

On Optimal Strategies for Stealing Cycles in Clusters of Workstations^{*}

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.

^{*} 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 (bhatt@bellcore.com)

Fan R. K. Chung: Dept. of Mathematics, Univ. of Pennsylvania, Philadelphia, PA 19104 (chung@math.upenn.edu)

F. Thomson Leighton: Dept. of Mathematics and Lab. for Computer Science, MIT, Cambridge, MA 02139 (ftl@math.mit.edu)

Arnold L. Rosenberg: Dept. of Computer Science, Univ. of Massachusetts, Amherst, MA 01003 (rsnbrg@cs.umass.edu)

Abstract. We study aspects of the parallel scheduling problem for a new modality of parallel computing: having one workstation “steal cycles” from another. We focus on a draconian mode of cycle-stealing, in which the owner of workstation B allows workstation A to take control of B 's processor whenever it is idle, with the promise of relinquishing control *immediately* upon demand. The costs for such cycle-stealing reside in the (typically high) communication overhead for supplying workstation B with work and receiving its results, coupled with the risk that work in progress when the owner of B reclaims the workstation is lost to the owner of A . The first of these costs militates toward supplying B with a large amount of work at once; the second of these costs militates toward repeatedly supplying B with many small packets of work. The challenge is to find balance these two pressures in a way that maximizes the amount of work accomplished.

We formulate two models of cycle-stealing. The first attempts to maximize the work accomplished during a single episode, when one knows the probability distribution of the return of B 's owner. The second attempts to match the productivity of an omniscient cycle-stealer, when one knows how much work the stealer can accomplish. We study cycle-stealing scenarios within each of these models, deriving optimal scheduling strategies for each studied scenario.

We uncover an as-yet unexplained coincidence, two quite distinct scenarios which lead to almost identical unique optimizing schedules. The first of these scenarios falls within our first model: it assumes that the probability of the return of B 's owner is distributed uniformly across the lifespan of the episode. The second of these scenarios falls within our second model: it assumes that B 's owner will interrupt our cycle-stealing at most once during the lifespan of the opportunity.

1 Introduction

1.1 Motivation

Research on parallel computing has focussed mainly on tightly coupled multiprocessors—single machines that are endowed with many (usually identical) processors. Advances in both (networking) hardware and (scheduling) software have combined with economic factors to make networks of workstations (NOWs) an increasingly important milieu for parallel computing [13]. One economic advantage of NOW-based parallel computing is that workstations in a cluster are seldom all constantly busy, giving rise to the phenomenon of *cycle stealing*—the use by one workstation of idle computing cycles of another. Of course, this new modality of parallel computing gives rise to scheduling complications that do not exist with tightly coupled machines, hence has evoked a substantial body of work [1, 3, 4, 5, 7, 8, 9, 10, 14, 17] on languages and/or operating systems that support the peculiarities of scheduling loosely coupled parallel computers. We know of just a few sources in the literature that deal with the problem of scheduling individual computations on NOWs—which is the aspect of cycle-stealing which interests us: [18] describes an experience in explicitly scheduling an application (chromosome mapping) across a far-flung assemblage of workstations (which communicate via email); [2] uses cycle stealing to execute multithreaded computations more efficiently.

The most interesting challenge in scheduling episodes of cycle-stealing resides in the tension between the following inherently conflicting aspects of the problem. On the one hand, a stealer of cycles must subordinate his/her claim to a remote workstation to that of the workstation’s owner. When the owner reclaims the workstation, the cycle-stealer immediately gets some form of reduced service. An extreme form of service reduction would be to kill the cycle-stealer’s job, thereby destroying all work since the last checkpoint. Such extreme reduction occurs, for instance, when a returning owner unplugs a laptop from the network; it occurs also, we are told, in cycle-stealing “contracts” in force at Bellcore and the University of Utah. Of course, a more modest form of reduction would just “nice” the cycle-stealer’s job, i.e., assign it very low priority.

No matter what the details of the service reduction, the fact of the reduction would induce a cycle-stealer to break a cycle-stealing episode into many short *periods*, supplying small amounts of work to the borrowed workstation each time, in order to forestall having work delayed for indeterminate periods, or even lost.

On the other hand, the inter-workstation communications that bracket every period in a cycle-stealing episode—to supply work to the borrowed workstation and to reclaim the results of that work—are, even in the presence of a high-speed LAN, quite expensive, both in setup time and latency.

No matter what the exact cost of these communications, their substantial overhead would induce a cycle-stealer to break a cycle-stealing episode into a few long periods, supplying large amounts of work to the borrowed workstation each time, in order to minimize the slowdown incurred by each communication.

Clearly, the challenge in scheduling episodes of cycle-stealing is to devise a strategy for balancing these conflicting factors in a way that maximizes the productive output of the episode.

1.2 A Preliminary Look at Our Model

In Section 1.3, we formulate our general formal model of the process of cycle-stealing. Informally, our model envisages a master workstation (workstation A , the cycle-stealer) which allocates tasks to client workstations (the ones whose cycles are being stolen). Work is allocated in an “architecture-independent” fashion, in the sense of [11]: the environment in which our workstations operate is characterized by the parameter c which is the communication overhead of a period of cycle-stealing; specifically, c is the (combined) cost of initiating both the communication in which the master workstation sends work to a client workstation, and the communication in which that client returns the results of that work. Our model measures work in discrete units; tasks are indivisible and may require any integral number of work units.¹

In Section 2, we specialize our general model to one which focuses on a single episode of cycle-stealing, starting with the departure of the owner of workstation B and terminating with his/her return. We assume in this model that the master workstation knows the probability distribution of the return of B 's owner; its goal is to allocate work to workstation B in a way that maximizes the expected amount of work accomplished during the episode. In Sections 2.3-2.5, we consider three scenarios within this model, each characterized by a distinct probability distribution; for each, we derive the unique optimal schedule for a single episode of cycle-stealing. We focus on these three scenarios since each represents a plausible real situation, and each requires a different mode of analysis.

