$

subject to these same constraints. Now, were it not for constraint (1), we could minimize the sum (4.8) simply by setting each u_i to its average value

$$u_i = \frac{L}{m} + \frac{c(m-1)}{2}.$$

Constraint (1) forces us to be a bit more careful. Specifically, we can use this simple minimize-by-averaging technique only for the first $r+1$ terms of sum (4.8), where r is the largest integer such that

$$\frac{L}{r+1} + \frac{cr}{2} \geq cr. \tag{4.9}$$

A short calculation is needed to verify that the lefthand side of inequality (4.9) is, indeed, the average of the first $r+1$ terms of (4.8). The naive expression for this average is

$$A_r \stackrel{\text{def}}{=} \frac{L + cm(m-1)/2 - u_{r+1} - u_{r+2} - \cdots - u_{m-1}}{r+1}.$$

Because of constraint (1), we can simplify this expression as follows.

$$\begin{aligned} (r+1)A_r &= L + cm(m-1)/2 - \sum_{i=r+1}^{m-1} ci \\ &= L + cr(r+1)/2 \end{aligned}$$

For the remainder of the sum (4.8)—the portion that cannot be minimized using averaging—we use the value forced on us by constraint (1), namely $u_i = ci$. Branching on the value of r, we end up with the following minimizing assignment for the u_i.

$$u_i = \begin{cases} L/(r+1) + cr/2 & \text{for } i \leq r \\ ci & \text{for } i > r \end{cases} \tag{4.10}$$

It remains only to determine the maximizing value of r. This is done easily by rewriting inequality (4.9) in the form $r^2 + 2r - 2L/c \leq 0$. It is now clear that the maximizing value of r is given by

$$r = \left\lfloor \sqrt{\frac{2L}{c} + \frac{1}{4}} - \frac{1}{2} \right\rfloor. \tag{4.11}$$

We can now, finally, turn back to our original setting and convert the minimizing assignment (4.10) to the variables $\{u_i\}$ to the sought maximizing assignment to the variables $\{t_i\}$:

$$t_i = \begin{cases} L/(r+1) + cr/2 - ci & \text{for } i \leq r \\ 0 & \text{for } i > r \end{cases} \tag{4.12}$$

One sees from assignment (4.12) that all periods beyond the first r have zero length, hence contribute no work. It follows that the maximizing value of r in equation (4.11) is in fact the sought number of periods in the optimal schedule—the desired value for the variable m; this verifies equation (4.1). Assignment (4.12) also yields directly the sought values for the period lengths $\{t_i\}$, thus verifying equation (4.2).

In order to complete the proof, we need only verify that our maximizing assignments for m and $\{t_i\}$ yield equation (4.3) as the expected amount of work performed by the optimal schedule. We accomplish this by direct evaluation of expression (4.7) with the maximizing values of all parameters.

$$\begin{aligned} E(\mathcal{S}^{(L)}; p_L) &= \frac{L}{2} + \frac{c^2 m(m-1)(2m-1)}{12L} - \frac{1}{2L} \sum_{i=0}^{m-1} u_i^2 \\ &= \frac{L}{2} + \frac{c^2 m(m-1)(2m-1)}{12L} - \frac{1}{2L} \left[\sum_{i=0}^{r} \left(\frac{L}{r+1} + \frac{cr}{2} \right) - \sum_{i=r+1}^{m-1} (ci)^2 \right] \\ &= \frac{L}{2} + \frac{c^2 r(r-1)(2r-1)}{12L} - \frac{r+1}{2L} \left(\frac{L}{r+1} + \frac{cr}{2} \right). \end{aligned}$$

Equation (4.3) now follows by direct calculation. □

For the sake of perspective, we note that $E(\mathcal{S}^{(L)}; p_L)$ is very close to

$$E(\mathcal{S}^{(L)}; p_L) = \frac{L}{2} - \frac{7}{6}\sqrt{2cL}$$

when L is very much larger than c (which is likely to be the case).

