Guidelines for Data-Parallel Cycle-Stealing in Networks of Workstations, II: On Maximizing Guaranteed Output

by
Arnold L. Rosenberg

CMPSCI Technical Report 98-34
August, 1998

Guidelines for Data-Parallel Cycle-Stealing in Networks of Workstations, II: On Maximizing Guaranteed Output

Arnold L. Rosenberg
Department of Computer Science
University of Massachusetts

Address Information:

Arnold L. Rosenberg
Department of Computer Science
University of Massachusetts
Amherst, MA 01003, USA

Contact Information:

TEL:	413/545-2743
FAX:	413/545-1249
E-MAIL:	rsnbrg@cs.umass.edu

Acknowledgment of Support:

This research was supported in part by NSF Grant CCR-97-10367.

Keywords: Cluster computing; cycle stealing; data-parallel computation; networks of workstations; parallel scheduling; quality guarantees

Abstract

We derive efficient guidelines for scheduling data-parallel computations within a draconian mode of cycle-stealing in networks of workstations. In this computing regimen, (the owner of) workstation A contracts with (the owner of) workstation B to take control of B's processor for a guaranteed total of U time units, possibly punctuated by up to some prespecified number p of interrupts which kill any work A has in progress on B. (Without a mechanism such as the bound p to curtail a "malicious adversary", it would be impossible to guarantee any work production.) On the one hand, the high overhead—of c time units—for setting up the communications that supply workstation B with work and receive its results recommends that A supply B with large amounts of work at a time. On the other hand, the risk of losing work in progress when workstation B is interrupted recommends that A supply B with a long sequence of small bundles of work. In this paper, we derive two sets of scheduling guidelines that balance these conflicting pressures in a way that optimizes, up to low-order additive terms, the amount of work that A is *guaranteed* to accomplish during the cycle-stealing opportunity, no matter when the opportunity is interrupted—up to p times. Our first set of guidelines schedule a cycle-stealing opportunity *non-adaptively*—cleaving to a single fixed strategy until all p interrupts have occurred; the produced schedules achieve at least $U - \sqrt{2pcU} + pc$ units of work. Our second set of guidelines schedule a cycle-stealing opportunity *adaptively*—changing strategy after each interrupt; the produced schedules achieve at least $U - 2\sqrt{2cU} \pm$(low-order terms) units of work. By deriving the theoretical underpinnings of both sets of guidelines, we show that our non-adaptive schedules are optimal in guaranteed work production and that our adaptive schedules are within low-order additive terms of being optimal.

1 Cluster-Based Computing and Cycle-Stealing

Numerous sources eloquently argue the technological and economical inevitability of an increasingly common modality of parallel computation, the use of a network of workstations (NOW) as a parallel computer; see, e.g., [8, 10]. Sources too numerous to list describe systems that facilitate the mechanics of cluster-based computation, especially through its common realization via *cycle stealing*—the use by one workstation of idle computing cycles of another. However, rather few sources study the problem of scheduling individual computations on NOWs, and even fewer develop abstract models that facilitate such scheduling for broad classes of computations. In the current paper, we refine the abstract model developed in [3] and derive guidelines for crafting cycle-stealing schedules for data-parallel computations, that approximately maximize the amount of work that one is *guaranteed* to accomplish during a cycle-stealing opportunity.

1.1 Background for Our Study

In [3], we developed and studied a mathematical model for the problem of scheduling data-parallel computations under the following rather draconian version of cycle-stealing. The owner of workstation A contracts with the owner of workstation B to take control of B's processor for a guaranteed lifespan of U time units, subject to possible interruptions that kill any active job(s)—thereby destroying all work since the last checkpoint.

> Such a draconian cycle-stealing "contract" is inevitable, for instance, when workstation B is a laptop that can be unplugged from the network. Such "contracts" are reported to be quite popular at many institutions, because of the degradation in service that B's owner receives when A's jobs remain active, even with lowered priority.

This "contract" creates a challenging tension between the following inherently conflicting aspects of the problem of stealing cycles. On the one hand, the threat of losing any work in progress when workstation B is interrupted recommends that the owner of workstation A break each cycle-stealing opportunity into many short "periods", supplying small amounts of work to B each time. On the other hand, the typically expensive setup time for the inter-workstation communications that bracket each period—to supply work to B and to reclaim the results of that work—recommends that the owner of A break each opportunity into a small number of long periods, supplying large amounts of work to B each time. The challenge in scheduling a cycle-stealing opportunity effectively is to balance these conflicting factors in a way that maximizes (some notion of) the work achieved during the opportunity.

The model developed in [3] is two-faceted, comprising one submodel that focuses on the *expected* work-output of a cycle-stealing opportunity and one—the one we study here—that focuses on the *guaranteed* work-output of the opportunity. Recognizing that cycle-stealing can accomplish productive work only if the metaphorical "malicious adversary" is restrained from just interrupting every period when A sends work to B, just before B returns its results—thereby nullifying the entire cycle-stealing opportunity—both submodels in [3] assume certain idealized knowledge that restrains the "adversary". The *guaranteed-output* submodel, which is our particular interest here, assumes that the owner of A knows both the total amount of time that workstation B will be available (the opportunity's *usable lifespan*) and an upper bound on the number of possible interruptions.

We derived in [3] exactly optimal cycle-stealing schedules for a small number of specific scenarios under each of the two submodels; however, the techniques used were specific to each scenario. In the current paper and its companion [11], we have sought broadly applicable guidelines that allow one to craft schedules for large classes of cycle-stealing scenarios that (nearly) optimize various measures of work production. In [11], we focused on the expected-output submodel; here, we focus on the guaranteed-output submodel. This paper is thus a second step in our program of rendering prescriptive the descriptive study of cycle-stealing in [3].

1.2 Our Main Results

The primary goal of the current study is to craft guidelines for developing schedules for data-parallel cycle-stealing whose guaranteed work production is close to optimal. In Section 2, we prepare for this goal by refining the guaranteed-output submodel of [3] in a way that makes the goal formal and precise. In Section 3, we present two computationally efficient sets of scheduling guidelines. The first set of guidelines (in Section 3.1) produce schedules that are *non-adaptive*, in the sense that they do not change their scheduling strategy until all possible interrupts have occurred. The second set of guidelines (in Section 3.2), which are the main focus of our study, produce schedules that are *adaptive*, in the sense that they change their scheduling strategy in response to each interrupt. The guidelines we derive produce schedules that achieve the following work production during a cycle-stealing opportunity with usable lifespan U and up to p ("maliciously" placed) potential interrupts. Our non-adaptive guidelines produce a schedule that is guaranteed to achieve at least[1] $U - \sqrt{2pcU} + pc$ units of work; this work production is easily shown to be optimal among non-adaptive schedules. Our adaptive guidelines produce a schedule that is guaranteed to achieve at least[2] $U - 2\sqrt{2cU}\pm$(low-order terms) units of work. This work production is optimal to within low-order additive terms, but demonstrating this near-optimality requires further insights into structure of optimal schedules. We develop these insights in Section 4, where we derive the theoretical results that motivate our adaptive guidelines. We then use these results in Section 5 to evaluate our adaptive schedules' guaranteed work production and to establish their near-optimality.

> We shall see in Section 5.3 that the results in Section 4 actually allow one to craft adaptive schedules whose guaranteed work production is superior to our schedules'—

[1] Recall that c denotes the setup overhead of each inter-workstation communication.

[2] The dependence of our adaptive schedules' work production on the parameter p is not discernible here, since it resides solely in the "low-order terms".

but this superiority can be only in low-order additive terms, and at the expense of a significant penalty in computational overhead.

1.3 Related Work

We briefly review chronologically the few other algorithmic studies of cycle-stealing that appear in the literature, emphasizing the way in which the activity is approached. We do not discuss the many empirical studies of cycle-stealing whose main foci are either systems that enable the activity or specific applications, rather than analyzed scheduling algorithms.