In Section 3, we specialize our general model to one which focuses on multi-episodic cycle-stealing, in which the owner of workstation B leaves and returns multiple times. We assume in this model that the master workstation knows how much work an omniscient cycle-stealer—which knows the number and times of returns by B 's owner—could accomplish; the master workstation's goal is to devise a schedule of work allocation which comes as close as possible to matching the productivity of the omniscient cycle-stealer. In Sections 3.2 and 3.3, we derive the unique optimal schedules for two scenarios under this model: Section 3.2 exhibits the unique optimal deterministic oblivious schedule within this model; Section 3.3 exhibits the unique deterministic (adaptive) schedule for the special situation wherein B 's owner returns at most once. The interest in this second, rather artificial scenario is in the fact that its unique optimal schedule is virtually identical to the unique optimal schedule for the apparently quite unrelated scenario of Section 2.4, wherein the probability of the return of B 's owner is distributed uniformly across the lifespan of the episode. This unexpected coincidence is as yet unexplained, but we feel that it may have significance which makes this common scheduling strategy worth further study.

Note that our framework does not mandate who initiates a period; hence, our study fits in with both the distributed scheduling philosophy of [14] and the centralized scheduling philosophy

¹We assume that task lengths are known perfectly.

of [3, 8, 17].

1.3 Details of the General Model

Lifespans. It is clear that no single strategy can optimally schedule all episodes of cycle-stealing, because of the multitude of ways such episodes can arise: some episodes will arise because of multi-week vacations, during which the owner of B is (almost) certain not to appear; others will involve weekends, wherein the presence and frequency of interruptions may depend on B 's owner's current workload; yet other episodes will involve lunch breaks of limited but unpredictable duration; some will involve (almost certainly short) telephone calls. Accordingly, we distinguish two *scenarios*, or classes of episodes, that require somewhat different groundrules. In the *unbounded lifespan* scenario, there is no known *a priori* bound on how long workstation B will remain idle; in the *bounded lifespan* scenario, it is known that workstation B will be idle for at most L time units.

Work schedules. The master workstation schedules a cycle-stealing episode by partitioning the lifespan of workstation B (during the episode) into a sequence of nonoverlapping *periods*. For technical convenience, we identify a *cycle-stealing schedule* \mathcal{S} indirectly, with its sequence of period-lengths: $\mathcal{S} = t_0, t_1, t_2, \dots$. The intended interpretation is that at time

$$\tau_k \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } k = 0 \\ T_{k-1} \stackrel{\text{def}}{=} t_0 + t_1 + \dots + t_{k-1} & \text{if } k > 0 \end{cases}$$

the k th period begins: the master workstation supplies workstation B with an amount of work chosen so that t_k time units are sufficient for

1. the master workstation to send the work to workstation B ;
2. workstation B to perform the work;
3. workstation B to return the results of the work.

In any valid schedule, each period-length t_i is nonnegative and finite, and the two main parameters c, L are positive; in any valid schedule for the *bounded lifespan* scenario, the number of periods m is positive, and the period-lengths sum to the lifespan L : $t_0 + t_1 + \dots + t_{m-1} = L$.

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 k th *tail* of a schedule $\mathcal{S} = t_0, t_1, \dots$, denoted $\phi_k(\mathcal{S})$, is the schedule $\phi_k(\mathcal{S}) = t_k, t_{k+1} \dots$; note that $\phi_0(\mathcal{S}) = \mathcal{S}$.

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

- A shared memory scenario:

- The master workstation “tells” workstation B where to get its data and/or programs.
 - 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:
 - The master workstation sends workstation B its programs and/or data.
 - The master workstation “fills the pipe,” to transmit the necessary data to workstation B .
 - Workstation B “fills the pipe” to transmit its results to the master workstation.

Notes on modeling.

1. Since the costs of the period-by-period communications combine via addition, we simplify our development by combining the overheads of the pre-period and post-period communications into one communication-overhead parameter c . Since the parameter c is typically quite large compared to the time required to compute a task, the master workstation tries to minimize the number of periods in a schedule, hence the number of communications.

2. The reader will note that parameter c is fixed, independent of the amount of data transmitted between the master and client workstations in a single communication. This does not mean that we ignore the marginal pipeline cost of transmitting large amounts of data. Rather, for technical simplification, our model absorbs these marginal costs—the cost of the master’s transmitting that task to the client and the client’s returning its results to the master—into the cost assessed for computing a task.

Work schedules revisited. At time τ_k (the beginning of period k), the master workstation supplies workstation B (either directly or indirectly, as indicated earlier) with a job containing² $w_k \stackrel{\text{def}}{=} t_k \ominus c$ 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 $w_0 + w_1 + \dots + w_{k-1}$. This reckoning reflects the fact that the master workstation loses all work from an interrupted period. 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 $t_i \geq c$) sum to *less than* L .

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

2 Operating with a Known Life Function

2.1 Specializing the Model

In this section we specialize our general model by assuming that we have quantitative knowledge of the risk of being interrupted in the midst of a period. Although we assume exact knowledge of the risk here, one can easily adapt our model to one in which one has only statistical knowledge, say via trace data.