An obvious way to approximate the optimal schedule using a simpler control mechanism is to consider the schedule $\mathcal{S}'$ that has the right "order" of number of periods, namely, $\sqrt{L}$, but that controls the periods "obliviously," by having them all of equal length, namely, $\sqrt{L}$. By direct calculation using expression (4.6), one sees that there is a significant penalty for such simplification, especially if c is large:

$$E(\mathcal{S}') = \frac{L}{2} + \frac{\sqrt{L}}{2} - \frac{c}{2}(\sqrt{L} + 1).$$

5 The Geometrically Increasing Risk Model

In this section, we consider the *geometrically increasing risk* (*GIR*) model, in which the probability of the return of the owner of workstation B doubles at each time unit. This draconian model, which makes sense only in the *bounded lifespan* scenario, may be appropriate when B's owner is likely to be absent for only a short period of time, say because of a telephone call. In contrast to Section 3, we interpret the word "double" literally here, rather than replacing the parameter 2 by an arbitrary risk parameter $a > 1$, in order to retain the arithmetic simplicity of a singly parameterized model.

Risk functions. The risk functions for the GIR model with lifespan L are given explicitly by:

- $q_L^g(t+1) \stackrel{\text{def}}{=} 2q_L^g(t) = (2^L - 1)2^{-(t-1)}$ for $t \geq 1$.
- $p_L^g(t) \stackrel{\text{def}}{=} (2^L - 2^t)/(2^L - 1)$ for $t \geq 0$.

Characterizing optimal schedules. The optimal schedule for the GIR model with lifespan L partitions the lifespan into periods of exponentially decreasing lengths; i.e., the kth period is exponentially longer than the $(k+1)$th period.

For each lifespan L, consider the m-period schedule $\widehat{\mathcal{S}}^{(L)} = \hat{t}_0^{(L)}, \hat{t}_1^{(L)}, \ldots, \hat{t}_{m-1}^{(L)}$, where, to within rounding,[5]

$$m = \log^\star L - \log^\star c, \tag{5.1}$$

and where the lengths $\hat{t}_k^{(L)}$ are given inductively as follows:

$$\hat{t}_k^{(L)} \stackrel{\text{def}}{=} 2^{\hat{t}_{k+1}^{(L)}} + c - 2 \quad \text{for } k \in \{0, 1, \ldots, m-2\} \tag{5.2}$$

$$\hat{t}_{m-1}^{(L)} \stackrel{\text{def}}{=} L - \left(\hat{t}_0^{(L)} + \hat{t}_1^{(L)} + \cdots + \hat{t}_{m-2}^{(L)}\right). \tag{5.3}$$

Theorem 5.1 $\widehat{\mathcal{S}}^{(L)}$ *is the optimal schedule for the GIR model with lifespan L.*

Proof. Let $\mathcal{S} = t_0, t_1, \ldots, t_{m-1}$ be an arbitrary m-period cycle-stealing schedule that is optimal for the GIR problem with lifespan L. If we instantiate equation (2.3) for the risk function p_L^g, we find, after some manipulation, that

$$E(\mathcal{S}; p_L^g) = \frac{1}{2^L - 1}\left(2^L(L-1) - mc2^L - \sum_{i=0}^{m-1} w_i 2^{T_i - 1}\right).$$

We use a perturbation argument to analyze schedule $\mathcal{S}$. The *kth-period positive and negative perturbations* of $\mathcal{S}$, respectively denoted $\mathcal{S}^{+k}$ and $\mathcal{S}^{-k}$, are the schedules

$$\mathcal{S}^{\pm k} \stackrel{\text{def}}{=} t_0, t_1, \ldots, t_{k-1}, t_k \pm 1, t_{k+1} \mp 1, t_{k+2}, \ldots, t_{m-1}.$$

Note that these perturbed schedules both have the same lifespan L as does $\mathcal{S}$.

The difference $E(\mathcal{S}; p_L^g) - E(\mathcal{S}^{+k}; p_L^g)$, which must be nonnegative because schedule $\mathcal{S}$ is optimal, is easily calculated as follows.

$$E(\mathcal{S}; p_L^g) - E(\mathcal{S}^{+k}; p_L^g) = w_k q_L^g(T_k) - p_L^g(T_k) + p_L^g(T_{k+1} - 1) = \frac{2^{T_i - 1}}{2^L - 1}\left(w_k - 2^{t_{k+1}} + 2\right) \geq 0.$$

[5]Throughout this section, all logarithms are to the base 2. If we inductively let $\log^{(i+1)} x = \log(\log^{(i)} x)$, then $\log^\star x$ denotes the smallest integer r for which $\log^{(r)} x \leq 1$.

