

# Optimal Schedules for Cycle-Stealing in a Network of Workstations with a Bag-of-Tasks Workload

Arnold L. Rosenberg, *Fellow, IEEE*

**Abstract**—We refine the model underlying our prior work on scheduling bag-of-tasks (“embarrassingly parallel”) workloads via cycle-stealing in networks of workstations [5], [17], obtaining a model wherein the scheduling guidelines of [17] produce optimal schedules for *every* such cycle-stealing opportunity. We thereby render *prescriptive* the *descriptive* model of those sources. Although computing optimal schedules usually requires the use of general function-optimizing methods, we show how to compute optimal schedules *efficiently* for the broad class of opportunities whose durations come from a *concave* probability distribution. Even when no such efficient computation of an optimal schedule is available, our refined model often suggests a natural notion of *approximately* optimal schedule, which may be efficiently computable. We illustrate such efficient approximability via the important class of cycle-stealing opportunities whose durations come from a *heavy-tailed* distribution. Such opportunities do not admit any optimal schedule—nor even a natural notion of approximately optimal schedule—within the model of [5], [17]. Within our refined model, though, we derive computationally simple schedules for heavy-tailed opportunities, which can be “tuned” to accomplish an expected amount of work that is arbitrarily close to optimal.

**Index Terms**—Cycle-stealing, bag-of-tasks workloads, heavy-tailed distributions, networks of workstations (NOWs), optimal scheduling, scheduling parallel computations.

## 1 INTRODUCTION

NUMEROUS sources eloquently argue the technological and economic inevitability of an increasingly common modality of parallel computing: the use of a network of workstations (NOW) as a parallel computer; cf. [1], [16]. Sources too numerous to list describe systems that facilitate the mechanics of NOW-based computing, often via the technique of *cycle-stealing*—the use by one workstation of idle computing cycles of another—which is our interest here. To this point, however, rather few sources have studied the problem of scheduling individual computations on NOWs, and even fewer have presented rigorously analyzed algorithms that schedule broad classes of individual computations well. In the current paper, we refine the model introduced in [5] and developed in [17], in a way that allows one to devise schedules that maximize the expected amount of work accomplished from a bag-of-tasks (“embarrassingly parallel”) workload, for *every* cycle-stealing opportunity, given knowledge of the instantaneous probability that the opportunity will be terminated by the owner of the “borrowed” workstation. We thereby render *prescriptive* the *descriptive* model of [5], [17]. We expose a number of situations wherein one can derive (nearly) optimal schedules *efficiently*.

### 1.1 Background

The model of [5], [17] views cycle-stealing in NOWs as an adversarial process in which the owner of workstation  $A$

contracts to take control of workstation  $B$  whenever its owner is absent, with the commitment of relinquishing control of  $B$  *immediately* when its owner returns.<sup>1</sup> In this context, “relinquishing control immediately” implies killing any active job(s)—thereby losing all results since the last checkpoint.

**Note 1.** 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 even when not inevitable, because of the degraded service that  $B$ ’s owner experiences when  $A$ ’s jobs remain active, even with lowered priority.

This contract presents a challenging dilemma for the owner of workstation  $A$  when scheduling a cycle-stealing “opportunity” (the period of time when  $A$ ’s owner has access to  $B$ ). On the one hand, the typically large overhead required to set up an interworkstation communication recommends that  $A$  communicate with  $B$  very infrequently, sending large quantities of work each time—in order to minimize the cumulative communication setup time. On the other hand, the harsh interrupt provision of the contract recommends that  $A$  communicate with  $B$  very frequently, sending small quantities of work each time—in order to keep the amount of (vulnerable) remote work small at all times.

Clearly, cycle-stealing within the described adversarial model can accomplish productive work only if the metaphorical “malicious adversary” is somehow restrained from just interrupting every period when  $B$  is doing work

• The author is with the Department of Computer Science, University of Massachusetts, Amherst, MA 01003. E-mail: rsnbrg@cs.umass.edu.

Manuscript received 17 Apr. 2000; accepted 2 July 2001.  
For information on obtaining reprints of this article, please send e-mail to: tpd@computer.org, and reference IEEECS Log Number 111939.

1. We distinguish the adversarial enterprise of cycle-stealing from its cooperative analogue, *work-sharing*, wherein workstation  $A$ ’s owner has guaranteed access to  $B$  for some period; cf. [19].

for  $A$ , thereby killing all work done by  $B$ . The restraint studied in the *Known-Risk* model of [5], [17] and the current paper resides in two assumptions: 1) that we know the instantaneous probability that workstation  $B$  has *not* been reclaimed and 2) that the *life function*  $\mathcal{P}$  that exposes this probabilistic information— $\mathcal{P}(t)$  is the probability that  $B$  has not been reclaimed by its owner by time  $t$ ; cf. Section 2.1—is “smooth.” It is shown in [17] (cf. Theorem 2.1) that, for bag-of-tasks (“embarrassingly parallel”) workloads, this simple model exposes constraints that any optimal schedule must satisfy—a schedule’s optimality being measured in terms of its expected production of work—and that the guidelines that emerge from these constraints yield optimal schedules for large classes of—but not all—cycle-stealing opportunities.

The current paper is motivated by the inability of the *Known-Risk* model to deal satisfactorily with all possible cycle-stealing opportunities. Specifically, there exist opportunities that provably do not admit any optimal schedule within the model [17]. Thus, the scheduling guidelines of the latter source, while necessary for optimal scheduling, are not sufficient. This shortcoming is not of just academic interest, since the important class of opportunities whose durations come from a *heavy-tailed* distribution<sup>2</sup>—wherein the probability that  $B$  has not been reclaimed roughly halves as the length of the opportunity doubles—do not admit optimal schedules within the model. Even worse, these opportunities have infinite mean durations, which obscures even a plausible definition of “approximately optimal” schedule for such an opportunity.

## 1.2 Our Main Results

In the present paper, we refine the *Known-Risk* model, obtaining a model within which the scheduling guidelines of [17] yield an optimal schedule for *every* cycle-stealing opportunity (Theorem 3.1) with a bag-of-tasks workload. Our refinement, developed in Section 3, resides in the notion of a *bounded-lifespan analogue* (BLA) of a cycle-stealing life function  $\mathcal{P}$ —a finite-duration life function that captures the essential risk-exposing structure of  $\mathcal{P}$ . While the process of computing optimal schedules for (BLAs of) arbitrary life functions usually requires the use of (often inefficient) general function-optimization techniques (such as, e.g., simulated annealing), we show in Section 4.1 that our scheduling guidelines yield *efficiently computable* optimal schedules for every cycle-stealing opportunity whose duration is governed by a *concave* life function (Theorem 4.1). Even when dealing with an opportunity whose life function is not concave, our scheduling guidelines for BLAs often suggest a natural notion of approximately optimal schedule, which may be efficiently computable. We illustrate this latter situation in Section 4.2, where we craft computationally simple schedules for (bounded-lifespan) heavy-tailed opportunities, which can be “tuned” to be arbitrarily close to optimal (Theorem 4.2). Finally, in Section 5, we use simulations to evaluate all of our schedules for heavy-tailed opportunities, both against one another and against

the ideal work production for such opportunities. These simulation-based comparisons reinforce our mathematical analyses by illustrating that our schedules are almost ideally productive, as long as the tasks comprising our workload are sufficiently coarse-grained (or, compute-intensive) to balance the large overhead for interworkstation communications.

## 1.3 Related Work

The literature contains relatively few rigorously analyzed scheduling algorithms for parallel computing in NOWs. Among those we know of, only [3], [5], [17], [18] and the current study deal with an *adversarial* model of cycle-stealing. One finds in [3] a randomized cycle-stealing strategy which, with high probability, accomplishes within a logarithmic factor of optimal work production. In [5], [17], [18], 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 kill all work in progress and thereby minimize the amount of work produced during a cycle-stealing opportunity. As noted earlier, the *Known-Risk* model of [5], [17] provides the starting point for our study; [18] develops the *Guaranteed-Output* model of [5], providing guidelines which optimize, to within low-order additive terms, the *guaranteed* amount of work produced during a cycle-stealing opportunity—given knowledge of the duration of the opportunity, plus an upper bound on the number of potential interruptions by the adversary.