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, we must have

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

- 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: $q(0) = 0$ and $p(0) = 1$.; additionally, in the *bounded lifespan* scenario with lifespan L , $p(t) = 0$ for all $t \geq L$.

We can render our continuous framework discrete, as well as relate the functions p and q , via the *marginal risk function* q^+ :

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

First, we can define p in terms of q :

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

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

$$q^+(t) = p(t) - p(t-1). \quad (2.2)$$

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

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

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 $\mathcal{S} = t_0, t_1, \dots$ 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 [6], wherein one finds some results that are quite close to some of our results, especially Theorem 2.2. Detailed differences in models and dramatic differences in methodology render the development here and in [6] 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 are smaller than 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 $\geq c$. If \mathcal{S} has m periods (in the bounded lifespan scenario), then every period of \mathcal{S} , save possibly the last, has length $\geq c$.

Lemma 2.1 *Let \mathcal{S} be a cycle-stealing schedule for an episode having life function p . There is a productive schedule \mathcal{S}' for the episode such that $E(\mathcal{S}'; p) \geq E(\mathcal{S}; p)$.*

Proof. Say that schedule $\mathcal{S} = t_0, t_1, \dots, t_{k-1}, t_k, t_{k+1}, \dots$ has a nonproductive period k , which is not its last period; i.e., $0 \leq t_k < c$. Construct the schedule $\mathcal{S}^{<k>} = s_0, s_1, \dots$ 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.

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

We claim that, for all life functions p , $E(\mathcal{S}^{<k>}; p) \geq E(\mathcal{S}; p)$. We verify this 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 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. By continuing such combining, one eventually ends up with a productive schedule \mathcal{S}' which accomplishes at least as much work as does \mathcal{S} . \square

In the remaining three subsections of this section, we study three specific scenarios within the model of this section, each scenario being defined via its life function. We derive the optimal cycle-stealing schedules for each scenario.

2.3 The Geometrically Decreasing Lifespan Model

**HERE 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 its 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:

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

Our study of the GDL model focusses on the existence and structure of optimal schedules. In Section 2.3.A, 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 2.1 that schedule $\mathcal{S}^{(a)}$ is the *unique* optimal schedule for the GDL model with risk parameter a . In Section 2.3.B, we prove a weak converse to Theorem 2.1, showing that any cycle-stealing episode for which a uniform schedule is optimal honors a weak analog of the GDL model (Proposition 2.1).

A. Optimal Schedules for the GDL Model

For each risk parameter $a > 1$, consider the uniform schedule $\mathcal{S}^{(a)} \stackrel{\text{def}}{=} t^{(a)}, t^{(a)}, t^{(a)}, \dots$ whose periods have common length $t^{(a)}$ defined implicitly by the equation³

$$t^{(a)} \ln a + a^{-t^{(a)}} = 1 + c \ln a. \quad (2.1)$$

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

$$E(\mathcal{S}^{(a)}; p_a) = \frac{a^{-t^{(a)}}}{\ln a}. \quad (2.2)$$

Theorem 2.1 $\mathcal{S}^{(a)}$ is the unique optimal schedule for the GDL model with risk parameter 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 $\mathcal{S}^{(a)}$.

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

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

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 $\mathcal{S} = t_0, t_1, t_2, \dots$ 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 $\mathcal{S} = t_0, t_1, t_2, \dots$ be any optimal schedule, and consider the tail $\phi_1(\mathcal{S}) = t_1, t_2, \dots$ 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

$$E(\mathcal{S}; p_a) = (t_0 - c) a^{-t_0} + a^{-t_0} E(\phi_1(\mathcal{S}); p_a) \leq (t_0 - c) a^{-t_0} + a^{-t_0} E(\mathcal{S}; p_a). \quad (2.3)$$

This recurrence yields the bound.

$$E(\mathcal{S}; p_a) \leq \frac{t_0 - c}{a^{t_0} - 1}. \quad (2.4)$$

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

$$E(\mathcal{S}_t; p_a) = \frac{t - c}{a^t - 1}, \quad (2.5)$$

so that the expected work of schedule \mathcal{S}_{t_0} matches the bound of inequality (2.4). Three conclusions follow.

1. The bound in (2.4) is, in fact, an equality.
2. The inequality in (2.3) 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 .

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 (2.5). This determination is a straightforward calculus exercise: expression (2.5) has a unique maximum that occurs when t assumes the value $t^{(a)}$ defined implicitly in equation (2.1). The resulting expression (2.2) for $E(\mathcal{S}^{(a)}; p_a)$ is immediate.

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

As we just noted, the expected work of schedule \mathcal{S} is dually given by expression (2.4), via direct derivation, and by expression (2.2), via the preceding analysis. Since $t^{(a)}$ is the *unique* value of t that maximizes expression (2.5), it follows that $t_0 = t^{(a)}$.

If we combine the expression (2.2) for $E(\mathcal{S}; p_a)$ with the expression in (2.3) 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 (2.3), 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. \square

B. A Weak Converse

It is easier to describe the weak converse of Theorem 2.1 which we expose in this section if we consider the following corollary of the Theorem.

If $p_a(t) = ap_a(t + 1)$, then all w_i are equal.