The nonnegativity of this difference implies that

$$t_k \geq 2^{t_{k+1}} + c - 2. \tag{5.4}$$

Symmetrically, if we consider the difference $E(\mathcal{S}; p_L^g) - E(\mathcal{S}^{-k}; p_L^g)$, which must also be nonnegative, we find the following.

$$E(\mathcal{S}; p_L^g) - E(\mathcal{S}^{-k}; p_L^g) = p_L^g(T_k - 2) - w_k q_L^g(T_k - 1) - p_L^g(T_{k+1} - 1) = \frac{2^{T_i - 1}}{2^L - 1}\left(2^{t_{k+1}} - 2 - w_k\right) \geq 0.$$

The nonnegativity of this difference implies that

$$t_k \leq 2^{t_{k+1}} + c - 2. \tag{5.5}$$

Inequalities (5.4) and (5.5) combine to verify that the period-lengths of the optimal schedule $\mathcal{S}$ satisfy the system of equations (5.2) that defines schedule $\widehat{\mathcal{S}}^{(L)}$.

To verify equation (5.1), we recall from Lemma 2.1 that, without loss of generality, we may assume that each period of the optimal schedule $\widehat{\mathcal{S}}^{(L)}$ must have length exceeding c, so that it will yield some productive work. It follows that we can continue to take the logarithm of

$$\hat{w}_0^{(L)} \stackrel{\text{def}}{=} \hat{t}_0^{(L)} - c,$$

then of

$$\hat{w}_1^{(L)} \stackrel{\text{def}}{=} \hat{t}_1^{(L)} - c = \log(\hat{w}_0^{(L)} + 2),$$

then of

$$\hat{w}_2^{(L)} \stackrel{\text{def}}{=} \hat{t}_2^{(L)} - c = \log(\hat{w}_1^{(L)} + 2) - c = \log(\hat{t}_1^{(L)} - c + 2) - c = \log(\log(\hat{w}_0^{(L)} + 2) - c + 2) - c,$$

and so on, only so long as we attain periods of length $t > 2^c + c - 2$, at which point any additional periods would not be productive. This reasoning verifies equation (5.1), whence the theorem. □

6 Operating with an Unknown Life Function

The model we have studied thus far focusses on just a single episode of cycle stealing, and it assumes complete knowledge of the life function of the episode. In future work, we intend to relax these assumptions, both by allowing one to have just approximate knowledge of the life function and by trying to maximize the amount of work one gets done in multi-episode cycle stealing—when the owner of workstation B comes and goes several times. In this section, we present a first attempt at generalizing our model to a situation wherein very little is known about the life function. (Of course, if nothing at all is known about the life function, then there is rather little that we can do with confidence since our first job can be killed just before it is finished, no matter how long the job is.)

6.1 A New Model: Competing against Omniscience

We now consider a multi-episode model wherein we assume no foreknowledge of the life functions of the cycle-stealing episodes. We assume, instead, that we have two other pieces of knowledge.

1. We know the *useful lifespan* U of the cycle-stealing opportunity: we know that the owner of workstation B will be absent for U time units, over a possibly longer period that is punctuated by possibly many interrupts caused by his/her return.

2. We know that the optimal, omniscient schedule, call it $\mathcal{S}^\star$, is able to accomplish at least $W^\star \stackrel{\text{def}}{=} \alpha U$ units of work during the opportunity, for some $\alpha < 1$.

We stress that we do *not* assume knowledge of the locations of the interrupts, nor, in the most general version of this model, of their number. Our goal in this model is to design a schedule that is guaranteed to accomplish βU units of work, independent of the number and location(s) of the interrupts, where β is as close to α as possible. In a typical application, the total lifespan might comprise a night, a 24-hour day, or a week. If we assume that this is a period of time in which the owner of workstation B is typically away from work, then we can expect α to be close to 1. In order to make our model conservative, yet challenging, let us assume that $\alpha > 1/3$.

This model differs from that of the previous sections in three important respects. First, we now allow multiple episodes within the total lifespan, rather than only one. Second, we now assume nothing about the life function of any particular episode, assuming instead knowledge of the useful lifespan and of how much work one could accomplish if one knew in advance where the interrupts would occur. Third, we now measure the performance of a candidate schedule by comparing its guaranteed work output to that of the ideal schedule $\mathcal{S}^\star$.