A number of sources view parallel computing in a NOW as a *cooperative* venture, wherein overloaded workstations share their load with idle ones (*work-sharing*) [2], [19] or idle workstations borrow load from busy ones (*work-stealing*) [6], [7], [8]. The study in [2] develops an “auction”-based model wherein one determines that subset of workstations which—according to the source’s cost model—promises the best performance on one’s workload. One can view [19] as a follow-up to [2], wherein one determines both how much work to allocate to the individual workstations in the selected subset and a schedule for sending this work, in a way that optimizes the amount of work that can be accomplished within the period of the subset’s availability. One finds in [4], [9] two approaches to a model for scheduling collective communication in a (hierarchical, in the case of [9]) heterogeneous NOW, similar to the communication-oriented portion of the model studied in [19]. A variety of communication schedules are studied via simulations in [4]; an approximation to an optimal algorithm for broadcast and reduction is derived in [9]. The CILK system of [6], [7], [8] implements a multithreading protocol for “work-stealing” in NOWs, wherein idle workstations borrow load asymptotically optimally, with respect to both speed of computation and space overhead. Finally, one finds in [10] a protocol for allocating resources “fairly” among the members of a “computational cooperative;” although the main focus is on far-flung assemblages of NOWs, the results are relevant to individual NOWs also.

We do not enumerate here the many studies of computation on NOWs, which focus either on systems that enable one workstation to steal cycles from another or on specific algorithmic applications. However, we point to [14]

2. In [12], [13], the occurrence of heavy-tailed distributions in important computational settings is discussed.

as an exemplar of the former type of study and to [20] as an exemplar of the latter.

## 2 FORMAL BACKGROUND

### 2.1 The Known-Risk Cycle-Stealing Model

The owner of workstation  $A$  wants to utilize workstation  $B$  (via cycle-stealing) to assist in the computation of a large (“infinite”) supply of mutually independent (“embarrassingly parallel”) tasks, which are identical in size and computational complexity. We assume that each pair of communications in which  $A$  sends work to  $B$  and  $B$  returns the results of that work to  $A$  incurs a fixed setup-cost  $c$ .<sup>3</sup> We keep  $c$  independent of the marginal per-task cost of communicating between  $A$  and  $B$  by incorporating the latter cost into the time for computing a task. Our scheduling model assumes that we know exactly how long each task takes on workstation  $B$ —which is consistent with our model’s view that  $B$  is dedicated to our work during the cycle-stealing opportunity. In accord with this assumption, our model measures time in work-units (rather than wall-clock time), where one *unit of work* is the time it takes for:

- workstation  $A$  to transmit a single task to workstation  $B$  (this is the marginal transmission time for this task: the (fixed) setup time for each communication—during which many tasks will typically be transmitted—is accounted for by the parameter  $c$ );
- workstation  $B$  to execute that task;
- workstation  $B$  to return its results for that task to workstation  $A$ .

We view a cycle-stealing opportunity as a sequence of *episodes* during which workstation  $A$  has access to workstation  $B$ , punctuated by *interrupts* caused by the return of  $B$ ’s owner. When scheduling an opportunity, we decrease our vulnerability to interrupts, with their attendant loss of work in progress on  $B$ , by partitioning each episode into *periods*, each beginning with  $A$  sending work to  $B$  and ending either with an interrupt or with  $B$  returning the results of that work. Since our discretionary power thus resides solely in deciding how much work to send in each period, we view an (*episode*)-*schedule* simply as a sequence of positive period-lengths:  $S = t_0, t_1, \dots$ . A length- $t$  period in an episode accomplishes  $t \ominus c \stackrel{\text{def}}{=} \max(0, t - c)$  units of work if it is not interrupted and 0 units of work if it is interrupted. Thus, the episode scheduled by  $S$  accomplishes  $\sum_{i=1}^{k-1} (t_i \ominus c)$  units of work when it is interrupted during period  $k$ .

As noted earlier, we assume that we know the risk of  $B$ ’s being reclaimed, via a decreasing *life function*,

$$\mathcal{P}(t) \stackrel{\text{def}}{=} \Pr(B \text{ is "alive" at time } t),$$

which: 1) satisfies  $\mathcal{P}(0) = 1$  (to indicate  $B$ ’s availability at the start of the episode); 2) when an upper bound  $L$  on the episode’s lifespan ( $\stackrel{\text{def}}{=} \text{its maximum possible duration}$ ) exists, satisfies  $\mathcal{P}(L) = 0$  (to indicate that the interrupt will

have occurred by time  $L$ ). Our earlier assertion that life functions must be “smooth” is embodied in the formal requirement that  $\mathcal{P}$  be *twice differentiable*. An important statistic of an episode with life function  $\mathcal{P}$  is its *mean lifespan*:<sup>4</sup>

$$\text{Mean-Lifespan}(\mathcal{P}) \stackrel{\text{def}}{=} - \int_0^U t \mathcal{P}'(t) dt = \int_0^U \mathcal{P}(t) dt. \quad (2.1)$$

**Note 2.** *The simplification of the integral in (2.1) results from integration by parts, in the light of the Known-Risk model’s just-described constraints on life functions.*

The upper limit  $U$  of the integral is the episode’s lifespan  $L$  if it is finite, and is  $\infty$  otherwise. Our challenge is to maximize the *expected work production* from an episode governed by  $\mathcal{P}$ , i.e., to find a schedule  $S$  whose expected work production,

$$\text{Exp-Work}(S; \mathcal{P}) \stackrel{\text{def}}{=} \sum_{i \geq 0} (t_i \ominus c) \mathcal{P}(T_i), \quad (2.2)$$

is maximum, over all schedules for  $\mathcal{P}$ . In summation (2.2), each  $T_i$  is the partial sum

$$T_i \stackrel{\text{def}}{=} t_0 + t_1 + \dots + t_i;$$

the upper limit of the summation is the episode’s lifespan  $L$  if it is finite, and is  $\infty$  otherwise.

We close this description of the Known-Risk model with a lemma which can be helpful when one tries to compute (almost) optimal schedules. The lemma allows one to use ordinary ( $-$ ), rather than positive ( $\ominus$ ) subtraction in all but the last term of (2.2) as one seeks good schedules. Say that a schedule  $S$  is *productive* if each of its periods—save the last if  $S$  has finitely many periods—has length  $> c$ .

**Lemma 2.1** ([5], [17]). *One can effectively replace any schedule  $S$  for life function  $\mathcal{P}$  by a productive schedule  $\hat{S}$  such that  $\text{Exp-Work}(\hat{S}; \mathcal{P}) \geq \text{Exp-Work}(S; \mathcal{P})$ .*

**Proof Hint.** One can never decrease the expected work production of a schedule if one combines a “short” nonterminal period with its successor.  $\square$

Lemma 2.1 allows us to rewrite (2.2) for any finite schedule  $S = t_0, t_1, \dots, t_{m-1}$  in the following form (whose “missing” last term reflects the fact that  $\mathcal{P}(L) = 0$ ).

$$\text{Exp-Work}(S; \mathcal{P}) = \sum_{i=0}^{m-2} (t_i - c) \mathcal{P}(T_i). \quad (2.3)$$

Henceforth, we restrict attention to *productive schedules unless otherwise indicated*.

### 2.2 The Scheduling Guidelines of [17]

In [17], we extended the case studies from [5] by deriving a set of guidelines for (almost optimally) scheduling large classes of cycle-stealing opportunities within the Known-Risk model. These guidelines partially expose the structure

3. Our cost parameter  $c$  plays the role of the communication-cost parameter of [15], the “overhead” parameter  $o$  of [11], and the setup-plus-latency sum  $\sigma + \lambda$  of [9].

4. As usual,  $f'$  (resp.,  $f''$ ) denotes the first (resp., the second) derivative of the univariate function  $f$ .

of any optimal schedule for a “smooth” life function  $\mathcal{P}$ —whenever  $\mathcal{P}$  admits an optimal schedule. The guidelines are, thus, *necessary* for the optimality of a schedule.

Say that a life function  $\mathcal{P}$  is *concave* (respectively, *convex*) if its derivative  $\mathcal{P}'$

- never vanishes at a point  $x$  where  $\mathcal{P}(x) > 0$ ;
- is everywhere nonincreasing (respectively, everywhere nondecreasing): for all positive real  $\xi$  and  $\eta > \xi$ , we have  $\mathcal{P}'(\xi) \geq \mathcal{P}'(\eta)$  (respectively,  $\mathcal{P}'(\xi) \leq \mathcal{P}'(\eta)$ ).