One finds in [12, 13] two opposing philosophies for scheduling cycle-stealing opportunities—"pushing", wherein a loaded workstation tries to find a idle colleague to adopt a set of tasks, and "pulling", wherein an idle workstation seeks work—but neither formal models nor rigorous analyses. In [1], a cycle-stealing schedule within a NOW is crafted by "auctioning off" large identical chunks of a compute-intensive task, to determine the sub-NOW that promises the best parallel speedup (computed using the authors' model); one then distributes appropriate-size chunks of the task within the "winning" sub-NOW. The companion papers [5, 6] present and analyze a system that schedules directed acyclic graphs on a NOW in a way that optimizes system time and space requirements. In [2], cycle-stealing is viewed as one application, among many, of a theory of how to make random decisions better than by random choices; with high probability, the proposed randomized scheduling algorithm accomplishes an amount of work that is within a logarithmic factor of optimal. Finally, in [3] (and its progeny, [11] and the current paper), cycle-stealing is viewed as a game against a malicious adversary who seeks to interrupt the borrowed workstation in order to minimize the work production of a cycle-opportunity. In the expected-output submodel of [3, 11], one assumes that the cycle-stealer knows the instantaneous probability of still controlling the borrowed workstation at time t (the opportunity's "life function"); we have already discussed the knowledge assumed under the guaranteed-output submodel of [3] and the current paper.

2 A Formal Model of Data-Parallel Cycle-Stealing

The cycle-stealing model that we study here derives from the guaranteed-output submodel of [3] but differs from that submodel in important details.

2.1 The General Framework

We schedule data-parallel cycle-stealing opportunities in an "architecture-independent" fashion, in the sense of [9]: the cost of inter-workstation communications is characterized by a single (overhead) parameter c, which is the (combined) cost of initiating (setting up) the paired communications in which workstation A sends work to workstation B and B returns the results of the work. We assume that: tasks are indivisible; task times may vary but are known perfectly; the time allotted to a task includes the marginal cost of transmitting its input and output data.

> The described framework: (a) allows us to keep c independent of the sizes of data transmissions; (b) does not mandate who initiates a transfer of work from A to B, hence is consistent with both the "pull"-oriented scheduling philosophy of [12] and the "push"-oriented philosophy of [13].

For the purposes of our study, a cycle-stealing opportunity is characterized by two quantities that are prespecified, hence, known to the owner of workstation A:

1. the *usable lifespan* $U > 0$ of the opportunity, which is the number of time units during which workstation B will be available to workstation A;
2. an upper bound p on the *potential* number of *interrupts* that will occur because of the return of B's owner during the usable lifespan.

Moderating the idealization inherent in giving the owner of A foreknowledge of p, we give him/her no knowledge of either the *actual* number $0 \leq a \leq p$ of interrupts that will occur or of their placements.

Our model allows us to view a cycle-stealing opportunity as a sequence of $a+1$ *episodes* during which workstation A has access to workstation B, punctuated by the a actual interrupts caused by the return of B's owner; each episode, save the last, is terminated by an interrupt. We emphasize that A's owner knows nothing about the durations $L_1, L_2, \ldots, L_{a+1}$ of the $a+1$ episodes, except that they sum to U.

2.2 Cycle-Stealing Schedules and Their Work Production

Episode-schedules. In order to decrease vulnerability to interrupts that kill work in progress on B, the owner of A partitions each episode into *periods*, each of which begins with A sending work to B and terminates with B returning the results of that work. Since A's discretionary power thus resides solely in deciding how much work to send in each period, and since task-lengths are known perfectly, we view an *episode-schedule* simply as a sequence of period-lengths: an m-period schedule for an episode of length $\leq L$ (the current residual lifespan) thus has the form[3] $\mathcal{S} = t_1, t_2, \ldots, t_m$, where: $m \geq 1$, each $t_i > 0$, and

$$t_1 + t_2 + \cdots + t_m = L. \tag{2.1}$$

[3]We simplify subsequent notation by using 1-origin indexing for the periods of episode-schedules, in contrast to the 0-origin indexing of [3, 11].

The upper bound on the length of the initial episode is U; if the current upper bound is L, then an interrupt at time t of the current episode leaves a residual upper bound of $L - t$. The intended interpretation is that at time

$$\tau_k \stackrel{\text{def}}{=} \begin{cases} T_0 \stackrel{\text{def}}{=} 0 & \text{if } k = 1 \\ T_{k-1} \stackrel{\text{def}}{=} t_1 + t_2 + \cdots + t_{k-1} & \text{if } k > 1 \end{cases}$$

the kth period of (the episode scheduled by) $\mathcal{S}$ begins: workstation A supplies workstation B with a job containing[4] $t_k \ominus c$ units of work. This quantity is chosen so that t_k time units are sufficient for A to send the work to B, and for B to both perform the work and return the results of the work to A.

Say that the residual lifespan at the beginning of a given episode is L time units. If workstation B *is not interrupted* during the kth period of the episode, i.e., by time $T_k = \tau_k + t_k$, then the amount of work done so far during this episode is augmented by $t_k \ominus c$; if B *is interrupted* during the kth period, say at time[5] $t \in [\tau_k,\ T_k)$, then the episode terminates with the total amount of work

$$\mathcal{W}(\mathcal{S}) \stackrel{\text{def}}{=} (t_1 \ominus c) + (t_2 \ominus c) + \cdots + (t_{k-1} \ominus c) \tag{2.2}$$

and with the new residual lifespan $L - t$. This reckoning reflects both the episode's termination and the loss of work from the interrupted period.

It is thus clear that creating an episode-schedule $\mathcal{S} = t_1, t_2, \ldots, t_m$ entails choosing both $\mathcal{S}$'s number of periods (m) and its period-lengths (the t_i). In our framework, all of these quantities are determined completely by the current residual lifespan L and the adversary's remaining allocation of interrupts

$$p = (\text{the initial allocation of interrupts}) - (\text{the number of preceding episodes}).$$

Consistent with this framework, we adopt the following notation.[6]

- The number m of periods in $\mathcal{S}$ is denoted $m^{(p)}[L]$.
- Each period-length t_i is denoted $t_i^{(p)}[L]$. By inheritance, the same is true of each cumulative period-length, $T_i = T_i^{(p)}[L]$.

To enhance legibility, we often use "abbreviated" notation, wherein one or both of the parameters p, L is omitted when it is either irrelevant or clear from context.

Opportunity-schedules. Since the lengths of episodes—which are dictated by the adversary's placement of interrupts—are not known *a priori*, the owner of workstation A has the choice of scheduling a given cycle-stealing opportunity either *adaptively* or *non-adaptively*.

[4] The operator "$\ominus$" denotes *positive subtraction* and is defined by: $x \ominus y \stackrel{\text{def}}{=} \max(0, x - y)$.
[5] As usual, the assertion "$a \in [b,\ c)$" means "$b \le a < c$".
[6] Throughout, square brackets are used only to enclose parameters that represent residual lifespans.

When proceeding *non-adaptively*, the owner of A crafts a single episode-schedule $\mathcal{S} = t_1, t_2, \ldots, t_m$. If a given period, say the ith, is interrupted, then upon regaining control of workstation B, the owner of A employs the "tail" $t_{i+1}, t_{i+2}, \ldots, t_m$ of schedule $\mathcal{S}$ for the remainder of the opportunity. The only exception to this "oblivious" behavior is that after the pth interrupt, the owner of A schedules the remainder of the opportunity as one long period. The *work* $\mathcal{W}(\mathcal{S})$ achieved by the non-adaptive opportunity schedule $\mathcal{S}$ is calculated as follows. Say that the periods that the adversary interrupts comprise the set $I = \{i_1, i_2, \ldots, i_p\}$ and that each period i_j is interrupted at its last instant (so that the entire period is "lost"). Then

$$\mathcal{W}(\mathcal{S}) \; = \; \sum_{k \notin I} (t_k \ominus c) + ((U - T_{i_p}) \ominus c). \tag{2.3}$$

The last term in (2.3) represents the last, "long", period which is invoked after p interrupts have occurred.

When proceeding *adaptively*, the owner of A specifies a schedule for episode $i+1$ only after episode i has been interrupted—by which time A knows how much of the usable lifespan remains. In this case, an adaptive opportunity-schedule Σ is a sequence of sequentially chosen multi-parameterized episode-schedules:

$$\Sigma \; = \; \mathcal{S}^{(p)}[U], \; \mathcal{S}^{(p-1)}[U - L_1], \; \mathcal{S}^{(p-2)}[U - L_1 - L_2], \; \ldots, \; \mathcal{S}^{(p-a)}\left[U - \sum_{i=1}^{a} L_i\right], \tag{2.4}$$

where $L_1, L_2, \ldots, L_a$ are the respective lengths of the a interrupted episodes.