6.2 Optimal Deterministic Oblivious Multi-Episode Schedules

The strategy we study first produces schedules for arbitrary multi-episode cycle-stealing opportunities, that are *deterministic*, in the sense of involving no randomization, and *oblivious*, in the sense of allocating work with no regard to where prior interrupts (if any) have occurred. These schedules will always start the ith job as soon as the $(i-1)$th job has finished and/or as soon as workstation B becomes available following an interrupt during the $(i-1)$th job. (Similar results will hold for the model in which the $(i-1)$th job is restarted if it is interrupted before the ith job is run.) A rather simple scheduling strategy can be shown to be optimal for this scenario.

The optimal schedule $\mathcal{S}^{(\alpha)}$. For all $\alpha < 1$, let $\mathcal{S}^{(\alpha)}$ be the uniform[6] multi-episode cycle-stealing schedule whose periods have common length $t^{(\alpha)} \stackrel{\rm def}{=} c/\sqrt{1-\alpha}$.

Theorem 6.1 *For all* $\alpha < 1$, *schedule* $\mathcal{S}^{(\alpha)}$ *accomplishes at least*

$$W(\mathcal{S}^{(\alpha)}) \geq (1-\sqrt{1-\alpha})^2 U \tag{6.6}$$

units of work, which is at least the fraction

$$\frac{W(\mathcal{S}^{(\alpha)})}{W^\star} \geq \frac{1-\sqrt{1-\alpha}}{1+\sqrt{1-\alpha}} \tag{6.7}$$

of the work achieved by the omniscient schedule $\mathcal{S}^\star$. *Moreover, schedule* $\mathcal{S}^{(\alpha)}$ *is uniquely optimal among deterministic oblivious schedules, in guaranteed work output.*

Proof. We begin by analyzing the work output of schedule $\mathcal{S}^{(\alpha)}$; then we prove its optimality. Throughout, let $\alpha < 1$ be fixed but arbitrary.

An analysis of schedule $\mathcal{S}^{(\alpha)}$. Assume that we are temporarily omniscient, and we see that the useful lifespan U will be partitioned by interrupts into k episodes of lifespans $L_0, L_1, \ldots, L_{k-1}$, respectively. Without loss of generality, we may assume that each $L_i > c$, for even schedule $\mathcal{S}^\star$ will get no work done during a shorter episode. Clearly, based on its perfect overview, schedule $\mathcal{S}^\star$ will accomplish $L_i - c$ units of work during each lifespan-L_i episode i. Since $\mathcal{S}^\star$ accomplishes at least $W^\star = \alpha U$ units of work in all, we know that

$$\sum_{i=0}^{k-1}(L_i - c) \geq \alpha U.$$

Importantly, we know also that $\alpha U + ck \leq U$, so that the number of interrupts (being an integer, and being 1 less than the number of episodes) satisfies

$$\text{Number-of-Episodes} = k-1 < \left\lfloor \frac{(1-\alpha)U}{c} \right\rfloor, \tag{6.8}$$

and so that the "average" episode has duration

$$\text{Average-Duration} = \frac{U}{k} \geq \frac{c}{1-\alpha}. \tag{6.9}$$

Consider now an arbitrary uniform scheduling strategy $\mathcal{S}_t$ that allocates t time units to every period of every episode. It makes sense to impose two constraints on the values of the parameter t; we assume:

[6]By extending our earlier terminology, we call a multi-episode cycle-stealing schedule *uniform* if all of its periods have the same length.

1. that $t > c$, so that we have a chance of accomplishing some work during each period (as long as the period is not interrupted);

2. that $t \leq c/(1-\alpha)$, so that we are guaranteed to encounter some productive episodes; cf. inequality (6.9).