**Theorem 2.1 ([17]).** *If the productive schedule  $\mathcal{S} = t_0, t_1, \dots$  is optimal for the differentiable life function  $\mathcal{P}$ , then:*

1. *For each period-index  $k \geq 0$ , save the last if  $\mathcal{S}$  is finite, period-length  $t_k$  is given implicitly by*

$$\mathcal{P}(T_k) = \max\left(0, -\sum_{j \geq k} (t_j - c)\mathcal{P}'(T_j)\right). \quad (2.4)$$

*Adjacent pairs of these equations combine to yield the following computationally friendlier system.*

2. *For each period-index  $k \geq 1$ , save the last if  $\mathcal{S}$  is finite, period-length  $t_k$  is given implicitly by*

$$\mathcal{P}(T_k) = \max(0, \mathcal{P}(T_{k-1}) + (t_{k-1} - c)\mathcal{P}'(T_{k-1})). \quad (2.5)$$

3. *When  $\mathcal{P}$  is convex (respectively, concave), the initial period-length  $t_0$  is bounded above and below as follows, with the parameter  $\varphi = 1$  (respectively,  $\varphi = 1/2$ ).*

$$\sqrt{\frac{c^2}{4} - \frac{c\mathcal{P}(t_0)}{\mathcal{P}'(t_0)}} + \frac{c}{2} \leq t_0 \leq 2\sqrt{\frac{c^2}{4} - \frac{c\mathcal{P}(t_0)}{\mathcal{P}'(\varphi t_0)}} + c. \quad (2.6)$$

**Proof Hint.** The chain of reasoning leading to systems (2.4, 2.5) begins with the observation that the expected work production of an optimal schedule  $\mathcal{S}$  is no smaller than that of any of  $\mathcal{S}$ 's “shifts.” The  $\langle k, -\delta \rangle$ -*shift*,  $\mathcal{S}^{(k, -\delta)}$ , of  $\mathcal{S}$  and the  $\langle k, +\delta \rangle$ -*shift*,  $\mathcal{S}^{(k, +\delta)}$ , of  $\mathcal{S}$  are schedules which have the same number of periods<sup>5</sup> as  $\mathcal{S}$  and the same period-lengths, save for period  $k$ . Specifically,

$$\mathcal{S}^{(k, -\delta)} \stackrel{\text{def}}{=} t_0, t_1, \dots, t_{k-1}, t_k - \delta, t_{k+1}, \dots,$$

$$\mathcal{S}^{(k, +\delta)} \stackrel{\text{def}}{=} t_0, t_1, \dots, t_{k-1}, t_k + \delta, t_{k+1}, \dots$$

The chain of reasoning leading to (2.6) begins with the observation that the expected work production of an optimal schedule  $\mathcal{S}$  cannot be increased by telescoping adjacent periods of  $\mathcal{S}$  into a single period.  $\square$

**Note 3.**

1. *As is pointed out in [17], the guidelines inherent in the system (2.5) can be applied in an online fashion, computing  $t_{k+1}$  only after period  $k$  ends. This means*

5. That is, if  $\mathcal{S}$  has finitely many periods, then  $\mathcal{S}^{(k, +\delta)}$  and  $\mathcal{S}^{(k, -\delta)}$  have the same number; if  $\mathcal{S}$  has infinitely many periods, then so also do  $\mathcal{S}^{(k, +\delta)}$  and  $\mathcal{S}^{(k, -\delta)}$ .

*that one can correct errors in life functions that are known only approximately, and/or one can use conditional rather than absolute probabilities to craft one's schedules.*

2. *It is shown in [17] that the guidelines yield, via general principles, optimal schedules for the life functions that were optimized via ad hoc analyses in [5].*

While the guidelines of Theorem 2.1 are shown in [17] to be quite useful in crafting near-optimal schedules for many life functions, it is also shown there that some cycle-stealing opportunities do not admit any optimal schedule within the Known-Risk model. The important (cf. footnote 2) opportunities whose durations are governed by the *heavy-tailed* life function

$$\mathcal{P}_{(\text{ht})}(t) \stackrel{\text{def}}{=} \frac{1}{t+1}$$

fall within this intransigent class.

**Proposition 2.1 ([17]).** *The heavy-tailed life function  $\mathcal{P}_{(\text{ht})}$  does not admit an optimal schedule.*

**Proof Hint.** One can always increase expected work production by replacing a given schedule  $\mathcal{S} = t_0, t_1, \dots$  for  $\mathcal{P}_{(\text{ht})}$  by the schedule  $\mathcal{S}^{(2)} \stackrel{\text{def}}{=} 2t_0, 2t_1, \dots$ .  $\square$

$\mathcal{P}_{(\text{ht})}$ 's intransigence, as exposed in Proposition 2.1, is exacerbated by its resistance to approximation: Since Mean-Lifespan( $\mathcal{P}_{(\text{ht})}$ ) is infinite, there is no apparent natural notion of “approximately optimal” expected work production to strive for when crafting a schedule for  $\mathcal{P}_{(\text{ht})}$ .

### 3 BOUNDED-LIFESPAN ANALOGUES OF LIFE FUNCTIONS

We now refine the Known-Risk model by replacing each life function with its family of BLAs, as described in Section 1.2. After defining BLAs formally and determining their impact on the Known-Risk model (Section 3.1) and on our scheduling guidelines (Section 3.2), we show that BLAs achieve the desired goal: Every BLA of every life function admits a computable optimal schedule whose period-lengths are given by our guidelines (Section 3.3). We turn to the issue of the ease of computing optimal schedules in Section 4.

#### 3.1 Lifespan- $L$ Analogues of Life Functions

Say that the lifespan  $L > 0$  is *relevant* for the life function  $\mathcal{P}$  if  $\mathcal{P}(t) > 0$  for all  $t < L$ . For each function  $\mathcal{P}$  and each relevant  $L$ , the *lifespan- $L$  analogue* of  $\mathcal{P}$ , denoted  $\mathcal{P}^{(L)}$ , is the life function

$$\mathcal{P}^{(L)}(t) \stackrel{\text{def}}{=} \frac{\mathcal{P}(t) - \mathcal{P}(L)}{1 - \mathcal{P}(L)}. \quad (3.1)$$

Easily, each BLA  $\mathcal{P}^{(L)}$  is a valid life function (cf. the definitions in Section 2.1) with maximum lifespan  $L$ . Moreover, BLAs extend the Known-Risk model *gracefully*, in the sense that  $\mathcal{P}^{(L)}(t) \equiv \mathcal{P}(t)$  whenever  $\mathcal{P}$  intrinsically has maximum lifespan  $L$  (as, e.g., do the *uniform-risk* life functions,  $\mathcal{P}_L(t) \stackrel{\text{def}}{=} 1 - t/L$ , which form one of the case studies in [5]).

**Note 4.** Each BLA  $\mathcal{P}^{(L)}$  is intended to preserve the “essential structure” of its parent life function  $\mathcal{P}$ , including mathematical properties such as differentiability and, when appropriate, concavity or (as with  $\mathcal{P}_{(\text{ht})}$ ) convexity.

To illustrate the transformation from a life function to its BLA:

- For  $\hat{L} \leq L$ , the lifespan- $\hat{L}$  BLA of the uniform-risk life function  $\mathcal{P}_L(t)$  is  $\mathcal{P}_L^{(\hat{L})}(t) = 1 - t/\hat{L}$ .
- The lifespan- $L$  BLA of the infinite mean-lifespan heavy-tailed life function  $\mathcal{P}_{(\text{ht})}(t)$  is

$$\mathcal{P}_{(\text{ht})}^{(L)}(t) = \frac{1}{L} \left( \frac{L-t}{t+1} \right). \quad (3.2)$$

### 3.2 The Impact of BLAs on the Scheduling Guidelines of [17]

Theorem 2.1’s guidelines for a life function  $\mathcal{P}$  translate easily to guidelines for  $\mathcal{P}$ ’s lifespan- $L$  analogue  $\mathcal{P}^{(L)}$ .

**Proposition 3.1.**

1. The difference in (2.4) for  $\mathcal{P}$  and  $\mathcal{P}^{(L)}$  resides only in the effect of the term  $(-\mathcal{P}(L))$  from the numerator of (3.1). Thus, the system becomes

$$\mathcal{P}(T_k) - \mathcal{P}(L) = \max \left( 0, - \sum_{j \geq k} (t_j - c) \mathcal{P}'(T_j) \right). \quad (3.3)$$

2. The recurrence of (2.5) for the noninitial period-lengths of life function  $\mathcal{P}^{(L)}$  is identical to the analogous recurrence for  $\mathcal{P}$ .
3. The difference in the bounds (2.6) on  $t_0$  for  $\mathcal{P}$  and  $\mathcal{P}^{(L)}$  when  $\mathcal{P}$  is concave or convex, resides only in the effect of the term  $(-\mathcal{P}(L))$  from the numerator of (3.1).