The work achieved under the opportunity-schedule Σ of (2.4) during the lifespan U is the sum of the work achieved by Σ's constituent episode-schedules:

$$\mathcal{W}(\Sigma) \; = \; \sum_{i=0}^{a} \mathcal{W}\left(\mathcal{S}^{(p-i)}\left[U - \sum_{j=1}^{i} L_j\right]\right). \tag{2.5}$$

In constructing and evaluating our adaptive opportunity-schedules, it is useful to have a notation for the maximum amount of work achievable by any adaptive cycle-stealing schedule in an opportunity with (residual) lifespan L and number p of potential interrupts. We denote this quantity by $\mathcal{W}^{(p)}[L]$.

3 Guidelines for Crafting (Near-Optimal) Schedules

This section is devoted to crafting guidelines for producing optimal non-adaptive schedules (Section 3.1) and nearly optimal adaptive ones (Section 3.2).

3.1 Guidelines for Optimal Non-Adaptive Schedules

When the lifespan U is "small" relative to the number p of potential interrupts, one would not do badly to schedule the lifespan as a single episode consisting of $p+1$ equal-length periods (to within rounding). Clearly this approach guarantees us at least $\lfloor U/(p+1) \rfloor$ units of work. When U is even modestly large, though, one can do better by partitioning the lifespan U evenly into roughly $\sqrt{U}$ periods (to within rounding). One shows easily that this approach achieves at least $U - (p+c)\sqrt{U} + pc$ units of work. A bit further analysis improves this last schedule to the following, optimal non-adaptive schedule.

Schedule specification. The p-interrupt non-adaptive schedule

$$\mathcal{S}_{\text{NA}}^{(p)}[U] \;=\; t_1^{(p)}[U],\; t_2^{(p)}[U], \ldots,\; t_m^{(p)}[U]$$

is specified as follows.

Number of periods: $m^{(p)}[U] \;=\; \lfloor \sqrt{pU/c} \rfloor$.

Period-lengths: Each $t_i^{(p)}[U] \;=\; \sqrt{cU/p}$, to within rounding, with up-rounded period-lengths having lower indices.

Analysis. Clearly the best strategy for the adversary is to kill the *last* p periods of schedule $\mathcal{S}_{\text{NA}}^{(p)}[U]$ (at their last instant), for this maximizes the effect of the communication overhead in diminishing work production. Under this strategy,

$$\mathcal{W}\left(\mathcal{S}_{\text{NA}}^{(p)}\right) \;=\; (m^{(p)} - p)\left(\frac{U}{m^{(p)}} - c\right) \;=\; U - \sqrt{2pcU} + pc + O(1).$$

Elementary calculus shows that this strategy cannot be improved.

3.2 Guidelines for Nearly Optimal Adaptive Schedules

In this section, we present and begin to analyze the adaptive opportunity-schedule that is a major contribution of our study. The development in Sections 4 and 5 is needed to complete the analysis and to establish the near-optimality of this schedule in guaranteed work production. The opportunity-schedule $\Sigma_{\text{A}}^{(p)}[U]$ is obtained by adaptive invocation of the following sequence of episode-schedules:

$$\Sigma_{\text{A}}^{(p)}[U] \;=\; \mathcal{S}_{\text{A}}^{(p)}[U],\; \mathcal{S}_{\text{A}}^{(p-1)}[U], \ldots,\; \mathcal{S}_{\text{A}}^{(0)}[U].$$

Schedule specification. The p-interrupt episode-schedule

$$\mathcal{S}_{\text{A}}^{(p)}[U] \;=\; t_1^{(p)}[U],\; t_2^{(p)}[U], \ldots,\; t_m^{(p)}[U]$$

is specified as follows. For $p = 0$, schedule $\mathcal{S}_{\text{A}}^{(0)}[U]$ has one period, of length U. For $p > 0$:

Number of periods: $m^{(p)}[U] \;=\; \lfloor 2^{p-1/2}\sqrt{U/c}\rfloor + p2^{2p-1}$.

Period-lengths: Let $\ell_p \stackrel{\text{def}}{=} \lceil 2p/3 \rceil$.

- For each $k \in \{m^{(p)} - \ell_p + 1, \ldots, m^{(p)}\}$:
  $t_k^{(p)}[U] \;=\; \frac{3}{2}c$.
- For $k = m^{(p)} - \ell_p$:
  $t_k^{(p)}[U] \;=\; \left(p - (2 - 2^{2-p})\sqrt{2p} + 1/2\right) c$.
- For each $k \in \{1, 2, \ldots, m^{(p)} - \ell_p - 1\}$:
  $t_k^{(p)}[U] \;=\; t_{k+1}^{(p)}[U] + 4^{1-p}c$.

We begin our analysis of schedule $\mathcal{S}_{\mathrm{A}}^{(p)}[U]$ by verifying that the indicated specification is sound, in the sense that the specified period-lengths are consistent with the specified number of periods. We do this by invoking equation (2.1) and our specified period-lengths to note that

$$\begin{aligned} U \;&\geq\; \sum_{i=m^{(p)}-\ell_p+1}^{m^{(p)}} t_i^{(p)} \;+\; (m^{(p)} - \ell_p)t_{m^{(p)}-\ell_p}^{(p)} \;+\; \binom{m^{(p)} - \ell_p}{2} 4^{1-p}c \\ &\geq\; pc \;+\; (m^{(p)} - 2p/3)\left(p - (2 - 2^{2-p})\sqrt{2p} + 1/2\right) c \;+\; \binom{m^{(p)} - 2p/3}{2} 4^{1-p}c \end{aligned}$$

Letting $n \;=\; m^{(p)} - 2p/3$, we therefore have

$$2^{2p-1}U/c \;\geq\; n^2 \;-\; 2^{2p-1}(p - 2\sqrt{2p} + 1/2)n.$$

It follows that

$$m^{(p)} \;\leq\; 2^{p-1/2}\sqrt{U/c} + p2^{2p-1},$$

as was claimed.

The remainder of our analysis of the opportunity-schedule $\Sigma_{\mathrm{A}}^{(p)}[U]$ via its constituent episode-schedules $\{\mathcal{S}_{\mathrm{A}}^{(p)}[U] \mid p = 1, 2, \ldots\}$ must be deferred until we better understand the structure of optimal schedules. We turn now to the study of such structure.

4 The Theoretical Underpinnings of Our Adaptive Guidelines

Our formal model leads us to view the cycle-stealing process as the following game against a "malicious adversary" who seeks to use the p available interrupts to minimize the work production of the given cycle-stealing opportunity, even as we seek to maximize this production. The first move is ours. Based on the current (residual) usable lifespan L and bound p on the number of possible interrupts, we invoke episode-schedule $\mathcal{S}^{(p)}[L]$. As long as the adversary has not yet used all of his/her allocated interrupts (i.e., as long as $p > 0$), s/he will decide either to let the current episode play out without an interrupt or to interrupt one of this schedule's periods, thereby nullifying some of our usable lifespan. If the adversary does interrupt us, say at time t of the current episode, then when we regain control of workstation B, we invoke episode-schedule $\mathcal{S}^{(p-1)}[L-t]$. The game continues until $p = 0$, at which point the episode plays out to the end of the residual lifespan without further participation by the adversary.

Our approach to this "game" is to bootstrap our way to optimal cycle-stealing schedules. We always assume, when constructing a schedule for an opportunity having p potential interrupts, that, for each (residual) lifespan L, we inductively have access to a $(p-1)$-interrupt schedule which accomplishes work $\mathcal{W}^{(p-1)}[L]$. We shall see in Proposition 4.4 that we readily do have access to $\mathcal{W}^{(0)}[L]$, hence are prepared to bootstrap from the "end game", wherein $p = 0$.

This section is devoted to deriving guidelines for constructing episode-schedules that minimize the effects of maliciously placed interrupts. In Section 4.1, we establish some basic properties of our model, which will be useful in crafting "good" episode-schedules. In Section 4.2, we derive the abstract scheduling guidelines that underlie schedule $\Sigma_{\mathrm{A}}^{(p)}[U]$.