Now, during each (uninterrupted) episode of lifespan L_i, schedule $\mathcal{S}_t$ clearly accomplishes $W_i^{(t)} \stackrel{\text{def}}{=} \lfloor L_i/t \rfloor (t-c)$ units of work. Invoking the bound (6.8), the total amount of work achieved by schedule $\mathcal{S}_t$ satisfies:

$$
\begin{aligned}
W(\mathcal{S}_t) = \sum_{i=0}^{k-1} W_i^{(t)} &= \sum_{i=0}^{k-1} \lfloor L_i/t \rfloor (t-c) \\
&\geq (t-c) \sum_{i=0}^{k-1} (L_i/t - 1) \\
&\geq (t-c)[\alpha U - (t-c)k]/t \\
&\geq (t-c)[\alpha U - (t-c)(1-\alpha)U/c]/t = (c + (\alpha - 1)t)(1/c - 1/t)U.
\end{aligned}
$$

Easily, the last expression is maximized by setting $t = t^{(\alpha)}$. Direct calculation verifies that the amount of work accomplished by the resulting uniform schedule, $\mathcal{S}^{(\alpha)} \stackrel{\text{def}}{=} \mathcal{S}_{t^{(\alpha)}}$, is bounded above by expression (6.6). Comparing this amount of work with the amount of work $W^\star$ accomplished by the omniscient schedule $\mathcal{S}^\star$, we find that schedule $\mathcal{S}^{(\alpha)}$ achieves the "competitive ratio" given in expression (6.7). Thus, when α is close to 1, the work accomplished under schedule $\mathcal{S}^{(\alpha)}$ is close to U, and the competitive ratio $W^{(\alpha)}/W^\star$ is close to 1. For instance, when α exceeds 3/4, the competitive ratio is greater than 1/3.

The optimality of schedule $\mathcal{S}^{(\alpha)}$. We show now that schedule $\mathcal{S}^{(\alpha)}$ is optimal among deterministic, oblivious schedules for a useful lifespan U during which the omniscient schedule $\mathcal{S}^\star$ can accomplish at least α U units of work. To this end, let us consider an arbitrary deterministic, oblivious schedule[7] $\mathcal{S} = t_0, t_1, \ldots, t_n$ satisfying the following.

1. Each $t_i > c$, so that some work will be accomplished during each uninterrupted period.

2. $t_0 + t_1 + \cdots + t_n \leq U$, so that schedule $\mathcal{S}$ is appropriate for an opportunity having useful lifespan U.

Now, the amount of work accomplished by schedule $\mathcal{S}$ is given by

$$\sum_{i \in S} (t_i - c),$$

[7] Note that it is precisely the determinism and obliviousness of schedule $\mathcal{S}$ that allows us to specify its periods without knowing where the interrupts (if any) will come.

where the summation ranges over the set $S \subseteq \{0, 1, \ldots, n\}$ of indices of uninterrupted periods. By inequality (6.8), the number of interruptions cannot exceed $\lfloor(1-\alpha)U/c\rfloor - 1$ if the omniscient schedule is to accomplish at least αU units of work. It follows that, if the interrupts were to be specified by an adversary whose job was to minimize the work-output under schedule $\mathcal{S}$, the adversary's optimal strategy would be to interrupt the $\lfloor(1-\alpha)U/c\rfloor - 1$ longest periods of schedule $\mathcal{S}$. Clearly, then, in order to maximize the amount of work that schedule $\mathcal{S}$ *is guaranteed* to accomplish, no matter how cleverly the adversary chooses the number and location of the interrupts, the designer of schedule $\mathcal{S}$ will make the schedule uniform. The claimed optimality of schedule $\mathcal{S}^{(\alpha)}$ is now clear, for it provably maximizes the guaranteed work-output among uniform deterministic, oblivious schedules. □

6.3 Optimal Adaptive One-Interrupt (Two-Episode) Schedules

Devising optimal *adaptive* cycle-stealing schedules—whose periods' lengths are chosen based on the lengths of prior periods and the pattern of prior interrupts—is far more difficult than devising optimal deterministic oblivious ones. While we have not yet met this design challenge for arbitrary multi-episode cycle-stealing opportunities, we do know how to design optimal adaptive schedules for the restricted scenario in which we are guaranteed that there is at most one interrupt, hence, no more than two episodes. The complexity of this simplified problem hints at the difficulty of the general multi-episode problem.

Remarks. **(a)** The optimal omniscient schedule for this scenario, call it $\mathcal{S}_1^\star$, can accomplish at least $W_1^\star \stackrel{\text{def}}{=} U - 2c$ units of work, since there are at most two episodes. **(b)** The naive two-period schedule $\mathcal{S} = U/2, U/2$ accomplishes at least $U/2 - c$ units of work in this scenario, for a competitive ratio of $1/2$.