**Proof Sketch.** The factor  $(1 - \mathcal{P}(L))$  from the denominator of (3.1) cancels out in all three cases. In Part 3, this is because

$$\frac{\mathcal{P}^{(L)}(t_0)}{(\mathcal{P}^{(L)})'(\varphi t_0)} = \frac{\mathcal{P}(t_0) - \mathcal{P}(L)}{\mathcal{P}'(\varphi t_0)}.$$

In Part 2, the term  $(-\mathcal{P}(L))$  from the numerator of (3.1) also cancels out when instantiated in (2.5).  $\square$

We now illustrate Proposition 3.1 by instantiating the guidelines of Theorem 2.1 for both  $\mathcal{P}_{(\text{ht})}$  and  $\mathcal{P}_{(\text{ht})}^{(L)}$ . This is not an empty exercise, even though  $\mathcal{P}_{(\text{ht})}$  does not admit any optimal schedule (Proposition 2.1). First, we shall see in Theorem 3.1 that these guidelines do specify an optimal schedule for  $\mathcal{P}_{(\text{ht})}^{(L)}$ . Second, the “guidelines” for  $\mathcal{P}_{(\text{ht})}$  supply the inspiration for the computationally simple, provably good schedules for  $\mathcal{P}_{(\text{ht})}^{(L)}$  that we present in Section 4.2.

**Proposition 3.2.** Assume that the heavy-tailed life function  $\mathcal{P}_{(\text{ht})}$  admitted an optimal schedule  $\mathcal{S} = t_0, t_1, \dots$  and that the heavy-tailed BLA  $\mathcal{P}_{(\text{ht})}^{(L)}$  admitted an optimal schedule  $\mathcal{S}^{(L)} = t_0^{(L)}, t_1^{(L)}, \dots, t_{m-1}^{(L)}$ . Then:

1. Letting  $\tilde{t}_i$  (respectively,  $\tilde{T}_i$ ) ambiguously denote  $t_i$  and  $t_i^{(L)}$  (respectively,  $T_i$  and  $T_i^{(L)}$ ) for  $i \geq 0$ , the sequence of period-lengths for both  $\mathcal{S}$  and  $\mathcal{S}^{(L)}$  would satisfy the recurrence

$$\tilde{t}_{k+1} = (\tilde{t}_k - c) \frac{\tilde{T}_k + 1}{\tilde{T}_{k-1} + c + 1}. \quad (3.4)$$

2. The initial period-length  $t_0$  for  $\mathcal{S}$  would be bounded as follows:

$$c + \sqrt{c^2 + c} \leq t_0 \leq 3c + \sqrt{9c^2 + 4c}.$$

3. The initial period-length  $t_0^{(L)}$  for  $\mathcal{S}^{(L)}$  would be bounded as follows:

$$\frac{cL}{L+2} + \sqrt{\left(\frac{cL}{L+2}\right)^2 + \frac{cL}{L+2}} \leq t_0^{(L)} \leq \frac{(3L-1)c}{L+5} + \sqrt{\left(\frac{(3L-1)c}{L+5}\right)^2 + \frac{4cL}{L+5}}.$$

### 3.3 Opportunities Governed by BLAs Admit Optimal Schedules

We show now that BLAs do, indeed, serve the purpose that motivated their invention. To wit, the guidelines of Theorem 2.1 provide optimal schedules for the BLAs of every life function  $\mathcal{P}$ . Modulo the complexity of actually computing these optimal schedules, we have thus succeeded in solving the scheduling problem for the Known-Risk model.

**Theorem 3.1.** Every BLA  $\mathcal{P}^{(L)}$  admits an optimal productive schedule whose period-lengths are determined by (3.3).

**Proof.** We first establish nonconstructively that every BLA admits an optimal finite schedule. We then invoke Lemma 2.1 to infer that every BLA admits a *productive* optimal schedule. We finally invoke Theorem 2.1 and Proposition 3.1 to infer that the period-lengths of a productive optimal schedule are specified by system (3.3).

**The existence of optimal schedules.** We build on two lemmas. The first lemma establishes an upper bound on the amount of expected work that one can produce during a finite-lifespan episode.

**Lemma 3.1.** For any schedule  $\mathcal{S}$  for life function  $\mathcal{P}$ ,

$$\text{Exp-Work}(\mathcal{S}; \mathcal{P}) \leq \text{Mean-Lifespan}(\mathcal{P}).$$

**Proof of Lemma.** By (2.2), the expected work production of  $\mathcal{S} = t_0, t_1, \dots$  can be viewed as an underestimate, obtained by abutting rectangles of widths  $t_i \ominus c$  and heights  $\mathcal{P}(T_i)$  (for  $i = 0, 1, \dots$ ), of the area under the curve  $\mathcal{P}(t)$ . By (2.1), the latter area is the mean-lifespan of the associated episode.  $\square$

Next, we invoke the weak-inequality version of Lemma 2.1 (which is the version that appears in [5]) to infer that, if a lifespan- $L$  BLA  $\mathcal{P}^{(L)}$  admits an optimal schedule, then it admits one of the form

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

where  $t_{m-1} \geq 0$  (by definition), and each other  $t_i \geq c$ . One consequence of these constraints on the period-lengths of  $\mathcal{S}$  is that we lose no generality if we restrict our search for optimal schedules for  $\mathcal{P}^{(L)}$  to schedules that have  $\leq \lceil L/c \rceil$  periods.

Embarking on this search, let us define, for each  $m \in \{2, 3, \dots, \lceil L/c \rceil\}$ , the  $m$ -variable formal<sup>6</sup> work-function for  $\mathcal{P}^{(L)}$ :

$$\mathcal{W}_m^{(L)}(\tau_0, \tau_1, \dots, \tau_{m-1}) \stackrel{\text{def}}{=} \sum_{i=0}^{m-2} (\tau_i - c) \mathcal{P}^{(L)}(\tau_0 + \tau_1 + \dots + \tau_i). \quad (3.5)$$

By (2.3), the expected work production of any  $m$ -period schedule  $\mathcal{S} = t_0, t_1, \dots, t_{m-1}$  for  $\mathcal{P}^{(L)}$  is given by

$$\text{Exp-Work}(\mathcal{S}; \mathcal{P}) = \mathcal{W}_m^{(L)}(t_0, t_1, \dots, t_{m-1}).$$

As just noted, therefore, we need consider only these  $\lceil L/c \rceil - 1$  work-functions as we search for a work-optimizing schedule for  $\mathcal{P}^{(L)}$ . By Lemma 3.1, each work-function  $\mathcal{W}_m^{(L)}$  is a bounded, continuous (indeed, differentiable) function. Therefore, on the compact set of real  $m$ -tuples  $\langle \tau_0, \tau_1, \dots, \tau_{m-1} \rangle$  defined by the three constraints:

$$\begin{aligned} \tau_{m-1} &\geq 0 \\ \text{each other } \tau_i &\geq c \\ \tau_0 + \tau_1 + \dots + \tau_{m-1} &= L, \end{aligned} \quad (3.6)$$

$\mathcal{W}_m^{(L)}$  must achieve a maximum value. It follows that any  $p$ -tuple

$$\mathcal{S}^* \stackrel{\text{def}}{=} t_0^*, t_1^*, \dots, t_{p-1}^*,$$

which simultaneously

- satisfies constraints (3.6);
- achieves the largest  $\mathcal{W}_m^{(L)}$ -value over all relevant numbers of periods  $m$

is an optimal schedule for  $\mathcal{P}^{(L)}$ . Since the last sentence may be hard to read due to its many quantifiers expressed in natural language, we state formally that our intention is that:

$$\begin{aligned} \mathcal{W}_p^{(L)}(t_0^*, t_1^*, \dots, t_{p-1}^*) &= \\ \max_{m \in \{2, 3, \dots, \lceil L/c \rceil\}} \{ &\mathcal{W}_m^{(L)}(t_0, t_1, \dots, t_{m-1}) \mid \\ \text{constraints (3.6) hold} \}. & \end{aligned}$$

**The Existence of Productive Optimal Schedules.** We can now invoke the strong-inequality version of Lemma 2.1 to infer the existence of a productive schedule  $\mathcal{S}^*$  whose expected work production matches  $\mathcal{S}^*$ 's. (Of course,  $\mathcal{S}^*$  may have fewer than  $p$  periods.)

**Computing a productive optimal schedule.** Finally, we invoke Theorem 2.1 to complete the proof.  $\square$

6. We term the function  $\mathcal{W}_m^{(L)}$  "formal" because it presents the  $m$  period-lengths as mutually independent variables. We know by Lemma 2.1 and Theorem 2.1 that the period-lengths for optimal schedules are not mutually independent.

Theorem 3.1 provides us with good news, yet does not provide a completely satisfying solution to our scheduling problem. Exemplifying the good news, we can now optimally schedule finite-duration heavy-tailed opportunities.