4.1 Observations that Underlie Our Guidelines

This section is devoted to uncovering properties of our model that determine both the adversary's strategy for interrupting the episodes mandated by our schedules and our responses to that strategy.

4.1.1 Some Naive Observations about Good Schedules

We begin with four simple, yet useful, results about the optimal work-functions $\mathcal{W}^{(p)}[U]$. The first two results establish the monotonicity of the functions in both parameters p and U: work production can only *increase* with a longer residual lifespan and can only *decrease* if the adversary gets more potential interrupts.

Proposition 4.1 *For all p, the function $\mathcal{W}^{(p)}[U]$ is nondecreasing with increasing U.*

Proof Sketch. Any schedule for lifespan U can be converted to an (at least) equally productive one for lifespan $U' > U$ by merely appending a new period of length $U' - U$ at the end. ■

Proposition 4.2 *For all U, the function $\mathcal{W}^{(p)}[U]$ is nonincreasing with increasing p.*

Proof Sketch. If $\mathcal{W}^{(p)}[U] < \mathcal{W}^{(p+r)}[U]$ for some $p \geq 0$ and $r > 0$, then an adversary who had access to $p + r$ potential interrupts would use only p of them—maliciously placed, in order to minimize our work production. ■

We next remark on two simple "boundary cases" of our scheduling problem, which are useful as the bases for our bootstrapping strategy for crafting schedules. We note first that the adversary can effectively nullify any sufficiently short lifespan.

Proposition 4.3 *If the lifespan* $U \leq (p+1)c$*, then* $\mathcal{W}^{(p)}[U] = 0$.

Proof. No matter how one partitions such a short lifespan into periods, no more than p periods can have length $> c$. The adversary can nullify each of these productive periods via a maliciously placed interrupt. This nullifies the entire opportunity, because the communication overhead c prevents us from accomplishing any work from the shorter periods. ■

Finally, we note, without proof, the triviality of scheduling the "end-game", i.e., the case $p = 0$, wherein the adversary has exhausted his/her allocation of interrupts.

Proposition 4.4 *The unique optimal schedule for the case* $p = 0$ *is the* 1*-period schedule* $\mathcal{S}^{(0)}_{\rm OPT}[U] = U$ *which achieves*

$$\mathcal{W}^{(0)}[U] \ = \ U - c \tag{4.1}$$

units of work.

4.1.2 Some Sophisticated Observations about Good Schedules

Our analysis of the cycle-stealing game requires us to consider what the adversary's various options are regarding an episode—whether to interrupt it, and where—and to how address each of them. At first blush, it appears that, when the adversary has access to $p > 0$ interrupts, these options are $m^{(p)}[U]+1$ in number: not to interrupt the episode (thereby letting the game "play out"), or to interrupt period $k \in \{1, 2, \ldots, m^{(p)}[U]\}$ of the episode. As we develop our understanding of the game in this section, we shall see that the adversary actually has fewer viable options. We detail the adversary's strategy in a sequence of "Observations".

Our first step in analyzing the cycle-stealing game is to establish a result that has three significant consequences. First, it materially narrows our search for optimal schedules. Second, it allows us to use ordinary subtraction, rather than positive subtraction, when discussing the potential work production from each of an episode's periods, save the last. Finally, it leads directly to two significant observations about the adversary's strategy during the game. The result is an analog of one proved in [3] for the expected-output single-episode submodel; it shows that we lose no generality by restricting attention to cycle-stealing schedules that are *productive*, in the sense of having all period-lengths, save perhaps the last in each episode-schedule, exceed c.

Theorem 4.1 *Any opportunity-schedule* Σ *can be replaced by a productive opportunity-schedule* $\widehat{\Sigma}$ *such that* $\mathcal{W}(\widehat{\Sigma}) \geq \mathcal{W}(\Sigma)$.

Proof. Say that the opportunity-schedule Σ contains one or more nonproductive episode-schedules $\mathcal{S}^{(k)}[L]$, where $k \geq 1$. (When $k = 0$, the optimal schedule has only one period, so productivity is not an issue.) Say, in particular that the kth constituent episode-schedule $\mathcal{S}^{(k)}[L]$ of Σ contains a nonterminal period, say the ith, of length $t_i \leq c$. Consider the schedule $\widetilde{\Sigma}$ which is identical to Σ, except that periods i and $i+1$ of its kth constituent episode-schedule $\widetilde{\mathcal{S}}^{(k)}[L]$ are combined:[7]

$$\begin{aligned} \mathcal{S}^{(k)}[L] &= t_1, t_2, \ldots, t_{i-1}, t_i, t_{i+1}, t_{i+2}, \ldots, t_m; \\ \widetilde{\mathcal{S}}^{(k)}[L] &= t_1, t_2, \ldots, t_{i-1}, t_i + t_{i+1}, t_{i+2}, \ldots, t_m. \end{aligned}$$

We claim that $\mathcal{W}(\widetilde{\Sigma}) \geq \mathcal{W}(\Sigma)$. The verification is a case analysis based on when, if at all, episode k is interrupted. First, we compare $\mathcal{W}(\widetilde{\mathcal{S}}^{(k)}[L])$ with $\mathcal{W}(\mathcal{S}^{(k)}[L])$.

1. If episode k is not interrupted, or if it is interrupted at time $t \geq T_{i+1}$, then, since $(t_i + t_{i+1}) \ominus c \;\geq\; (t_i \ominus c) + (t_{i+1} \ominus c)$, we easily have

   $\mathcal{W}(\widetilde{\mathcal{S}}^{(k)}[L]) \geq \mathcal{W}(\mathcal{S}^{(k)}[L])$.

2. If episode k is interrupted at time $t \in [T_{j-1},\ T_j)$ for some $j \leq i$, then

   $$\mathcal{W}(\widetilde{\mathcal{S}}^{(k)}[L]) \;=\; \mathcal{W}(\mathcal{S}^{(k)}[L]) \;=\; (t_1 \ominus c) + (t_2 \ominus c) + \cdots + (t_{j-1} \ominus c).$$

3. Finally, if episode k is interrupted at time $t \in [T_i,\ T_{i+1})$, then

   $$\mathcal{W}(\widetilde{\mathcal{S}}^{(k)}[L]) \;=\; \mathcal{W}(\mathcal{S}^{(k)}[L]) \;=\; (t_1 \ominus c) + (t_2 \ominus c) + \cdots + (t_{i-1} \ominus c),$$

   because $t_i \leq c$.

Next, we note that $\widetilde{\mathcal{S}}^{(k)}[L]$ and $\mathcal{S}^{(k)}[L]$ always leave the same residual lifespan: 0 when episode k is not interrupted and $L - t$ otherwise. We thus have $\mathcal{W}(\widetilde{\Sigma}) \geq \mathcal{W}(\Sigma)$.

Now, episode-schedule $\widetilde{\mathcal{S}}^{(k)}[L]$ has one fewer violation of period-productivity than $\mathcal{S}^{(k)}[L]$. We can, therefore, continue eliminating such violations until we obtain a productive episode-schedule $\overline{\mathcal{S}}^{(k)}[L]$ which has $\mathcal{W}(\overline{\mathcal{S}}^{(k)}[L]) \geq \mathcal{W}(\mathcal{S}^{(k)}[L])$ and which leaves the same residual lifespan as does $\mathcal{S}^{(k)}[L]$. At that point, we shall have eliminated one violation of episode-productivity from schedule Σ. By continuing in this way, we eventually produce the desired schedule $\widehat{\Sigma}$. ∎

Theorem 4.1 assures us that we lose no generality by focusing henceforth only on episode-schedules $\mathcal{S} = t_1^{(p)}, t_2^{(p)}, \ldots, t_m^{(p)}$ for which each $t_k^{(p)} > c$, save perhaps $t_m^{(p)}$. Having thus narrowed

[7] We use abbreviated notation throughout the proof, to enhance legibility.

our focus, our detailed analysis splits logically into two parallel threads, one focusing on episode-schedules that are *fully productive*, in the sense of having *all* period-lengths $> c$, the other focusing on episode-schedules whose terminal period-length is $\leq c$. These two threads yield to identical reasoning but distinct calculations. In order to conserve space, *we focus henceforth on fully productive episode-schedules*, leaving the complementary case to the interested reader.