The optimal schedule $\mathcal{S}^{(\alpha,1)}$. We shall show that the following adaptive schedule for the single-interrupt scenario, call it $\mathcal{S}^{(\alpha,1)}$, is optimal for the scenario, in terms of guaranteed work-output. $\mathcal{S}^{(\alpha,1)}$ operates as follows. We prespecify a sequence of periods $t_0^{(\alpha,1)}, t_1^{(\alpha,1)}, \ldots, t_{\ell-1}^{(\alpha,1)}$ that sum to the useful lifespan U. Before the interrupt occurs (if it ever does), $\mathcal{S}^{(\alpha,1)}$ allocates, in turn, a period of length $t_0^{(\alpha,1)}$, then a period of length $t_1^{(\alpha,1)}$, then a period of length $t_2^{(\alpha,1)}$, and so on, terminating after the ℓth period if there is no interrupt. If there is an interrupt, then upon returning from it, $\mathcal{S}^{(\alpha,1)}$ allocates all of the remaining time as a single period.

In order to specify $\mathcal{S}^{(\alpha,1)}$, we first choose the number of periods, call it $\ell^\star$, to be the integer that maximizes the expression

$$\frac{\ell^\star - 1}{\ell^\star}(U - c) - \frac{c}{2}(\ell^\star - 3).$$

Note that $\ell^\star$ lies in the range

$$\binom{\ell^\star}{2} \le \frac{U}{c} - 1 \le \binom{\ell^\star + 1}{2}$$

and, in fact, that $\ell^\star = \sqrt{2U/c - 2} + \text{l.o.t.}$ Next, we specify the periods $t_i^{(\alpha,1)}$: for each $j \in \{0, 1, \ldots, \ell^\star - 2\}$, we define the jth *pre-interrupt* period $t_j^{(\alpha,1)}$ of $\mathcal{S}^{(\alpha,1)}$ to be

$$t_j^{(\alpha,1)} \stackrel{\text{def}}{=} \frac{1}{\ell^\star}(U - c) + \frac{c}{2}(\ell^\star - 1) - jc \; = \; t_0^{(\alpha,1)} - jc; \tag{6.10}$$

and we define $t_{\ell^\star-1}^{(\alpha,1)} \stackrel{\text{def}}{=} t_{\ell^\star-2}^{(\alpha,1)}$. By direct calculation,

$$t_0^{(\alpha,1)} + t_1^{(\alpha,1)} + \cdots + t_{\ell^\star-1}^{(\alpha,1)} = U. \tag{6.11}$$

Theorem 6.2 *For all $\alpha < 1$, schedule $\mathcal{S}^{(\alpha,1)}$ accomplishes at least*

$$W(\mathcal{S}^{(\alpha,1)}) \ge U - t_0^{(\alpha,1)} = \frac{\ell^\star - 1}{\ell^\star}(U - c) - \frac{c}{2}(\ell^\star - 3) \tag{6.12}$$

units of work, which is at least the fraction

$$\frac{W(\mathcal{S}^{(\alpha,1)})}{W_1^\star} \ge \left(1 - \sqrt{\frac{c}{2U}}\right) + l.o.t. \tag{6.13}$$

of the work accomplished by the omniscient single-interrupt schedule $\mathcal{S}_1^\star$. Moreover, $\mathcal{S}^{(\alpha,1)}$ is uniquely optimal among adaptive schedules for the single-interrupt scenario, in terms of guaranteed work output.

Proof. We begin by analyzing the work output of schedule $\mathcal{S}^{(\alpha,1)}$; then we establish its optimality. Throughout, let $\alpha < 1$ be fixed but arbitrary.

An analysis of schedule $\mathcal{S}^{(\alpha,1)}$. We begin our analysis by considering the work output of an arbitrary ℓ-period schedule $\mathcal{S} = t_0, t_1, \ldots, t_{\ell-1}$ which, in common with schedule $\mathcal{S}^{(\alpha,1)}$:

- exhausts the useful lifespan, in the sense that

$$t_0 + t_1 + \cdots + t_{\ell-1} = U; \tag{6.14}$$

- operates by using the periods t_i in turn before an interrupt and by using a single final comprehensive period after returning from an interrupt.