**Corollary 3.1.** *The schedule  $\mathcal{S}^{(L)}$  of Proposition 3.2 is optimal for the heavy-tailed BLA  $\mathcal{P}_{(\text{ht})}^{(L)}$ .*

Moderating the Theorem's good news is the potential computational difficulty of finding the optimal schedules it guarantees. We remarked in [17] on the computational unfriendliness of system (2.4). This observation led us there to propose the less comprehensive, but (in our experience) quite friendly system (2.5) to specify the noninitial period-lengths of optimal schedules, augmented, in the case of concave and convex life functions, by the bounds (2.6) on the initial period-lengths of optimal schedules. The non-comprehensive nature of (2.5, 2.6) means that, even with Theorem 3.1's guarantee that optimal schedules always exist for BLAs, one may have to employ general (and usually inefficient) function-maximizing techniques (such as, e.g., simulated annealing) to the work-functions (3.5) in order to find those schedules. In the next section, we show that such inefficiency can sometimes be avoided, at least in special cases.

## 4 BLAs THAT EFFICIENTLY ADMIT "GOOD" SCHEDULES

This section is devoted to the question of the computational efficiency of deriving (almost) optimal schedules for bounded-lifespan cycle-stealing opportunities. In Section 4.1, we show that the guidelines of Theorem 2.1 efficiently yield exactly optimal schedules for *concave* life functions. In Section 4.2, we exhibit a parameterized family of simply computed schedules for the important family of heavy-tailed BLAs, which can be tuned to be as close as desired to optimal in expected work production.

### 4.1 Efficient Optimal Schedules for Concave Life Functions

When the life function  $\mathcal{P}$  that governs a cycle-stealing opportunity is concave, we can improve on Theorem 3.1 by guaranteeing a rather efficient computation of an optimal productive schedule for  $\mathcal{P}$ , using the computationally friendly guidelines of (2.5), supplemented by the bounds of (2.6).

The reader may have noted that, in contrast with our careful distinction between a life function and its BLAs since the beginning of Section 3, we have been careless in the last two paragraphs about making this distinction. We begin our development in this section by justifying this carelessness, via a lemma which verifies the (not-surprising) fact that every cycle-stealing opportunity which is governed by a concave life function has a bounded lifespan. This fact follows from a bound on how fast the period-lengths of the opportunity's optimal schedule must decrease. This rate of decrease shows also that optimal schedules for lifespan- $L$  concave life functions have only (roughly)  $\sqrt{2L/c}$  periods, in contrast to our bound of  $\lceil L/c \rceil$  for general lifespan- $L$  life functions.

**Lemma 4.1.** *If  $\mathcal{S} = t_0, t_1, \dots$  is an optimal productive schedule for a concave life function  $\mathcal{P}$ , then:*

1. *for each nonterminal period-index  $i$ ,  $t_i \leq t_{i-1} - c$ ;*
2. *the life function  $\mathcal{P}$  has a bounded lifespan  $L_{\mathcal{P}}$ ;*
3. *schedule  $\mathcal{S}$  has fewer than*

$$\left\lceil \sqrt{\frac{2L_{\mathcal{P}}}{c} + \frac{1}{4} + \frac{1}{2}} \right\rceil \text{ periods.}$$

**Note 5.**

1. *The reader can easily adapt the proof of Lemma 4.1 to prove that, when  $\mathcal{P}$  is convex, then each nonterminal  $t_i \geq t_{i-1} - c$ .*
2. *In contrast to concave life functions, general life functions need not have finite schedules, nor need their optimal schedules have decreasing period-lengths: the unique optimal schedule for the life function  $\mathcal{P}(t) \stackrel{\text{def}}{=} 2^{-t}$  is infinite and has all period-lengths equal [5].*
3. *The quantitative claims of Lemma 4.1 cannot be improved in general: the unique optimal schedule  $\mathcal{S} = t_0, t_1, \dots, t_{m-1}$  for the lifespan- $L$  uniform-risk life function has*

$$m = \left\lceil \sqrt{\frac{2L_{\mathcal{P}}}{c} + \frac{1}{4} + \frac{1}{2}} \right\rceil$$

*periods, and, for each nonterminal period-index  $i$ ,  $t_i = t_{i-1} - c$  [5].*

4. *Lemma 4.1's assertion that the period-lengths of optimal schedules for concave life functions are strictly decreasing strengthens an analogous result in [5], which is proved there only with weak inequalities and only for the uniform-risk life function.*

**Proof.**

1. We exploit the optimality of  $\mathcal{S}$  only to infer that it is at least as productive as any of its  $\delta$ -perturbations,

$$\mathcal{S}^{[i, \delta]} \stackrel{\text{def}}{=} t_0, t_1, \dots, t_{i-1}, t_i + \delta, t_{i+1} - \delta, t_{i+2}, \dots$$

In other words, for every nonterminal period-index  $i$  and every real  $\delta > 0$ , the following difference is nonnegative:

$$\begin{aligned} & \text{Exp-Work}(\mathcal{S}; \mathcal{P}) - \text{Exp-Work}(\mathcal{S}^{[i, \delta]}; \mathcal{P}) \\ &= (t_i - c)[\mathcal{P}(T_i) - \mathcal{P}(T_i + \delta)] + \delta[\mathcal{P}(T_{i+1}) - \mathcal{P}(T_i + \delta)]. \end{aligned}$$

This nonnegativity implies that

$$\delta[\mathcal{P}(T_{i+1}) - \mathcal{P}(T_i + \delta)] \geq (t_i - c)[\mathcal{P}(T_i + \delta) - \mathcal{P}(T_i)],$$

so that:

$$\left( \frac{t_{i+1} - \delta}{t_i - c} \right) \frac{\mathcal{P}(T_{i+1}) - \mathcal{P}(T_i + \delta)}{t_{i+1} - \delta} \geq \frac{\mathcal{P}(T_i + \delta) - \mathcal{P}(T_i)}{\delta}. \quad (4.1)$$

Next, the Mean Value Theorem of the differential calculus asserts that, for every  $\delta > 0$ , there

exist real numbers  $\xi \in (T_i, T_i + \delta)$  and  $\eta \in (T_i + \delta, T_{i+1})$  such that

$$\begin{aligned} \mathcal{P}'(\xi) &= \frac{\mathcal{P}(T_i + \delta) - \mathcal{P}(T_i)}{\delta} \quad \text{and} \\ \mathcal{P}'(\eta) &= \frac{\mathcal{P}(T_{i+1}) - \mathcal{P}(T_i + \delta)}{t_{i+1} - \delta}. \end{aligned} \quad (4.2)$$

Finally, the concavity of  $\mathcal{P}$  implies that

$$\mathcal{P}'(\xi) \geq \mathcal{P}'(\eta), \quad (4.3)$$

because  $\xi < \eta$ . Since  $\mathcal{P}'$  is negative, (4.3) can coexist with (4.1) and (4.2) only if  $t_{i+1} - \delta < t_i - c$ . Since this last inequality holds for each  $i$  and for arbitrarily small  $\delta$ , we conclude that each

$$t_{i+1} \leq t_i - c.$$

2. The bound on  $\mathcal{P}$ 's lifespan follows by conjoining the fact that  $\mathcal{S}$ 's period-lengths decrease at the rate of at least  $c$  per period (by part 1) with the fact that all of  $\mathcal{S}$ 's periods, save the last, have length  $> c$  (Lemma 2.1).
3. Part 1 implies that schedule  $\mathcal{S}$  has some finite number  $m \leq t_0/c$  periods. If we look at parts 1 and 2 "from the vantage point of  $t_{m-1}$ ," we find that

$$\begin{aligned} L_{\mathcal{P}} &= t_0 + t_1 + \dots + t_{m-2} + t_{m-1} \\ &\geq mt_{m-1} + \binom{m}{2}c > \binom{m}{2}c. \end{aligned}$$

Part 3 now follows from "solving" the preceding bound on  $m$  in terms of  $L_{\mathcal{P}}$  and  $c$ .  $\square$

We are now ready for the main theorem of the section.

**Theorem 4.1.** *Every concave life function admits an efficiently computed optimal productive schedule whose period-lengths are determined by system (2.4).*

**Proof.** We address the theorem's two assertions in turn.

**The Existence of an Optimal Schedule.** All that Theorem 3.1 needed in order to establish the existence of guideline-based optimal schedules for BLAs of general life functions were upper bounds on the work production of BLAs and on the numbers of periods of their optimal productive schedules. Since Lemma 4.1 affords us analogous bounds for any concave life function, we can invoke the proof of Theorem 3.1 to infer that every concave life function admits an optimal productive schedule whose period-lengths are determined by (2.4).