Interrupted Period	Interruption Time	Episode Work-Output	Residual Lifespan
No interrupt	N/A	$U - mc \ = \ T_m^{(p)} - mc$	0
1	$t \in [0, \ T_1^{(p)})$	0	$U - t$
k	$t \in [T_{k-1}^{(p)}, \ T_k^{(p)})$	$T_{k-1}^{(p)} - (k-1)c$	$U - t$
m	$t \in [T_{m-1}^{(p)}, \ U)$	$T_{m-1}^{(p)} - (m-1)c$	0

Table 1: *The immediate consequences of the adversary's options*

In the light of Theorem 4.1 and our focus on fully productive episode-schedules, expression (2.2) gives us access to the immediate consequences of each of the adversary's $m^{(p)}+1$ apparent options; these are enumerated in Table 1 (using abbreviated notation). The table combines with Proposition 4.1 to yield our first observation about the adversary's preferred strategy during the "game": the adversary has no incentive to interrupt a period anywhere but at its end. Interrupting a period anywhere else involves the same expenditure—one of the p available interrupts—but achieves less (from the adversary's perspective), since it leaves a larger residual lifespan.

Observation 4.1 *The adversary will always strive for maximum "mileage" from each interrupt by interrupting a period at its last instant, thereby nullifying a full* $t_k^{(p)}[U]$ *time units from our usable lifespan.*

In the light of our bootstrapping strategy, Observation 4.1 allows us to extrapolate from the local information in Table 1, to obtain the global information in Table 2.

The information in Table 2 combines with Theorem 4.1 to yield our second observation about the adversary's strategy. To wit, the adversary can strictly decrease the work production of a

Interrupted Period	Opportunity Work Production
No interrupt	$U - mc \; = \; T_m^{(p)} - mc$
1	$\mathcal{W}^{(p-1)}[U - T_1^{(p)}]$
k	$T_{k-1}^{(p)} - (k-1)c + \mathcal{W}^{(p-1)}[U - T_k^{(p)}]$
m	$T_{m-1}^{(p)} - (m-1)c$

Table 2: *The long-term consequences of the adversary's options*

fully productive episode-schedule by interrupting the schedule's last period. If this interruption occurs within c time units of the end of the episode, it actually decreases the work production of the entire opportunity, since it leaves no usable residual lifespan.[8] We infer the following.

Observation 4.2 *The adversary will always interrupt every possible episode of a cycle-stealing opportunity as long as $p > 0$ and the residual lifespan $U > c$.*

The proviso "$p > 0$" in Observation 4.2 means that the adversary still has available interrupts; the proviso "$U > c$" means that the episode can achieve actual work, hence is worth interrupting.

Observation 4.2 tells us that not all of the $m^{(p)}[U] + 1$ options we ascribed to the adversary are actually viable; specifically, the no-interrupt option would not achieve the adversary's goal of minimizing our work production. By similar reasoning, one can show that the adversary will never interrupt a long episode too near its end, in the following sense.

Observation 4.3 *When the adversary has $p \geq 1$ potential interrupts left, s/he will always interrupt an episode of lifespan $U > (p+1)c$ during a period that begins at some time $t < U - pc$.*

[8]This fact is not a consequence of our insisting on fully productive episode-schedules: by Theorem 4.1, the next-to-last period of any productive episode-schedule has length $> c$. If the last period is short, then the adversary can decrease the work production of the opportunity by interrupting this next-to-last period (again, near its end).

Observation 4.3 follows from Proposition 4.3's assurance that the adversary's remaining $p-1$ interrupts will suffice to nullify the last pc time units of the residual lifespan.

We now establish the computational significance of our Observations, via the following notion. An episode-schedule $\mathcal{S}^{(p)}[U] = t_1^{(p)}, t_2^{(p)}, \ldots, t_m^{(p)}$ is *r-immune*, where $r \in \{0, 1, \ldots, m-1\}$, if the adversary will never interrupt a period whose index *exceeds* $m-r$ (although s/he may interrupt period $m-r$). In particular, an episode-schedule is 0-immune if any period can be interrupted. We now show that the lengths of the $r+1$ highest-index periods of an optimal r-immune episode-schedule can be constrained within a narrow range.

Theorem 4.2 *For any r-immune episode-schedule $\mathcal{S}^{(p)}[U]$, there exists an r-immune episode-schedule $\widehat{\mathcal{S}}^{(p)}$ each of whose period-lengths, $\hat{t}_{m-r}^{(p)}[U], \hat{t}_{m-r+1}^{(p)}[U], \ldots, \hat{t}_m^{(p)}[U]$, lies in the range $(c,\ 2c]$, such that $\mathcal{W}(\widehat{\mathcal{S}}^{(p)}[U]) \geq \mathcal{W}(\mathcal{S}^{(p)}[U])$.*

Proof. Since we consider only fully productive episode-schedules, we need focus only the upper bound on the period-lengths. To this end, say that the r-immune episode-schedule $\mathcal{S}^{(p)}[U] = t_1^{(p)}, t_2^{(p)}, \ldots, t_m^{(p)}$ has $t_\ell^{(p)} > 2c$ for some $\ell \geq m-r$. Define the episode-schedule $\widehat{\mathcal{S}}^{(p)}$ to be identical to $\mathcal{S}^{(p)}$, except for its $(m-r)$th period, which is split into two equal-length periods; using abbreviated notation:

$$\begin{aligned}
\mathcal{S}^{(p)}[U] &= t_1, t_2, \ldots, t_{\ell-1}, t_\ell, t_{\ell+1}, \ldots, t_m; \\
\widehat{\mathcal{S}}^{(p)}[U] &= t_1, t_2, \ldots, t_{\ell-1}, \tfrac{1}{2}t_\ell, \tfrac{1}{2}t_\ell, t_{\ell+1}, \ldots, t_m.
\end{aligned}$$

We claim that $\mathcal{W}(\widehat{\mathcal{S}}^{(p)}[U]) \geq \mathcal{W}(\mathcal{S}^{(p)}[U])$. The verification is a case analysis based on when (i.e., during which period) the given episode is interrupted. (The fact that the episode will be interrupted follows by Observation 4.2, in the light of the fact that schedule $\widehat{\mathcal{S}}^{(p)}[U]$ is fully productive whenever schedule $\mathcal{S}^{(p)}[U]$ is.)

1. If the episode is interrupted at time $t \in [T_{j-1},\ T_j)$ for some $j \leq \ell - 1$, then
$$\mathcal{W}(\widehat{\mathcal{S}}^{(p)}[U]) \ = \ \mathcal{W}(\mathcal{S}^{(p)}[U]) \ = \ T_{j-1} - (j-1)c.$$
2. If the episode is interrupted at time $t \in [T_{\ell-1},\ T_{\ell-1} + \frac{1}{2}t_\ell)$, then
$$\mathcal{W}(\widehat{\mathcal{S}}^{(p)}[U]) \ = \ \mathcal{W}(\mathcal{S}^{(p)}[U]) \ = \ T_{\ell-1} - (\ell-1)c.$$
3. If the episode is interrupted at time $t \in [T_{\ell-1} + \frac{1}{2}t_\ell,\ T_\ell)$, then
$$\mathcal{W}(\widehat{\mathcal{S}}^{(p)}[U]) \ = \ T_{\ell-1} + \tfrac{1}{2}t_\ell - \ell c \ > \ \mathcal{W}(\mathcal{S}^{(p)}[U]) \ = \ T_{\ell-1} - (\ell-1)c.$$

Additionally, $\widehat{\mathcal{S}}^{(p)}[U]$ and $\mathcal{S}^{(p)}[U]$ always leave the same residual lifespan, $U-t$, hence allow the same amount of residual work production. By definition of immunity, the three enumerated cases exhaust the possibilities, whence the theorem. ∎

A concrete example of the interplay between Observation 4.3 and Theorem 4.2 is that, when the adversary has $p \geq 2$ potential interrupts left, s/he will never interrupt the last period of an episode, since it can be nullified later using his/her last interrupt.