(Analyzing such a general $\mathcal{S}$ somewhat simplifies the analysis of $\mathcal{S}^{(\alpha,1)}$ and is useful when we address its optimality.)

Case 1: no interrupt. Assume first that no interrupt occurs. By equation (6.14), then, schedule $\mathcal{S}$ accomplishes

$$W(\mathcal{S}) = \sum_{i=0}^{\ell-1}(t_i - c) = U - \ell c$$

units of work.

Case 2: a single interrupt. Assume next that an interrupt does occur, say during the $(i+1)$th period (of length t_i), where $i \in \{0, 1, \ldots, \ell-1\}$. Then,

- *before the interrupt*, schedule $\mathcal{S}$ accomplishes $(t_0 - c) + (t_1 - c) + \cdots + (t_{i-1} - c)$ units of work;
- *after the interrupt*, $\mathcal{S}$ accomplishes

  $$U - (t_0 + t_1 + \cdots + t_i) \ominus c \geq U - (t_0 + t_1 + \cdots + t_i) - c$$

  units of work.

Thus, in this case, schedule $\mathcal{S}$ accomplishes

$$W(\mathcal{S}) = U - t_i - ic \ominus c \tag{6.15}$$

units of work.

Specializing the preceding analysis to $\mathcal{S}^{(\alpha,1)}$, by instantiating the values of the period lengths $t_j^{(\alpha,1)}$, we find that, in any single-interrupt cycle-stealing opportunity, the work output of $\mathcal{S}^{(\alpha,1)}$ is no smaller than expression (6.12), irrespective of whether or where the interrupt occurs. Direct calculation now verifies that schedule $\mathcal{S}^{(\alpha,1)}$ achieves the competitive ratio of expression (6.13).

The optimality of schedule $\mathcal{S}^{(\alpha,1)}$. Let us consider again the general ℓ-period schedule $\mathcal{S} = t_0, t_1, \ldots, t_{\ell-1}$ from the beginning of the proof. Let us assume that schedule $\mathcal{S}$ is optimal in work output among adaptive schedules for the single-interrupt scenario, and let us successively deduce properties of $\mathcal{S}$'s structure that will ultimately establish that $\mathcal{S} = \mathcal{S}^{(\alpha,1)}$.

The progression of $\mathcal{S}$'s period lengths. An adversary wishing to minimize the amount of work that schedule $\mathcal{S}$ accomplishes would clearly interrupt the schedule at the end of one of its periods, so as to remove that entire period from potential productivity for $\mathcal{S}$. But, which period should the adversary interrupt? If we review the reasoning that leads to expression (6.15), then we find the following.

- If the adversary interrupts schedule $\mathcal{S}$ during period $i \in \{0, 1, \ldots, \ell - 2\}$, which has duration t_i, then $\mathcal{S}$ accomplishes $W(\mathcal{S}) = U - t_i - (i+1)c$ units of work.
- If the adversary interrupts schedule $\mathcal{S}$ during period $\ell - 1$, which has duration $t_{\ell-1}$, then $\mathcal{S}$ accomplishes $W(\mathcal{S}) = U - t_{\ell-1} - (\ell-1)c$ units of work.

The adversary will clearly choose to interrupt the period that minimizes the work output of $\mathcal{S}$. The designer of schedule $\mathcal{S}$ clearly combats this strategy optimally by making all of the following quantities equal:

$$t_0 + c = t_1 + 2c = t_2 + 3c = \cdots = t_{\ell-3} + (\ell-2)c = t_{\ell-2} + (\ell-1)c = t_{\ell-1} + (\ell-1)c.$$

This is equivalent to setting period lengths so that

$$\begin{array}{rcl} t_j & = & t_0 - jc \quad \text{for } j \in \{0, 1, \ldots, \ell-2\} \\ t_{\ell-1} & = & t_{\ell-2} \end{array} \tag{6.16}$$

Note that schedule $\mathcal{S}^{(\alpha,1)}$ obeys this regimen of arithmetically decreasing period lengths.