**Efficiently Computing Optimal Schedules.** Let  $\mathcal{P}$  be an arbitrary concave life function, and let  $\mathcal{S}^* = t_0^*, t_1^*, \dots, t_{m-1}^*$  be an optimal productive schedule for  $\mathcal{P}$ . Let us revisit the definition (3.5) of the formal work-function  $\mathcal{W}_m^{(L_{\mathcal{P}})}$ , which we henceforth abbreviate as just  $\mathcal{W}_m$  (since the lifespan  $L_{\mathcal{P}}$  is a property of  $\mathcal{P}$  here). By direct calculation, one verifies that schedule  $\mathcal{S}^*$  satisfies (2.4) if, and only if, every first partial derivative

$$\begin{aligned} \frac{\partial}{\partial \tau_j} \mathcal{W}_m(\tau_0, \tau_1, \dots, \tau_{m-1}) &= \mathcal{P}(\tau_0 + \tau_1 + \dots + \tau_j) \\ &+ \sum_{k \geq j} (\tau_k - c) \mathcal{P}'(\tau_0 + \tau_1 + \dots + \tau_k) \end{aligned}$$

of  $\mathcal{W}_m$  vanishes at the point  $\vec{t}^* \stackrel{\text{def}}{=} (t_0^*, t_1^*, \dots, t_{m-1}^*)$ . Since  $\mathcal{S}^*$  has maximum expected work production over all schedules for  $\mathcal{P}$ , we expect all of the second partial derivatives of  $\mathcal{W}_m$  to be negative at point  $\vec{t}^*$ . If we look at these derivatives, though, we find an even stronger consequence of  $\mathcal{P}$ 's concavity: all of  $\mathcal{W}_m$ 's second partial derivatives are negative *throughout the region of interest*. To wit:

- For  $k < i$ :

$$\begin{aligned} \frac{\partial^2}{\partial \tau_i \partial \tau_k} \mathcal{W}_m(\tau_0, \tau_1, \dots, \tau_{m-1}) &= \mathcal{P}'(\tau_0 + \tau_1 + \dots + \tau_i) \\ &+ \sum_{j=i}^{m-2} (t_j - c) \mathcal{P}''(\tau_0 + \tau_1 + \dots + \tau_j). \end{aligned}$$

- For  $k > i$ :

$$\begin{aligned} \frac{\partial^2}{\partial \tau_i \partial \tau_k} \mathcal{W}_m(\tau_0, \tau_1, \dots, \tau_{m-1}) &= \mathcal{P}'(\tau_0 + \tau_1 + \dots + \tau_k) \\ &+ \sum_{j=k}^{m-2} (t_j - c) \mathcal{P}''(\tau_0 + \tau_1 + \dots + \tau_j). \end{aligned}$$

- For  $k = i$ :

$$\begin{aligned} \frac{\partial^2}{\partial \tau_i^2} \mathcal{W}_m(\tau_0, \tau_1, \dots, \tau_{m-1}) &= \mathcal{P}''(\tau_0 + \tau_1 + \dots + \tau_i) + \mathcal{P}'(\tau_0 + \tau_1 + \dots + \tau_i) \\ &+ \sum_{j=i}^{m-2} (t_j - c) \mathcal{P}''(\tau_0 + \tau_1 + \dots + \tau_j). \end{aligned}$$

The important thing to notice is that each second partial derivative is a sum of terms, each containing precisely one instance of precisely one of  $\mathcal{P}'$  and  $\mathcal{P}''$ . The negativity of  $\mathcal{W}_m$ 's second partial derivatives therefore follows from the fact that for a concave life function  $\mathcal{P}$ , both  $\mathcal{P}'$  and  $\mathcal{P}''$  are negative throughout the opportunity's lifespan. Since  $\mathcal{W}_m$ 's first derivatives vanish at point  $\vec{t}^*$ , and since its second derivatives are always negative, we infer that  $\vec{t}^*$  is the *unique* maximum of  $\mathcal{W}_m$ . We now exploit this uniqueness to compute the point  $\vec{t}^*$ , hence the desired optimal schedule  $\mathcal{S}^*$ .

1. We invoke the interperiod dependencies for optimal schedules specified by (2.5) to convert  $\mathcal{W}_m(\tau_0, \tau_1, \dots, \tau_{m-1})$  into a (formally, rather complex) function  $\overline{\mathcal{W}}_m$  of the single variable  $\tau_0$ .

2. We note from the preceding discussion of the derivatives of  $\mathcal{W}_m$ , coupled with the guarantees of Theorem 2.1, that the derivative of  $\overline{\mathcal{W}}_m$  vanishes at a unique value of  $\tau_0$  within the interval specified by the bounds (2.6).
3. We determine the unique root of  $\overline{\mathcal{W}}_m$  within the specified interval to any desired accuracy, using the technique of recursive doubling followed by binary search. This specifies the initial period-length  $t_0^*$  of schedule  $\mathcal{S}^*$ .
4. We invoke (2.5) again to determine all subsequent period-lengths of schedule  $\mathcal{S}^*$ .

Of course, we cannot quantify our assertion of the “efficiency” of this procedure, as such quantification depends on the functional form of  $\mathcal{P}$  and the desired accuracy in determining the period-lengths of schedule  $\mathcal{S}^*$ .  $\square$

## 4.2 Efficient Near-Optimal Schedules for Heavy-Tailed BLAs

This section is dedicated to indicating that, even in the absence of an efficient algorithm for computing an optimal schedule, one can sometimes infer from the guidelines of Theorem 2.1 and the bound of Lemma 3.1 an efficient way to approximate the expected work production of an optimal schedule. Happily, we are able to illustrate this for the important, intransigent heavy-tailed life function.

If the schedule  $\mathcal{S}^{(L)}$  of Proposition 3.2 were, in fact, optimal for  $\mathcal{P}_{(\text{ht})}^{(L)}$ , then by (3.4), the sequence of ratios of  $\mathcal{S}^{(L)}$ 's successive period-lengths would deviate very slightly (but in a computationally complicated way) from being constant. This suggests that a schedule whose period-lengths grow geometrically, with an appropriate constant interperiod ratio  $\alpha > 1$ , would have quite good expected work production—and would be computationally very simple. We now craft a family of such schedules, parameterized by the ratio  $\alpha$ , that verify this conjecture. (Implicit in our using a single interperiod ratio is the fact that the lifespan  $L$  for each BLA affects only a schedule's number of periods.) We shall see that, by choosing values of  $\alpha$  progressively closer to 1, one obtains schedules which produce progressively greater amounts of expected work.

### 4.2.1 An Upper Bound on Possible Work Production

Complicating our desire for an analytical comparison of the work productions of our parameterized schedules with the expected work production of the optimal schedule  $\mathcal{S}^{(L)}$  of Proposition 3.2 is the fact that we have no explicit expression for the latter quantity. We overcome this difficulty by instantiating Lemma 3.1 for  $\mathcal{P}_{(\text{ht})}^{(L)}$ , to obtain a *strict* upper bound on the maximum possible expected work production of any schedule for the life function. Since the upper bound we thus obtain is unattainable by any actual schedule, we henceforth term it the *ideal* work production.



**Proposition 4.1.** For any schedule  $S$  for the heavy-tailed BLA

$$\mathcal{P}_{(\text{ht})}^{(L)},^7$$

$$\begin{aligned} \text{Exp-Work}(S; \mathcal{P}_{(\text{ht})}^{(L)}) &< \int_0^L \mathcal{P}_{(\text{ht})}^{(L)}(t) dt \\ &= \left(1 + \frac{1}{L}\right) \ln(L+1) - 1. \end{aligned} \quad (4.4)$$

#### 4.2.2 A Parameterized Family

of Schedules  $\{S_{(\text{ht})}^{(L)}[\alpha]\}$  for  $\mathcal{P}_{(\text{ht})}^{(L)}$

The advertised family of schedules is defined as follows:

For any lifespan  $L$  and interperiod ratio  $\alpha > 1$ , the schedule  $S_{(\text{ht})}^{(L)}[\alpha]$  has  $m_L$  periods, where

$$m_L \stackrel{\text{def}}{=} \left\lceil \log_\alpha \left( \frac{\alpha-1}{c} L + \alpha \right) \right\rceil.$$

$S_{(\text{ht})}^{(L)}[\alpha]$ 's period-lengths,  $t_0, t_1, \dots, t_{m_L-1}$ , are specified as follows:

- for each  $k \in \{0, 1, 2, \dots, m_L - 2\}$ ,  $t_k \stackrel{\text{def}}{=} \alpha^{k+1} c$ ;
- $t_{m_L-1} \stackrel{\text{def}}{=} L - \sum_{k=0}^{m_L-2} t_k = L - \frac{c}{\alpha-1} (\alpha^{m_L} - \alpha)$ .

Before turning to our analytical assessment of the schedules in the family  $\{S_{(\text{ht})}^{(L)}[\alpha]\}$ , we should note that one can often improve the schedules' quality trivially. To wit, for simplicity, we have not taken steps to ensure that  $t_{m_L-1} \leq c$ . If  $t_{m_L-1}$ , as defined, exceeds  $c$ , then one can easily increase  $\text{Exp-Work}(S_{(\text{ht})}^{(L)}[\alpha]; \mathcal{P}_{(\text{ht})}^{(L)})$  by splitting the schedule's last period. Even without this improvement, though, schedule  $S_{(\text{ht})}^{(L)}[\alpha]$  has good expected work production, in the sense of the following theorem.