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

$$w \sum_{i=k}^{\infty} q(T_i) \geq p(T_k). \quad (2.6)$$

On the other hand, we have

$$p(T_k - 1) = 1 - \sum_{i=1}^{T_k-1} q(i) = \sum_{i=T_k}^{\infty} q(i) \geq \sum_{i=k}^{\infty} q(T_i). \quad (2.7)$$

Combining Facts (2.6) and (2.7), we find the desired weak converse:

Proposition 2.1 *Say that the uniform schedule $\mathcal{S} = w + c, w + c, \dots$ is optimal for a given cycle-stealing episode. Then for infinitely many t , including all $t \equiv 0 \pmod{w + c}$,*

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

2.4 The Uniform Risk Model

In this section, we consider the *uniform risk (UR)* model, a *bounded lifespan* model wherein the owner of workstation B is likely to be absent for L time units, but wherein the probability of the owner's return is the same at every moment, so that the probability of our retaining control of workstation B decreases at a fixed constant rate throughout the episode.

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 $0 \leq t \leq L$.

By Lemma 2.1, we may restrict our search for an optimal schedule for the UR model to productive schedules. Simplifying our search is the following result, which asserts that the sought schedule must have periods whose lengths are nondecreasing.

Lemma 2.2 *If the productive schedule $\mathcal{S} = t_0, t_1, \dots, t_{m-1}$ is optimal for the UR model, then $t_0 \geq t_1 \geq \dots \geq t_{m-1}$.*

Proof. Let t_k be the first period-length that violates monotonicity, i.e., for which $t_k < t_{k+1}$; clearly, $k < m - 1$ since \mathcal{S} has only m periods. Let us compare the expected work of schedule \mathcal{S} with that of schedule $\widehat{\mathcal{S}}$ which interchanges \mathcal{S} 's k th and $(k + 1)$ th periods: $\widehat{\mathcal{S}} = \hat{t}_0, \hat{t}_1, \dots, \hat{t}_{m-1}$, where

$$\hat{t}_i = \begin{cases} t_{k+1} & \text{if } i = k \\ t_k & \text{if } i = k + 1 \\ t_i & \text{otherwise.} \end{cases}$$

Instantiating expression (2.3) with schedules \mathcal{S} and $\widehat{\mathcal{S}}$ under risk function p_L , we obtain

$$\begin{aligned} E(\mathcal{S}) &= \sum_{i=0}^{m-1} (t_i \ominus c) \left(1 - \frac{1}{L} T_i\right) \\ E(\widehat{\mathcal{S}}) &= \sum_{i=0}^{m-1} (\hat{t}_i \ominus c) \left(1 - \frac{1}{L} \hat{T}_i\right), \end{aligned}$$

where the \hat{T}_i are obtained from the \hat{t}_i just as the T_i are obtained from the t_i .

Since \mathcal{S} is a productive schedule, we know that $t_k \geq c$, so that $t_{k+1} \geq c$ also. In the light of this, direct evaluation yields

$$E(\widehat{\mathcal{S}}) - E(\mathcal{S}) =$$

$$(t_{k+1} - t_k) \left(1 - \frac{1}{L} T_{k-1}\right) + (t_k - c) \left(1 - \frac{1}{L} (T_{k-1} + t_{k+1})\right) - (t_{k+1} - c) \left(1 - \frac{1}{L} (T_{k-1} + t_k)\right).$$

After simplification, we find that

$$E(\widehat{\mathcal{S}}) - E(\mathcal{S}) = \frac{c}{L}(t_{k+1} - t_k) > 0.$$

This contradicts the assumed optimality of schedule \mathcal{S} , hence proves the lemma. \square

The optimal schedule. We now present the unique optimal schedule for the UR model. This schedule partitions the anticipated lifespan into periods whose lengths form a decreasing arithmetic sequence with common difference c (the communication overhead parameter); moreover, these periods are maximal in number, given this rate of decrease.

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

$$m = \left\lfloor \sqrt{\frac{2L}{c} + \frac{1}{4}} + \frac{1}{2} \right\rfloor, \quad (2.1)$$

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

$$t_i^{(L)} = \frac{L}{m} + \frac{1}{2}c(m-1) - ci. \quad (2.2)$$

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

$$E(\mathcal{S}^{(L)}; p_L) = \frac{L}{2} - mc - \frac{L}{2(m+1)} + \frac{m(m+1)(m+2)c^2}{24L}. \quad (2.3)$$

We now verify that schedule $\mathcal{S}^{(L)}$ has the claimed optimality properties.

Theorem 2.2 $\mathcal{S}^{(L)}$ is the unique optimal productive schedule for the UR model with lifespan L .

Proof. Let us consider an arbitrary optimal *productive* schedule, $\mathcal{S} = t_0, t_1, \dots, t_{m-1}$, for the UR model with lifespan L . (By Lemma 2.1, if there is any optimal schedule, there is an optimal productive one.) We shall deduce properties of \mathcal{S} that will specify values for the $m+1$ unknowns here: the number of periods m , and the period lengths t_0, t_1, \dots, t_{m-1} . This instantiation of values will establish that schedule \mathcal{S} in fact exists (as we noted earlier, some life functions do not admit optimal schedules) and that it actually coincides with schedule $\mathcal{S}^{(L)}$.