Note that the reasoning underlying Theorem 4.2 does not apply to any interruptable period of an episode but the last. Partitioning a long "interior" period (as in the proof) could decrease the worst-case productivity of schedule $\widehat{\mathcal{S}}^{(p)}[U]$ (by c) if the adversary were to interrupt the episode *after* the partitioned long period.

4.2 Characteristics of the Optimal Adaptive Episode-Schedules

Once we understand the operative options of the adversary, we formulate a plan to counteract the fact that s/he will choose the option that minimizes our overall work production, as given in Table 2. Our counterstrategy is to craft episode-schedule $\mathcal{S}^{(p)}[U]$ to *equalize* the impacts of all potential interruptions, thereby maximizing our minimum work production (under the adversary's potential actions). We thus arrive at the following result, whose constituent equalities—in (4.2)—are justified by the preceding discussion and explained by their accompanying annotations.

Lemma 4.1 *Under the optimal episode-schedule $\mathcal{S}^{(p)}_{\mathrm{OPT}}[U] = t_1^{(p)}, t_2^{(p)}, \ldots, t_m^{(p)}$ for a cycle-stealing opportunity with $\leq p$ interrupts and with usable lifespan $U > (p+1)c$, the optimal work production of the opportunity satisfies the following equalities.*

If $p = 0$, then $\mathcal{W}^{(p)}[U] = U - c$.

If $p > 0$, then, letting ℓ_p be the smallest period-index k for which $U - T_{k-1}^{(p)} > pc$:[9]

$$\begin{array}{rcll} \mathcal{W}^{(p)}[U] & = & \mathcal{W}^{(p-1)}[U - T_1^{(p)}] & \text{-period } 1 \text{ interrupted} \\ & = & T_1^{(p)} - c + \mathcal{W}^{(p-1)}[U - T_2^{(p)}] & \text{-period } 2 \text{ interrupted} \\ & & \vdots & \\ & = & T_{\ell_p-2}^{(p)} - c + \mathcal{W}^{(p-1)}[U - T_{\ell_p-1}^{(p)}] & \text{-period } \ell_p - 1 \text{ interrupted} \\ & = & T_{\ell_p-1}^{(p)} - (\ell_p - 1)c & \text{-period } \ell_p \text{ interrupted} \end{array} \quad (4.2)$$

For the sake of concreteness, note that $\ell_p = m^{(p)}[U]$ when $p = 1$ (by our assumption of full productivity) and $\ell_p = m^{(p)}[U] - 1$ when $p = 2$ (by Theorem 4.2). One cannot predict the value of ℓ_p *a priori* for larger p.

Our work-equalizing strategy for counteracting the adversary translates into the partial specification of the period-lengths of the optimal episode-schedule in the following theorem.

[9] To connect system (4.2) to the preceding discussion, recall that $U - T_k^{(p)} = t_{k+1}^{(p)} + \cdots + t_m^{(p)}$.

Theorem 4.3 *The period-lengths of the optimal episode-schedule $\mathcal{S}^{(p)}_{\mathrm{OPT}}[U]$ for a cycle-stealing opportunity with usable lifespan U and $\leq p$ interrupts satisfy the following system of equalities.*

If $p = 0$, then $m^{(p)}[U] = 1$, and $t^{(p)}_1[U] = U$.

If $p > 0$, then, letting ℓ_p be the smallest period-index k for which $U - T^{(p)}_{k-1} > pc$:

$$t^{(p)}_k \;=\; \begin{cases} c + \mathcal{W}^{(p-1)}[U - T^{(p)}_k] - \mathcal{W}^{(p-1)}[U - T^{(p)}_{k+1}] & \text{for } 1 \leq k \leq \ell_p - 2 \\ c + \mathcal{W}^{(p-1)}[U - T^{(p)}_{\ell_p - 1}] & \text{for } k = \ell_p - 1 \\ c + \alpha \;\in\; (c,\; 2c] & \text{for } \ell_p \leq k \leq m. \end{cases} \tag{4.3}$$

Proof Sketch. The bounds on period-lengths $t^{(p)}_{\ell_p}[U], \ldots, t^{(p)}_m[U]$ in (4.3) follow from Theorem 4.2; the specification of all other period-lengths result from equating consecutive pairs of work-expressions in (4.2). ■

One can often gain computational advantage from the "telescoping" property of the period-length expressions in (4.3); e.g., for all $i < k < \ell_p$:

$$t^{(p)}_i + t^{(p)}_{i+1} + \cdots + t^{(p)}_k \;=\; (k - i + 1)c + \mathcal{W}^{(p-1)}[U - T^{(p)}_i] - \mathcal{W}^{(p-1)}[U - T^{(p)}_{k+1}].$$

(Recall that the last term vanishes when $k = \ell_p - 1$.)

5 From Underpinnings to Guidelines

This section is devoted to evaluating the adaptive opportunity-schedule $\Sigma^{(p)}_{\mathrm{A}}[U]$ and its constituent episode-schedules $\mathcal{S}^{(p)}_{\mathrm{A}}[U]$, in two senses. First, in Section 5.1, we evaluate $\Sigma^{(p)}_{\mathrm{A}}[U]$ in an absolute sense, by estimating its guaranteed work production. Next, in Section 5.2, we evaluate the schedule in a relative sense, using the abstract guidelines implicit in Section 4—most notably in Theorems 4.2 and 4.3—as our baseline. We construct the actual optimal episode-schedule $\mathcal{S}^{(1)}_{\mathrm{OPT}}[U]$ for the case $p = 1$ and compare the optimal work production $\mathcal{W}^{(1)}[U]$ with $\mathcal{W}(\Sigma^{(1)}_{\mathrm{A}}[U])$ and, more generally, with $\mathcal{W}(\Sigma^{(p)}_{\mathrm{A}}[U])$. We thereby discover that our approximate schedules $\Sigma^{(p)}_{\mathrm{A}}[U]$ have work production that deviates from optimality by only a low-order additive term. We close our study in Section 5.3 by illustrating how the (computationally cumbersome) structure of optimal episode-schedules—as exposed in Section 4—suggests the detailed structure of the schedules $\mathcal{S}^{(p)}_{\mathrm{A}}[U]$.

5.1 The Guaranteed Work Production of Schedule $\Sigma^{(p)}_{\mathrm{A}}[U]$

In this section we use the analyses of Section 4 as tools for estimating $\mathcal{W}(\Sigma^{(p)}_{\mathrm{A}}[U])$.

Theorem 5.1 *For all* $p \geq 0$,

$$\mathcal{W}(\Sigma_{\mathrm{A}}^{(p)}[U]) \;\geq\; U - (2 - 2^{1-p})\sqrt{2cU} - O(U^{1/4} + pc). \tag{5.1}$$

Proof. We proceed by induction on p.

The case $p = 0$. Since $\mathcal{W}(\mathcal{S}_{\mathrm{A}}^{(0)}[U]) = U - c$ by fiat, that schedule's work production automatically complies with (5.1).

The case $p = 1$. In this case, we can use the same reasoning that leads to Lemma 4.1 to prove that

$$\mathcal{W}(\Sigma_{\mathrm{A}}^{(1)}[U]) \;\geq\; U - t_m^{(1)} - (m^{(1)} - 1)c. \tag{5.2}$$

Using inequality (5.2) and the development in Section 3.2, we find that

$$\mathcal{W}(\Sigma_{\mathrm{A}}^{(1)}[U - t_1^{(p)}]) \;\geq\; U - \sqrt{2cU} - \frac{5}{2}c. \tag{5.3}$$

We thus have compliance with (5.1).