**Theorem 4.2.** For any fixed  $\varepsilon > 0$ , there exists a fixed  $\alpha > 1$  such that, for sufficiently large  $L$ ,  $\text{Exp-Work}(S_{(\text{ht})}^{(L)}[\alpha]; \mathcal{P}_{(\text{ht})}^{(L)})$  is within a factor  $(1 + \varepsilon)$  of ideal.

**Proof.** Invoking (2.2, 3.2), we find by direct calculation and standard estimates that, for any fixed constant  $c$ :

$$\begin{aligned} \text{Exp-Work}(S_{(\text{ht})}^{(L)}[\alpha]; \mathcal{P}_{(\text{ht})}^{(L)}) &= \frac{L+1}{L} \sum_{k=0}^{m_L-2} (\alpha^{k+1} - 1) \cdot \\ &\quad \left( \frac{(\alpha-1)c}{\alpha^{k+2}c - \alpha c + \alpha - 1} - \frac{c}{L+1} \right) \\ &= \frac{L+1}{L} \sum_{k=0}^{m_L-2} \left( \frac{\alpha-1}{\alpha} - O(\alpha^{-k}) - \frac{\alpha^{k+1}c - c}{L+1} \right) \\ &\geq \frac{\alpha-1}{\alpha} \cdot \frac{L+1}{L} \cdot \log_\alpha L - \log_\alpha c - O(1). \end{aligned} \quad (4.5)$$

When we write the final inequality in the chain (4.5) in the more perspicuous form

$$\begin{aligned} \text{Exp-Work}(S_{(\text{ht})}^{(L)}[\alpha]; \mathcal{P}_{(\text{ht})}^{(L)}) &\geq \frac{\alpha-1}{\alpha \ln \alpha} \left(1 + \frac{1}{L}\right) \ln L - \log_\alpha c - O(1), \end{aligned} \quad (4.6)$$

it becomes clear that we can make

$$\text{Exp-Work}(S_{(\text{ht})}^{(L)}[\alpha]; \mathcal{P}_{(\text{ht})}^{(L)})$$

arbitrarily close to an additive constant away from the upper bound in (4.1) by choosing  $\alpha$  appropriately close to 1 and letting  $L$  grow without bound.  $\square$

## 5 A SIMULATION-BASED ANALYSIS OF

$S_{(\text{ht})}^{(L)}$  AND  $\{S_{(\text{ht})}^{(L)}[\alpha]\}$

This section is devoted to comparing the schedules  $S_{(\text{ht})}^{(L)}$  and  $\{S_{(\text{ht})}^{(L)}[\alpha]\}$  via simulations, both against one another and against the ideal work production exposed by the upper bound in (4.4). We have a dual goal here. First, we wish to determine how the quality of these schedules depends on the specifics of the NOW architecture (determined by the communication overhead  $c$ ), the granularities of tasks in the workload, and the lifespan of the cycle-stealing opportunity. Second, we wish to determine the magnitude of the “eventually” promised in Theorem 4.2: How quickly do the schedules  $S_{(\text{ht})}^{(L)}[\alpha]$  approach the productivity of the optimal schedule  $S_{(\text{ht})}^{(L)}$ ? We base our comparisons on simulations; for each lifespan  $L$ , we use: the upper bound in (4.4) as the ideal work production, the optimal schedule of Proposition 3.2 to compute  $S_{(\text{ht})}^{(L)}$ 's (optimal) work production, and the recipe of Section 4.2.2 to compute the work production of the schedules  $S_{(\text{ht})}^{(L)}[\alpha]$ .

### 5.1 The Experimental Setup

We select wall-clock times for the communication overhead  $c$  and for the execution times of tasks via interpretive extrapolation from a variety of published sources and consultation with practitioners. We consider two values for the wall-clock time of the communication overhead  $c$ : 500 $\mu$ sec and 500msec. These values, which are consistent with a variety of current NOW architectures, reflect both the (in)efficiency of the NOW's network and the heaviness of the associated communication protocol. In either case, the size of  $c$  demands that tasks be rather “coarse-grained” (or, “compute-intensive”), in order for cycle-stealing to afford one performance advantages even approaching that of a (tightly coupled) multiprocessor. We explore the effect of task-granularity on cycle-stealing performance by considering three levels of compute-intensiveness that are “reasonable” for bag-of-tasks workloads: “somewhat coarse” tasks that take 1 second to complete,<sup>8</sup> “coarse” tasks that take 20 seconds to complete, and “very coarse” tasks that take

<sup>8</sup> See the beginning of Section 2.1 for a description of what it takes to “complete” a task.

<sup>7</sup> In  $x$  denotes the natural (base- $e$ ) logarithm of  $x$ .

TABLE 1  
Converting  $c$  and  $L$  from Wall-Clock Times to Work-Unit (WU) Values

Per-Task Clock-Time = 1 Work-Unit (WU)	WU range of $L$	Wall-clock Value of $c$	Work-unit Value of $c$
1 sec	180–36,000 WU	$500\mu\text{sec}$	0.0005000 WU
		500msec	0.5000000 WU
20 sec	9–1800 WU	$500\mu\text{sec}$	0.0000250 WU
		500msec	0.0250000 WU
1 min	3–600 WU	$500\mu\text{sec}$	0.0000083 WU
		500msec	0.0083333 WU

TABLE 2  
 $\mathcal{S}_{(\text{ht})}^{(L)}$ 's Expected Work Production as a Fraction of the Ideal

Comparing $\mathcal{S}_{(\text{ht})}^{(L)}$ with the ideal:			
How long does it take $\mathcal{S}_{(\text{ht})}^{(L)}$ to produce almost-ideal work-output?			
Wall-clock Value of $c$	Per-Task Clock-Time	EXP-WORK $(\mathcal{S}_{(\text{ht})}^{(L)}; \mathcal{P}_{(\text{ht})}^{(L)}) \div \text{Ideal}$	Required Lifespan
$500\mu\text{sec}$	1 minute	> 99%	< 3 minutes
	20 seconds	> 99%	< 3 minutes
	1 second	$\approx$ 99%	< 3 minutes
500msec	1 minute	> 90%	< 10 minutes
	20 seconds	$\approx$ 90%	< 30 minutes
	1 second	> 83%	< 10 hours

1 minute to complete. We let the lifespan  $L$  range over a spectrum of “reasonable” lifespans for cycle-stealing opportunities, from 3 minutes (a telephone call) to 10 hours (an overnight opportunity)—except for one experiment in which we assess how quickly the schedules  $\mathcal{S}_{(\text{ht})}^{(L)}[\alpha]$  approach their advertised asymptotic behavior. Since the mathematical study of earlier sections measures  $c$  and  $L$  in work-units rather than wall-clock time (cf. Section 2.1), we use the (common) wall-clock time of the tasks being sent to the “borrowed” workstation to convert units, thereby giving us access to the analytical bounds and recipes needed for our simulations. Table 1 illustrates the conversion.

## 5.2 Comparing $\mathcal{S}_{(\text{ht})}^{(L)}$ against the Ideal

Our simulation-based evaluation of the optimal schedule  $\mathcal{S}_{(\text{ht})}^{(L)}$  aims at determining how *closely* the schedule’s expected work production approaches the ideal of (4.4)—and how *quickly* it approaches that value. The results of this experiment are encapsulated in Table 2, whose approximate entries are rounded to two decimal places. The table reflects the strong impact of  $c$ ’s value on  $\mathcal{S}_{(\text{ht})}^{(L)}$ ’s relative productivity—which is not surprising given that the ideal work production is computed ignoring communication overhead (which means that no actual schedule can produce this amount of work!). The table also indicates that all of the considered task granularities are sufficiently coarse to amortize communication overhead quickly in NOWs that admit the smaller value of  $c$ , but that task

granularity is a much bigger concern in NOWs that require the larger value of  $c$ . This latter point is particularly evident when one compares  $\mathcal{S}_{(\text{ht})}^{(L)}$ ’s relative productivity with 20 second tasks and with 1 second tasks for the larger value of  $c$ . Both of the messages of this experiment are emphasized and refined by the results of the “reasonable” lifespan experiment of the next section.