Defining the problem. We begin by instantiating expression (2.3) with schedule \mathcal{S} and risk function p_L :

$$E(\mathcal{S}) = \sum_{i=0}^{m-1} (t_i \ominus c) \left(1 - \frac{1}{L}T_i\right). \quad (2.4)$$

We remark that the three main parameters here, namely, c, L, m , are all positive, and the period-lengths $\{t_i\}$ sum to L (see Section 1.3); moreover, since \mathcal{S} is a productive schedule, we also have (see Lemma 2.1)

$$t_i \geq c \text{ for all } 0 \leq i < m-1; \text{ and } t_{m-1} \geq 0. \quad (2.5)$$

Two simplifications. Next, we make two simplifications to expression (2.4), which will help us infer values for its $m + 1$ unknowns, i.e., values which constitute a maximizing assignment to the unknowns. The first simplification, which is justified by constraints (2.5), substitutes ordinary subtraction for positive subtraction in all terms of (2.4) save the last, yielding the expression

$$\sum_{i=0}^{m-2} (t_i - c) \left(1 - \frac{1}{L} T_i\right) + (t_{m-1} \ominus c) \left(1 - \frac{1}{L} T_{m-1}\right). \quad (2.6)$$

The second simplification is much more drastic. First, we substitute ordinary subtraction for positive subtraction in the last term of expression (2.6) also, yielding the expression

$$\sum_{i=0}^{m-1} (t_i - c) \left(1 - \frac{1}{L} T_i\right). \quad (2.7)$$

Second, we replace the productivity constraints (2.5) by the much weaker constraints

$$t_i \geq 0 \quad \text{for all } 0 \leq i < m. \quad (2.8)$$

This problem transformation cannot be justified *a priori*, for it broadens the space over which we are searching for a maximizing assignment to the unknowns. However, we shall see that the unique maximizing assignment for expression (2.7) with constraints (2.8), over the broader space, actually lies within the space of sought maximizing assignments for expression (2.6) with constraints (2.5); hence, the assignment we end up with will also maximize expression (2.4) with constraints (2.5).

Exposing structure. We continue with our search for an optimal schedule by performing two transformations of expression (2.7) which expose its underlying structure. First, By direct manipulation, plus the fact that $L^2 = (t_0 + t_1 + \cdots + t_{m-1})^2$, we find that

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

the last expression resulting from completing the square. Finally, we let $u_i = t_i + ci$ for $0 \leq i < m$, and we divide by L (to compensate for having factored L out earlier), to arrive (after some elementary manipulation) at the expression for $E(\mathcal{S}^{(L)})$ which we shall actually work with.

$$E(\mathcal{S}) = \frac{1}{2} L + \frac{c^2 m(m-1)(2m-1)}{12L} - \frac{1}{2L} \sum_{i=0}^{m-1} u_i^2. \quad (2.9)$$

Seeking a maximizing assignment. Our goal now is to maximize expression (2.9) subject to the following constraints.

1. Since each $t_i \geq 0$, we have each $u_i \geq ci$.
2. Since $t_0 + t_1 + \dots + t_{m-1} = L$, we have

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

Easily, we shall have *maximized* expression (2.9) once we have *minimized* the sum

$$u_0^2 + u_1^2 + \dots + u_{m-1}^2 \tag{2.10}$$

subject to these same constraints.

Now, were it not for constraint (1.), we could minimize the sum (2.10) simply by setting each u_i to the average value of the u_i :

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

Constraint (1.) demands that we employ a more complicated minimization procedure. Specifically, we use the minimize-by-averaging technique for the largest set of u_i that the bounds of constraint (1.) will permit; and we assign to each of the other u_i the smallest value that the constraint permits, namely, $u_i = ci$. In the light of Lemma 2.2, this means that we use the minimize-by-averaging technique for the first $r+1$ of the u_i , where r is the largest index such that u_r 's "average value" is large enough, i.e., such that

$$\frac{L}{r+1} + \frac{cr}{2} \geq cr. \tag{2.11}$$

We digress to verify that the test of inequality (2.11) is legitimate, i.e., that the lefthand side of the inequality is, indeed, the average of the first $r+1$ unknowns u_i . To this end, we note that the direct expression for this average is

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

Because of constraint (1.), we can instantiate specific values for the u_i that are subtracted off here, yielding the following, which converts readily to our expression.

$$\begin{aligned} (r+1)A_r &= L + c \binom{m}{2} - \sum_{i=r+1}^{m-2} ci \\ &= L + c \binom{r+1}{2}. \end{aligned}$$

For the tail of sum (2.10) which cannot be minimized via 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 .

$$u_i = \begin{cases} L/(r+1) + cr/2 & \text{for } i \leq r \\ ci & \text{for } i > r \end{cases} \tag{2.12}$$

It remains only to determine the maximizing value of r . This is done easily by rewriting inequality (2.11) in the form $r^2 + r - 2L/c \leq 0$. It is now clear that the maximizing value of r is given by

$$r = \left\lfloor \sqrt{\frac{2L}{c} + \frac{1}{4}} - \frac{1}{2} \right\rfloor. \quad (2.13)$$

Return to the $\{t_i\}$. Now, we return to our original setting and convert the minimizing assignment (2.12) for the $\{u_i\}$ to the sought maximizing assignment for the $\{t_i\}$:

$$t_i = \begin{cases} L/(r+1) + cr/2 - ci & \text{for } i \leq r \\ 0 & \text{for } i > r \end{cases} \quad (2.14)$$

One sees from assignment (2.14) that:

- the first r periods of schedule \mathcal{S} (those of lengths t_0, t_1, \dots, t_{r-1}) are guaranteed to be productive;
- all periods beyond the first $r+1$ have zero length, hence contribute no work.

It follows that the optimal productive schedule $\mathcal{S} = t_0, t_1, \dots, t_{m-1}$ has $m = r+1$ periods which verifies equation (2.1). If we now instantiate assignment (2.14) with this value of m , we arrive at the sought values for the period lengths $\{t_i\}$ specified in expression (2.2).