The initial period length of $\mathcal{S}$. Next, note that the arithmetic-progression structure of schedule $\mathcal{S}$ exposed in (6.16) combines with the fact that the period lengths exhaust the useful lifespan U (expression (6.14)) to yield

$$U = t_0 + t_1 + \cdots + t_{\ell-1} = \ell t_0 - \binom{\ell-1}{2} c - (\ell-2)c,$$

so that

$$t_0 = \frac{1}{\ell}(U - c) + \frac{c}{2}(\ell - 1). \tag{6.17}$$

This is precisely the functional form of $t_0^{(\alpha,1)}$, with $\ell^\star$ substituted for ℓ; see expression (6.10).

The number of periods in $\mathcal{S}$. Finally, we determine the number ℓ of periods in schedule $\mathcal{S}$, after which we shall have a complete description of the schedule. To this end, we note that, by (6.16) and (6.17), the amount of work that $\mathcal{S}$ is guaranteed to accomplish is given by

$$W(\mathcal{S}) = U - t_0 = \frac{\ell-1}{\ell}(U - c) - \frac{c}{2}(\ell - 3)$$

(cf. (6.12)). Now, since schedule $\mathcal{S}$ is optimal, its number of periods ℓ must maximize this guarantee. Recall, however, that the number of periods $\ell^\star$ of schedule $\mathcal{S}^{(\alpha,1)}$ was chosen precisely to maximize this expression. We infer that $\ell = \ell^\star$.

Our three-stage analysis of the optimal schedule $\mathcal{S}$ establishes that the schedule is identical to our schedule $\mathcal{S}^{(\alpha,1)}$. This completes the proof. □

The reader will note that the optimal schedule $\mathcal{S}^{(\alpha,1)}$ for the two-episode, single-interrupt scenario is very similar to the optimal schedule $\mathcal{S}^{(L)}$ for single-episode cycle stealing with a uniform life function (cf. Section 4). Indeed, for U (which is identical to L in the single-episode model) much larger than c, we begin both schedules with a job of length about $\sqrt{2U/c}$, and then select jobs with lengths that successively decrease in length by c, until the interrupt occurs. Hence, the optimal strategy against a maliciously placed unknown interrupt is very similar to the optimal strategy wherein the interrupt will be uniformly distributed, although the two scenarios are rather different.

6.4 Open Problems

Finding an optimal strategy when two or more interrupts are allowed appears to be more difficult, and we will leave matter as an open question. In addition, the problem of devising optimal randomized strategies, for both of the models we have studied here, is of interest. We conjecture that the competitive performance of randomized strategies will be better than that of deterministic strategies, and we leave the matter as another open question.

Acknowledgments. The authors would like to thank David Kaminsky and David Gelernter for discussions that got us started on this work.

The research of S. N. Bhatt at Rutgers was supported in part by ONR Grant N00014-93-0944; the research of F. T. Leighton was supported in part by Air Force Contract OSR-86-0076, DARPA Contract N00014-80-C-0622; the research of A. L. Rosenberg was supported in part by NSF Grants CCR-90-13184 and CCR-92-21785. A portion of this research was done while F. T. Leighton and A. L. Rosenberg were visiting Bell Communications Research.

References

[1] D. Cheriton (1988): The V distributed system. *C. ACM*, 314-333.

[2] E.G. Coffman, Jr., L. Flatto, A.Y. Krenin (1993): Scheduling saves in fault-tolerant computations. *Acta. Inform. 30*, 409-423.

[3] D. Gelernter and D. Kaminsky (1991): Supercomputing out of recycled garbage: preliminary experience with Piranha. Tech. Rpt. RR883, Yale Univ.

[4] M. Litzkow, M. Livny, M. Matka (1988): Condor - A hunter of idle workstations. *8th Ann. Intl. Conf. on Distributed Computing Systems.*

[5] D. Nichols (1990): *Multiprocessing in a Network of Workstations.* Ph.D. thesis, CMU.

[6] J. Ousterhout, A. Cherenson, F. Douglis, M. Nelsom, B. Welch (1988): The Sprite Network Operating System. *IEEE Computer 21*, 6, 23-36.

[7] W.R. Pearson (1993): PVM, the "Parallel Virtual Machine," vs. Net Express. Message on comp.parallel, March 9, 1993.

[8] A. Tannenbaum (1990): Amoeba: a distributed operating system for the 1990s. *IEEE Computer*, 44-53.

[9] S.W. White and D.C. Torney (1993): Use of a workstation cluster for the physical mapping of chromosomes. *SIAM NEWS*, March, 1993, 14-17.