The case $p > 1$. Let us now assume, for the sake of induction, that (5.1) holds for all $p \leq q$, and let us consider the case $p = q + 1$. In order to deal with general p it is easier now to use a different insight from Lemma 4.1. Specifically, using the same reasoning that leads to that lemma, we can prove that, for general p,

$$\mathcal{W}(\Sigma_{\mathrm{A}}^{(p)}[U]) \;\geq\; \mathcal{W}(\Sigma_{\mathrm{A}}^{(p-1)}[U - t_1^{(p)}]). \tag{5.4}$$

By inequality (5.4) and our inductive hypothesis, we thus have

$$\begin{aligned} \mathcal{W}(\Sigma_{\mathrm{A}}^{(q+1)}[U]) \;&\geq\; \mathcal{W}(\Sigma_{\mathrm{A}}^{(q)}[U - t_1^{(q+1)}]) \\ &\geq\; U - t_1^{(q+1)} - (2 - 2^{1-q})\sqrt{2c(U - t_1^{(q+1)})}. \end{aligned}$$

By the specification of episode-schedule $\mathcal{S}_{\mathrm{A}}^{(q+1)}$,

$$\begin{aligned} t_1^{(q+1)} &= \left(q + 3/2 - (2 - 2^{1-q})\sqrt{2(q+1)}\right)c + \left(2^{q+1/2}\sqrt{U/c} + (q+1)2^{2q+1} - \frac{2}{3}(q+1)\right)4^{-q}c \\ &= 2^{-q}\sqrt{2cU} + O(qc). \end{aligned}$$

Finally, we combine (5.5) and (5.5) to obtain

$$\mathcal{W}(\Sigma_{\mathrm{A}}^{(q+1)}[U]) \;\geq\; U - (2 - 2^{-q})\sqrt{2cU} + O(U^{1/4} + qc).$$

This extends our induction and completes the proof. ■

5.2 Comparing Schedule $\Sigma_{\mathrm{A}}^{(p)}[U]$ against Optimality

In this section, we apply the abstract guidelines of Section 4 to derive the actual optimal episode-schedule $\mathcal{S}_{\mathrm{OPT}}^{(1)}[U]$ for the case $p = 1$.[10] (We have been unable to derive actual optimal schedules for any $p > 1$, because our abstract guidelines become computationally cumbersome even for this case—more about this later.) Importantly, though, even the case $p = 1$ suffices to show that our approximate schedules $\Sigma_{\mathrm{A}}^{(p)}[U]$ have work production that deviates from optimality by only a low-order additive term.

Throughout this subsection, let $m = m^{(1)}[U]$.

Determining the period-lengths of $\mathcal{S}_{\mathrm{OPT}}^{(1)}[U]$. We begin by symbolically determining the relationships among the m period-lengths of the optimal episode-schedule $\mathcal{S}_{\mathrm{OPT}}^{(1)}[U] = t_1^{(1)}[U], t_2^{(1)}[U], \ldots, t_m^{(1)}[U]$. Since the case $p = 1$ is 0-immune, Theorems 4.2 and 4.3 assert that these period-lengths satisfy the following system of equations, for some $\alpha \in (0, 1]$:

$$t_k^{(1)}[U] \;=\; \begin{cases} (1+\alpha)c & \text{for } k = m \\ t_m^{(1)}[U] \;=\; (1+\alpha)c & \text{for } k = m-1 \\ t_{k+1}^{(1)}[U] + c \;=\; (m-k+\alpha)c & \text{for } k \leq m-2 \end{cases} \tag{5.5}$$

Determining α as a function of m. Revisiting equation (2.1) in the light of the system (5.5), we find that

$$U \;=\; (\alpha m + 1)c + \binom{m}{2} c, \tag{5.6}$$

so that

$$\alpha \;=\; \frac{1}{mc}(U - c) - \frac{1}{2}(m-1). \tag{5.7}$$

Determining the optimal m. Examining equation (5.6) in the light of the bounds $0 < \alpha \leq 1$ (from Theorem 4.2), we find that

$$\sqrt{\left(\frac{2U}{c} - \frac{7}{4}\right)} - \frac{1}{2} \;\leq\; m^{(1)}[U] \;<\; \sqrt{\left(\frac{2U}{c} - \frac{7}{4}\right)} + \frac{1}{2}.$$

Since there is only one integer in the indicated range, the preceding inequalities determine the exact optimal value of m uniquely:

$$m^{(1)}[U] \;=\; \left\lceil \sqrt{\left(\frac{2U}{c} - \frac{7}{4}\right)} - \frac{1}{2} \right\rceil. \tag{5.8}$$

[10] A similar derivation appears in [3], using a model which does not insist that schedules be fully productive and using rather different techniques of analysis.

Explicit (approximately) optimal parameters. Equation (5.8) permits us to determine an optimal value for α and, thereby, for the period-lengths $t_k^{(1)}[U]$ and the work-output $\mathcal{W}^{(1)}[U]$. We must settle for approximate values because of the complicated expression for $m^{(1)}[U]$ in (5.8), coupled with the broad range of values of U for which we will have to use this expression. In this latter regard, although it is not difficult to obtain simple good approximations for our parameters when U is very large relative to c, it is much more difficult to obtain simple approximations that are good both when U is large and when it is commensurate with c. (The latter occurs toward the end of a yet-to-be-interrupted episode, when U is a small multiple of $(1+\alpha)c$.) These difficulties notwithstanding, it is instructive to observe the approximate behavior of the relevant parameters (with some indication of the errors induced by the approximations). To this end, we begin by approximating the optimal value of α by instantiating the value $m^{(1)}[U] \approx \sqrt{(2U/c - 7/4)}$ in (5.7). We find thereby that the optimal α is roughly

$$\alpha \;\approx\; \left(\frac{1}{2} - \left(\frac{c}{8U}\right)^{3/2}\right) \;\approx\; \frac{1}{2}.$$

Notably, the final approximation incurs an error of less than $1/20$, even when U (which must exceed c if we are to accomplish any work) is only $c+\epsilon$. When we instantiate the value $\alpha = 1/2$ in system (5.5), we find that the optimal period-lengths of $\mathcal{S}_{\rm OPT}^{(1)}[U]$ are given approximately by

$$t_k^{(1)}[U] \;\approx\; \begin{cases} \dfrac{3}{2}c & \text{for } k \in \{m-1, m\} \\[2ex] \left(\sqrt{\left(2cU - \dfrac{7c^2}{4}\right)} - k + \dfrac{1}{2}\right) c & \text{for } k \le m-2. \end{cases}$$

Finally, we invoke Lemma 4.1 for the case $p = 1$, to find that

$$\mathcal{W}^{(1)}[U] \;=\; U - t_m^{(1)}[U] - (m-1)c \;\approx\; U - \sqrt{\left(2cU - \frac{7c^2}{4}\right)} - \frac{1}{2}c. \tag{5.9}$$

In Table 3, we summarize our closed-form algebraic approximations of the optimal values of the single-interrupt ($p = 1$) parameters, comparing them with the analogous parameters of schedule $\mathcal{S}_{\rm A}^{(1)}[U]$.

As we have already stressed repeatedly, the most important message of this section is the following.

Theorem 5.2 *The guaranteed work production of the opportunity-schedule $\Sigma_{\rm A}^{(p)}[U]$ deviates from optimality by only low-order additive terms.*

Proof Sketch. The result follows by comparing $\mathcal{W}^{(1)}[U]$ with $\mathcal{W}(\Sigma_{\rm A}^{(p)}[U])$. The former quantity is revealed in (5.9), the latter in inequality (5.1), in the light of Proposition 4.2. ■

Parameter	Approximate Value for $\mathcal{S}^{(1)}_{\rm OPT}$	Value for $\mathcal{S}^{(1)}_{\rm A}[U]$
$m^{(1)}[U]$	$\sqrt{2U/c - 7/4}$	$\lfloor \sqrt{2U/c} + 2 \rfloor$
α	$1/2$	N/A
$t^{(1)}_k[U]$ $1 \le k \le m-2$	$\sqrt{2cU} - kc$	$\sqrt{2cU} - (k - 7/2)c$
$t^{(1)}_m[U] \ = \ t^{(1)}_{m-1}[U]$	$3c/2$	$3c/2$
$\mathcal{W}^{(1)}[U]$	$U - \sqrt{2cU} - c/2$	$U - \sqrt{2cU} - O(U^{1/4} + c)$

Table 3: *Parameter values for the case* $p = 1$

5.3 Our Guidelines' Basis in the Cases $p > 1$

In this section, we extrapolate from the form of the optimal episode-schedule $\mathcal{S}^{(1)}_{\rm OPT}[U]$ and from the abstract guidelines of Section 4 to motivate the structure of the episode-schedules $\mathcal{S}^{(p)}_{\rm A}[U]$. Before we do so, however, we wish to indicate briefly the sources of the computational difficulties inherent in our abstract guidelines, which have prevented us from deriving the exact forms of the optimal episode-schedules $\mathcal{S}^{(p)}_{\rm OPT}[U]$ for $p > 1$.