ALR: I ran out of steam before I checked the upcoming expression for the expected work. It is bound to be off a bit, because we were off a bit on the value of m !

We complete the proof by verifying that our maximizing assignments for m and $\{t_i\}$ yield equation (2.3) as the expected amount of work performed by the optimal schedule. We accomplish this by direct evaluation of expression (2.9) with the maximizing values of all parameters.

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

Equation (2.3) now follows by direct calculation. \square

For the sake of perspective, we note that $E(\mathcal{S}^{(L)}; p_L)$ is very close to $\frac{1}{2}L - \frac{7}{6}\sqrt{2cL}$ when L is very much larger than c (which is likely to be the case).

There is an obvious simple heuristic approximation \mathcal{S}' to the somewhat complicated optimal schedule $\mathcal{S}^{(L)}$. \mathcal{S}' would have the right “order” of number of periods: \sqrt{L} , and the right “order” of period-lengths: \sqrt{L} , but it would “obliviously” have all periods of equal length. Direct calculation using expression (2.7), and comparison with expression (2.3), demonstrates that there is a significant penalty for such simplification whenever c is large: $E(\mathcal{S}'; p_L)$ is only $\frac{1}{2}L + \frac{1}{2}\sqrt{L} - \frac{1}{2}c(\sqrt{L} + 1)$.

2.5 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 2.3, 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:

- $q_L^g(t+1) = 2q_L^g(t) = (2^L - 1)2^{-(t-1)}$ for $t \geq 1$.
- $p_L^g(t) = (2^L - 2^t)/(2^L - 1)$ 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 k th period is exponentially longer than the $(k+1)$ th period.

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

$$m = \log^* L - \log^* c, \quad (2.1)$$

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

$$\hat{t}_k^{(L)} = 2^{\hat{t}_{k+1}^{(L)}} + c - 2 \quad \text{for } k \in \{0, 1, \dots, m-2\} \quad (2.2)$$

$$\hat{t}_{m-1}^{(L)} = L - \left(\hat{t}_0^{(L)} + \hat{t}_1^{(L)} + \hat{t}_{m-2}^{(L)} \right). \quad (2.3)$$

Theorem 2.3 $\widehat{\mathcal{S}}^{(L)}$ is the optimal schedule for the GIR model with lifespan L .

Proof. Let $\mathcal{S} = t_0, t_1, \dots, t_{m-1}$ 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

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

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

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

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

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.

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

The nonnegativity of this difference implies that

$$t_k \geq 2^{t_{k+1}} + c - 2. \quad (2.4)$$

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.

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

The nonnegativity of this difference implies that

$$t_k \leq 2^{t_{k+1}} + c - 2. \quad (2.5)$$

Inequalities (2.4) and (2.5) combine to verify that the period-lengths of the optimal schedule \mathcal{S} satisfy the system of equations (2.2) that defines schedule $\widehat{\mathcal{S}}^{(L)}$.

To verify equation (2.1), 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 (2.1), whence the theorem. \square

3 Operating with an Unknown Life Function

The model we studied in Section 2 focusses on just a single episode of cycle stealing. We turn now to an alternative model which considers opportunities for cycle-stealing with multiple episodes. In contrast to the previous model, which tries to maximize expected work, assuming knowledge of life functions, the present model tries to maximize the guaranteed amount of work, assuming knowledge of how much work could be achieved by a scheduler that knew the exact number and times of the returns of workstation B 's owner. This is another idealized assumption, of course, yet it seems to be a reasonable early step in understanding the rather new phenomenon of cycle-stealing. (One must assume some knowledge about a cycle-stealing opportunity in order to be able to understand the phenomenon in any analytical setting. If nothing at all is known about the opportunity, then there is 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.)

3.1 Competing against Omniscience

The model we present now demands schedules for multi-episode cycle-stealing, in the presence of the following two pieces of knowledge.

1. The master workstation knows the *usable lifespan* U of the cycle-stealing opportunity. This quantity is defined as follows. The owner of workstation B will be absent for episodes of durations L_0, L_1, \dots . We define U as the sum of those L_i which exceed c . We call U the “usable lifespan” since even an omniscient scheduler can get no work done during an episode of length $\leq c$.
2. The master workstation knows that the optimal *omniscient* scheduler \mathcal{S}^* , which does know which, and how many L_i exceed c , is able to accomplish $W^* \stackrel{\text{def}}{=} \alpha U$ units of work during the opportunity, for some $\alpha < 1$.⁶

We wish to stress that, even though the master workstation knows U , it does *not* know which, or how many L_i exceed c (although schedule \mathcal{S}^* does know both of these).

Our goal in this model is to design a schedule $\mathcal{S}^{(\alpha)}$ which is guaranteed to accomplish almost as much work as \mathcal{S}^* , independent of the number and location(s) of the interrupts. Specifically, we judge the quality of schedule $\mathcal{S}^{(\alpha)}$ by its *competitive ratio* with \mathcal{S}^* , i.e., the ratio

$$\frac{W^{(\alpha)}}{W^*} = \frac{\beta_\alpha}{\alpha},$$

where $W^{(\alpha)}$ is the amount of work that $\mathcal{S}^{(\alpha)}$ accomplishes.

⁶If this opportunity arises from a situation when B 's owner is typically away from work, then we can expect α to be close to 1.

This model differs from that of Section 2 in three important respects. First, we now focus on schedules which encompass multiple episodes of cycle-stealing, not only one. Second, we now assume no knowledge of the life function of any particular episode. Third, we now measure the performance of a candidate schedule by comparing its *guaranteed* (hence, worst-case) work output to that of the omniscient schedule \mathcal{S}^* .

3.2 Optimal Deterministic Oblivious Multi-Episode Schedules

The scenario we study first produces schedules for arbitrary multi-episode cycle-stealing opportunities, which 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 i th 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 i th 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 uniform⁷ multi-episode cycle-stealing schedule whose periods have common length $t^{(\alpha)} \stackrel{\text{def}}{=} c/\sqrt{1 - \alpha}$.

Theorem 3.1 *For all $\alpha < 1$, schedule $\mathcal{S}^{(\alpha)}$ accomplishes at least*

$$W(\mathcal{S}^{(\alpha)}) = (1 - \sqrt{1 - \alpha})^2 U \tag{3.6}$$

units of work, which is at least the fraction

$$\frac{W(\mathcal{S}^{(\alpha)})}{W^*} = \frac{1 - \sqrt{1 - \alpha}}{1 + \sqrt{1 - \alpha}} \tag{3.7}$$

of the work achieved by the omniscient schedule \mathcal{S}^ . 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 length- U usable portion of the cycle-stealing opportunity consists of k episodes, of lifespans L_0, L_1, \dots, L_{k-1} , respectively, each of length $L_i > c$. Clearly, based on its perfect overview, for each i , schedule \mathcal{S}^* will accomplish $L_i - c$ units of work during the i th episode (which has duration L_i). Since \mathcal{S}^* accomplishes at least $W^* = \alpha U$ units of work in all, we know that

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

⁷By extending our earlier terminology, we call a multi-episode cycle-stealing schedule *uniform* if all of its periods have the same length.

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

$$\text{Number-of-Episodes} = k - 1 < \left\lfloor \frac{(1 - \alpha)U}{c} \right\rfloor, \quad (3.8)$$

and so that the “average” episode has duration

$$\text{Average-Duration} = \frac{U}{k} \geq \frac{c}{1 - \alpha}. \quad (3.9)$$

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:

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 (3.9).

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 (3.8), 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 (3.6). Comparing this amount of work with the amount of work W^* accomplished by the omniscient schedule \mathcal{S}^* , we find that schedule $\mathcal{S}^{(\alpha)}$ achieves the “competitive ratio” given in expression (3.7). Thus, when α is close to 1, the work accomplished under schedule $\mathcal{S}^{(\alpha)}$ is close to L , and the competitive ratio $W^{(\alpha)}/W^*$ 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}^* can accomplish at least αU units of work. To this end, let us consider an arbitrary deterministic, oblivious schedule⁸ $\mathcal{S} = t_0, t_1, \dots, t_n$ satisfying the following.

⁸Note 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.

1. Each $t_i > c$, so that some work will be accomplished during each uninterrupted period.
2. $t_0 + t_1 + \dots + t_n \leq U$, 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

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

where the summation ranges over the set $S \subseteq \{0, 1, \dots, n\}$ of indices of uninterrupted periods. By inequality (3.8), 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. \square

3.3 Optimal Adaptive Two-Episode Schedules

Devising optimal *adaptive* cycle-stealing schedules—whose period-lengths are chosen based on the lengths of prior periods and the pattern of prior interrupts—is far more difficult than devising optimal 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. ****HERE Remarks.** (a) The optimal omniscient schedule for this scenario, call it \mathcal{S}_1^* , can accomplish at least $W_1^* \stackrel{\text{def}}{=} U - 2c$ 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 $t_0^{(\alpha,1)}, t_1^{(\alpha,1)}, \dots, t_{\ell-1}^{(\alpha,1)}$ 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 ℓ th 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 ℓ^* , to be the integer that maximizes the expression

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

Note that ℓ^* lies in the range

$$\binom{\ell^*}{2} \leq \frac{U}{c} - 1 \leq \binom{\ell^* + 1}{2}$$

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

$$t_j^{(\alpha,1)} \stackrel{\text{def}}{=} \frac{1}{\ell^*}(U - c) + \frac{c}{2}(\ell^* - 1) - jc = t_0^{(\alpha,1)} - jc; \quad (3.10)$$

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

$$t_0^{(\alpha,1)} + t_1^{(\alpha,1)} + \dots + t_{\ell^*-1}^{(\alpha,1)} = U. \quad (3.11)$$

Theorem 3.2 *For all $\alpha < 1$, schedule $\mathcal{S}^{(\alpha,1)}$ accomplishes at least*

$$W(\mathcal{S}^{(\alpha,1)}) = U - t_0^{(\alpha,1)} = \frac{\ell^* - 1}{\ell^*}(U - c) - \frac{c}{2}(\ell^* - 3) \quad (3.12)$$

units of work, which is at least the fraction

$$\frac{W(\mathcal{S}^{(\alpha,1)})}{W_1^*} = \left(1 - \sqrt{\frac{c}{2U}}\right) + \text{l.o.t.} \quad (3.13)$$

of the work accomplished by the omniscient single-interrupt schedule \mathcal{S}_1^ . 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 $\mathcal{S} = t_0, t_1, \dots, t_{\ell-1}$ which, in common with schedule $\mathcal{S}^{(\alpha,1)}$:

- exhausts the useful lifespan, in the sense that

$$t_0 + t_1 + \dots + t_{\ell-1} = U; \quad (3.14)$$

- 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 (3.14), 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 $i \in \{0, 1, \dots, \ell-1\}$. Then,

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

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

units of work.