The development in Section 4 forces one to bootstrap from a specification of the optimal p-interrupt schedule to a specification of the optimal $(p+1)$-interrupt schedule. Such bootstrapping is complicated technically by five factors. (1) The nondefinitive period-specifications of Theorem 4.3 force us to calculate period-lengths via bounds rather than equations. (2) Since the residual lifespans after successive interrupts are not known until interrupts actually occur, we must perform our bootstrapping calculations symbolically, rather than numerically. (3) Most of our symbolic computations cannot be performed exactly, because of both nested radicals (when $p > 1$) and nonalgebraic operations such as floors and ceilings. We are forced, therefore, to replace complicated actual expressions by simple approximations—thereby introducing errors that accumulate as we bootstrap. (4) These errors are compounded by our need to ignore—or retrofit—the ineluctable integrality of certain parameters. Most obviously, the number of periods, $m^{(p)}$, must be integra; in certain settings, one might insist also that the period-lengths, $t^{(p)}_k$, be integral. (One finds both integral and nonintegral task-lengths studied in the literature.) (5) The approximation errors are very difficult to estimate, since the estimates one would want to use often depend on the relative sizes of the externally specified parameters U, p, and c.

We turn now to the main topic of this subsection—those aspects of the structure of truly optimal schedules that led us to specify schedule $\mathcal{S}_{\rm A}^{(p)}[U]$ as we have. Throughout this subsection, let $m = m^{(p)}[U]$.

Determining optimal period-lengths. We attempt to determine the relationships among the m period-lengths of the optimal episode-schedule $\mathcal{S}_{\rm OPT}^{(p)}[U] \ = \ t_1^{(p)}[U], t_2^{(p)}[U], \ldots, t_m^{(p)}[U]$. Now, Theorems 4.2 and 4.3 tell us that once we determine the constant of immunity for the particular value of p, we know that the highest index period-lengths $t_{\ell_p}^{(p)}[U], \ldots, t_m^{(p)}[U]$ all have the form

$$t_k^{(2)}[U] \ = \ (1 + \alpha_k)c$$

for some $\alpha_k \in (0, \ 1]$. One gets no further help in determining explicit values for the α_k, since it is only the sum of the α_k, rather than their individual values, that enter into the determination of all lower-index period-lengths. For the sake of determinacy, we have crafted schedule $\mathcal{S}_{\rm A}^{(p)}[U]$ by equating all of the α_k, setting all to the value $1/2$ that is approximately optimal for the case $p = 1$. This has the secondary benefit of explicitly specifying all of the constants of immunity ℓ_p.

In order to invoke Theorem 4.3 to determine all other period-lengths, we must have access to a (computationally tractable) expression for the work production of all schedules for smaller values of p. Again inspired by the case $p = 1$, we have used the working assumption that each $\mathcal{W}^{(p)}[U]$ can be approximated by the expression

$$\mathcal{W}^{(p)}[U] \ = \ U - \delta_p\sqrt{2cU} + \ \text{(low-order terms)}, \tag{5.10}$$

where δ_p is a constant that satisfies $\delta_0 = 0$ and $\delta_1 = 1$. The form we ultimately selected for δ_p, namely, $\delta_p = 2 - 2^{1-p}$ was one that could be perpetuated inductively without compromising good work production (as we saw in Section 5.1).

Thus armed with values for the high-index period-lengths and a tractable approximate expression for the work production of schedules for smaller values of p, we invoked the $m - \ell_p$ instance of system (4.3) to determine a value for $t_{m-\ell_p}^{(p)}[U]$:

$$t_{m-\ell_p}^{(p)} \ = \ c + \mathcal{W}^{(p-1)}[U - T_{m-\ell_p}^{(p)}] \ \approx \ (p+1)c - \delta_{p-1}\sqrt{2pc}.$$

Finally, we proceeded to period-lengths with indices $k < m - \ell_p$ by simplifying the rather cumbersome expression for these period-lengths in (4.3), in the light of expression (5.10). In the following derivation, each unspecified residual lifespan is understood to be U. For each such period-index k:

$$\begin{aligned} t_k^{(p)} &= c + \mathcal{W}^{(p-1)}[U - T_k^{(p)}] - \mathcal{W}^{(p-1)}[U - T_{k+1}^{(p)}] \\ &\approx c + t_{k+1}^{(p)} - \delta_{p-1}\sqrt{2c}\left(\sqrt{U - T_k} - \sqrt{U - T_{k+1}}\right) \\ &= c + t_{k+1}^{(p)} - \delta_{p-1}\sqrt{2c}\left(1 - \sqrt{1 - \frac{t_{k+1}^{(p)}}{U - T_k}}\right)\sqrt{U - T_k} \qquad (5.11) \\ &\approx t_{k+1}^{(p)} + \left(1 - \delta_{p-1}\sqrt{\frac{t_{k+1}^{(p)}}{2c(U - T_k)}}\right) c. \end{aligned}$$

In the last step of this chain, we approximate $\sqrt{1-x}$ (where $x = t_{k+1}^{(p)}/(U - T_k)$) by $1 - \frac{1}{2}x$.

The final approximation in (5.11) indicates that when $p > 1$, each $t_k^{(p)}$ is obtained from $t_{k+1}^{(p)}$ by adding some multiple of c that is less than unity. Even though a careful analysis shows that this multiple decreases with k, we have opted in Section 3.2 for computational simplicity and fixed this multiple at $\gamma_p = 4^{1-p}$.

Having made the indicated decisions, we have completely specified the computationally efficient episode-schedules $\mathcal{S}_{\mathrm{A}}^{(p)}[U]$, hence also the opportunity-schedule $\Sigma_{\mathrm{A}}^{(p)}[U]$, in a way that achieves nearly optimal guaranteed work production.

Acknowledgment. It is a pleasure to acknowledge several helpful conversations with Matt Green and Ramesh Sitaraman.

References

[1] M.J. Atallah, C.L. Black, D.C. Marinescu, H.J. Siegel, T.L. Casavant (1992): Models and algorithms for coscheduling compute-intensive tasks on a network of workstations. *J. Parallel Distr. Comput. 16*, 319-327.

[2] B. Awerbuch, Y. Azar, A. Fiat, F.T. Leighton (1996): Making commitments in the face of uncertainty: how to pick a winner almost every time. *28th ACM Symp. on Theory of Computing*, 519-530.

[3] S.N. Bhatt, F.R.K. Chung, F.T. Leighton, A.L. Rosenberg (1997): On optimal strategies for cycle-stealing in networks of workstations. *IEEE Trans. Comp. 46*, 545-557.

[4] R. Blumofe and C.E. Leiserson (1993): Space-efficient scheduling of multithreaded computations. *25th ACM Symp. on Theory of Computing*, 362-371.

[5] R. Blumofe and C.E. Leiserson (1994): Scheduling multithreaded computations by work stealing. *35th IEEE Symp. on Foundations of Computer Science*, 356-368.

[6] R. Blumofe and D.S. Park (1994): Scheduling large-scale parallel computations on networks of workstations. *3rd Intl. Symp. on High-Performance Distributed Computing*, 96-105.

[7] T. Hagerup (1998): Allocating independent tasks to parallel processors: an experimental study. *J. Parallel Distr. Comput.*, to appear.

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

[9] C.H. Papadimitriou and M. Yannakakis (1990): Towards an architecture-independent analysis of parallel algorithms. *SIAM J. Comput. 19*, 322-328.

[10] G.F. Pfister (1995): *In Search of Clusters.* Prentice-Hall, Upper Saddle River, N. J.

[11] A.L. Rosenberg (1998): Guidelines for data-parallel cycle-stealing in networks of workstations, I: on maximizing expected output. Tech. Rpt. 98-15, Univ. Massachusetts. See also *12th IEEE Intl. Parallel Proc. Symp.* (1998) 519-523.

[12] M. Stumm (1988): The design and implementation of a decentralized scheduling facility for a workstation cluster. *2nd IEEE Conf. on Computer Workstations*, 12-22.

[13] M.M. Theimer and K.A. Lantz (1989): Finding idle machines in a workstation-based distributed environment. *IEEE Trans. Software Eng'g. 15*, 1444-1458.