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

$$W(\mathcal{S}) = U - t_i - ic \ominus c \tag{3.15}$$

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 (3.12), irrespective of whether or where the interrupt occurs. Direct calculation now verifies that schedule $\mathcal{S}^{(\alpha,1)}$ achieves the competitive ratio of expression (3.13).

The optimality of schedule $\mathcal{S}^{(\alpha,1)}$. Let us consider again the general ℓ -period schedule $\mathcal{S} = t_0, t_1, \dots, t_{\ell-1}$ 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 (3.15), then we find the following.

- If the adversary interrupts schedule \mathcal{S} during period $i \in \{0, 1, \dots, \ell-2\}$, 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:

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

This is equivalent to setting period lengths so that

$$\begin{aligned} t_j &= t_0 - jc \quad \text{for } j \in \{0, 1, \dots, \ell - 2\} \\ t_{\ell-1} &= t_{\ell-2} \end{aligned} \tag{3.16}$$

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 (3.16) combines with the fact that the period lengths exhaust the useful lifespan U (expression (3.14)) to yield

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

so that

$$t_0 = \frac{1}{\ell}(U - c) + \frac{c}{2}(\ell - 1). \tag{3.17}$$

This is precisely the functional form of $t_0^{(\alpha,1)}$, with ℓ^* substituted for ℓ ; see expression (3.10).

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 (3.16) and (3.17), 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. (3.12)). Now, since schedule \mathcal{S} is optimal, its number of periods ℓ must maximize this guarantee. Recall, however, that the number of periods ℓ^* of schedule $\mathcal{S}^{(\alpha,1)}$ was chosen precisely to maximize this expression. We infer that $\ell = \ell^*$.

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. \square

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 2.4). 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.

4 Prospects for Future Work

We have just noted that the schedule that starts with a job of length about $\sqrt{2U/c}$, and then select jobs with lengths that successively decrease in length by c is optimal under two quite different scenarios. It is an inviting challenge to explain this coincidence and to determine the range of scenarios wherein this general strategy is optimal.

Our models assume a variety of exact knowledge that one may have only inexactly in real situations. It should not be difficult to relax some these assumptions, by allowing one to have just approximate knowledge of both life functions and task lengths. We would expect the results of Section 2 to change very little under this relaxed model.

Next, the problem of finding optimal scheduling strategies when two or more interrupts are allowed appears to be quite difficult, for both of the models we have studied here. In the context of both of our models, but especially the probabilistic model of Section 2, it would be interesting to study the scheduling problem when interrupted processes are “niced” (assigned very low priority) rather than killed.

Finally, we have yet to explore the problem of devising efficient *randomized* scheduling strategies for either of our models. In the context of the competitive model of Section 3, we conjecture that the competitive performance of even simple randomized strategies will be better than that of deterministic strategies.

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 Grant 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] R.D. Blumofe, C.F. Joerg, B.C. Kuszmaul, C.E. Leiserson, K.H. Randall, Y. Zhou (1995): Cilk: an efficient multithreaded runtime system. *5th ACM SIGPLAN Symp. on Principles and Practice of Parallel Programming*.
- [2] R.D. Blumofe and C.E. Leiserson (1994): Scheduling multithreaded computations by work stealing. *35th IEEE Symp. on Foundations of Computer Science*.
- [3] S.J. Chapin (1995): Distributed scheduling support in the presence of autonomy. *4th Heterogeneous Computing Wkshp.*, 22-29.

- [4] S.J. Chapin (1995): Preliminary performance results for MESSIAHS. *Bull. IEEE TC on Operating Systems and Application Environments* 7, 12-23.
- [5] D. Cheriton (1988): The V distributed system. *C. ACM*, 314-333.
- [6] E.G. Coffman, Jr., L. Flatto, A.Y. Krenin (1993): Scheduling saves in fault-tolerant computations. *Acta. Inform.* 30, 409-423.
- [7] D. Gelernter and D. Kaminsky (1991): Supercomputing out of recycled garbage: preliminary experience with Piranha. Tech. Rpt. RR883, Yale Univ.
- [8] M. Litzkow, M. Livny, M. Matka (1988): Condor - A hunter of idle workstations. *8th Ann. Intl. Conf. on Distributed Computing Systems*.
- [9] D. Nichols (1990): *Multiprocessing in a Network of Workstations*. Ph.D. thesis, CMU.
- [10] J. Ousterhout, A. Cherenon, F. Douglass, M. Nelsom, B. Welch (1988): The Sprite Network Operating System. *IEEE Computer* 21, 6, 23-36.
- [11] C.H. Papadimitriou and M. Yannakakis (1990): Towards an architecture-independent analysis of parallel algorithms. *SIAM J. Comput.* 19, 322-328.
- [12] W.R. Pearson (1993): PVM, the "Parallel Virtual Machine," vs. Net Express. Message on comp.parallel, March 9, 1993.
- [13] G.F. Pfister (1995): *In Search of Clusters*. Prentice-Hall, Upper Saddle River, N. J.
- [14] M. Stumm (1988): The design and implementation of a decentralized scheduling facility for a workstation cluster. *2nd IEEE Conf. on Computer Workstations*, 12-22.
- [15] V.S. Sunderam (1990): PVM: A framework for parallel distributed computing. *Concurrency: Practice and Experience* 2.
- [16] A. Tannenbaum (1990): Amoeba: a distributed operating system for the 1990s. *IEEE Computer*, 44-53.
- [17] M.M. Theimer and K.A. Lantz (1989): Finding idle machines in a workstation-based distributed environment. *IEEE Trans. Software Engineering* 15, 1444-1458.
- [18] 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.