## 5.3 Comparing $\{\mathcal{S}_{(\text{ht})}^{(L)}[\alpha]\}$ against the Ideal and the Optimal

Our simulation-based evaluation of the (asymptotically) good schedules  $\{\mathcal{S}_{(\text{ht})}^{(L)}[\alpha]\}$  resides in two complementary experiments. Our first experiment investigates how the expected work production of these schedules compares with *ideal* work production during our three “reasonable” lifespans. By including  $\mathcal{S}_{(\text{ht})}^{(L)}$ ’s (optimal) expected work production as a baseline throughout this experiment, we also garner information about how closely the  $\{\mathcal{S}_{(\text{ht})}^{(L)}[\alpha]\}$  approximate *optimal* performance during “reasonable” lifespans. Our second experiment investigates this approximation in a different way. We determine how quickly the expected work production of the best of the schedules  $\{\mathcal{S}_{(\text{ht})}^{(L)}[\alpha]\}$  approaches  $\mathcal{S}_{(\text{ht})}^{(L)}$ ’s expected work production. The fact of the approach is guaranteed by Theorem 4.2, but this guarantee is asymptotic: “as  $L$  grows without bound.” One gets some intuition for all of the comparisons of this section by considering how the performance of the

TABLE 3  
 $\mathcal{S}_{(ht)}^{(L)}$ 's and  $\mathcal{S}_{(ht)}^{(L)}[\alpha]$ 's Proximity to *Ideal* Work Production During “Reasonable” Lifespans

Comparing actual schedules against the ideal: What percentage of <i>ideal</i> work-output do $\mathcal{S}_{(ht)}^{(L)}$ and $\mathcal{S}_{(ht)}^{(L)}[\alpha]$ produce during “reasonable” lifespans?					
Wall-clock Value of $c$	Per-Task Clock-Time	“ratio (best value of $\alpha$ )” at:			
		3 minutes	10 minutes	1 hour	10 hours
500 $\mu$ sec	1 minute	.993(OPT)	.997(OPT)	.998(OPT)	.999(OPT)
		.989(1.010)	.992(1.008)	.994(1.005)	.996(1.004)
	20 seconds	.994(OPT)	.996(OPT)	.997(OPT)	.998(OPT)
.986(1.013)		.989(1.010)	.992(1.008)	.993(1.006)	
1 second	.987(OPT)	.989(OPT)	.991(OPT)	.993(OPT)	
	.969(1.027)	.973(1.024)	.977(1.020)	.980(1.017)	
500 msec	1 minute	.868(OPT)	.905(OPT)	.920(OPT)	.956(OPT)
		.783(1.201)	.837(1.180)	.884(1.099)	.913(1.079)
	20 seconds	.835(OPT)	.875(OPT)	.911(OPT)	.937(OPT)
.750(1.265)		.802(1.180)	.849(1.141)	.882(1.101)	
1 second	.668(OPT)	.723(OPT)	.782(OPT)	.831(OPT)	
	.594(1.412)	.648(1.282)	.706(1.241)	.755(1.200)	

$\{\mathcal{S}_{(ht)}^{(L)}[\alpha]\}$  depends (asymptotically) on the value of  $\alpha$ . When  $\alpha = 1.015$ , the leading term,  $(\alpha - 1)/(\alpha \ln \alpha)$ , in the bound (4.6) approaches 0.993; when  $\alpha = 1.01$ , the term is roughly 0.995. Thus, in either of these cases,

$$\text{Exp-Work}\left(\mathcal{S}_{(ht)}^{(L)}[\alpha]; \mathcal{P}_{(ht)}^{(L)}\right)$$

grows asymptotically (with  $L$ ) to within 1 percent of ideal work production.

**The “Reasonable”-Lifespan Experiment.** Table 3 illustrates the performance of the schedules  $\{\mathcal{S}_{(ht)}^{(L)}[\alpha]\}$  for our two values of the communication overhead  $c$ , three levels of task coarseness, and four “reasonable” lifespans. For each overhead-granularity-lifespan triple, we determine the value of  $\alpha$  that makes  $\mathcal{S}_{(ht)}^{(L)}[\alpha]$  perform best—measured by the ratio of

$$\text{Exp-Work}\left(\mathcal{S}_{(ht)}^{(L)}[\alpha]; \mathcal{P}_{(ht)}^{(L)}\right)$$

to ideal work production—and we present both the ratio and the value of  $\alpha$ . As before, the table entries must be viewed in the light of the unachievability of ideal work production by any actual schedule. We try to balance this unfairness by providing also a more realistic baseline in the table: we indicate how well the optimal schedule  $\mathcal{S}_{(ht)}^{(L)}$  performs under the same groundrules. The patterns one perceives in the table yield no surprises.

- For all overhead-granularity pairs, all schedules benefit from longer lifespans.
- For all overhead-lifespan pairs, all schedules benefit from coarser task granularity.
- The schedules  $\mathcal{S}_{(ht)}^{(L)}[\alpha]$  do not cope with communication overhead as well as  $\mathcal{S}_{(ht)}^{(L)}$  does. The latter’s advantage increases with the size of the overhead.

The good news from this experiment is that, for each tested combination of parameters, an appropriately chosen  $\mathcal{S}_{(ht)}^{(L)}[\alpha]$  performs quite well relative to  $\mathcal{S}_{(ht)}^{(L)}$ , approaching or exceeding 90 percent of the latter schedule’s work production. Somewhat tempering this good news is the fact that, for best performance, different combinations of parameters demand different values of  $\alpha$  (although discernible patterns can facilitate the search for the best  $\alpha$ ).

**The Race-against-Optimality Experiment.** Table 4 illustrates, for each of our communication overheads and each of our task granularities, how quickly the best of the schedules  $\{\mathcal{S}_{(ht)}^{(L)}[\alpha]\}$  comes close to  $\mathcal{S}_{(ht)}^{(L)}$ ’s expected work production. To be precise, the table indicates the smallest<sup>9</sup>  $L$  for which there exists an  $\alpha$  such that

$$\frac{\text{Exp-Work}\left(\mathcal{S}_{(ht)}^{(L)}[\alpha]; \mathcal{P}_{(ht)}^{(L)}\right)}{\text{Exp-Work}\left(\mathcal{S}_{(ht)}^{(L)}; \mathcal{P}_{(ht)}^{(L)}\right)} \geq z,$$

where  $z$  is the percentage we are identifying with “closeness,” and it presents such an  $\alpha$ . For our smaller value of  $c$ , we can insist that  $z = .99$  (so that “comes close” means “to within 1 percent”). For our larger value of  $c$ , this condition leads to unrealistically large lifespans, so we settle for  $z = .95$  (i.e., “to within 5 percent”); the one exception is with our finest grain tasks, where any value of  $z$  exceeding .9 leads to unrealistically large lifespans. Note that when lifespans start getting large in the table, we also indicate how close the schedules  $\{\mathcal{S}_{(ht)}^{(L)}[\alpha]\}$  come to  $\mathcal{S}_{(ht)}^{(L)}$ ’s expected work production within a short lifespan.

The messages of this experiment are consistent with those of the “reasonable lifespan” experiment: both coarser

9. We did not consider lifespans below three minutes.

TABLE 4  
The Lifespan Needed for  $\mathcal{S}_{(ht)}^{(L)}[\alpha]$  to Approach *Optimal* Work Production

Comparing $\mathcal{S}_{(ht)}^{(L)}[\alpha]$ with optimality:			
How long does it take $\mathcal{S}_{(ht)}^{(L)}[\alpha]$ to produce $\approx 99\%$ as much work as $\mathcal{S}_{(ht)}^{(L)}$ ?			
Wall-clock Value of $c$	Per-Task Clock-Time	Best $\alpha$	The “99% lifespan” for $\mathcal{S}_{(ht)}^{(L)}[\alpha]$
500 $\mu$ sec	1 minute	1.01	< 3 minutes
	20 seconds	1.013	< 3 minutes
	1 second	1.021 (1.027)	< 37 minutes (98% in < 3 minutes)
How long does it take $\mathcal{S}_{(ht)}^{(L)}[\alpha]$ to produce $\approx 95\%$ as much work as $\mathcal{S}_{(ht)}^{(L)}$ ?			
Wall-clock Value of $c$	Per-Task Clock-Time	Best $\alpha$	The “95% lifespan” for $\mathcal{S}_{(ht)}^{(L)}[\alpha]$
500msec	1 minute	1.10	$\approx$ 62 minutes
	20 seconds	1.087 (1.141)	< 34 hours (93% in < 1 hour)
	1 second	N/A (1.241)	> 10 <sup>50</sup> years (90% in < 1 hour)

We did not consider lifespans below three minutes.

tasks and longer lifespans help one amortize the effect of the significant communication overheads that one encounters in NOWs; larger communication overheads are harder to amortize.

## 6 CONCLUSIONS

We have achieved three major goals in this study:

1. We refined the Known-Risk cycle-stealing model of [5], [17], ending up with a model within which every cycle-stealing opportunity admits an optimal schedule; moreover—importantly—we have shown how to compute such a schedule.
2. We responded to the fact that optimal schedules for some cycle-stealing opportunities are computationally quite complex—and that, in particular, the important class of cycle-stealing opportunities whose durations come from a heavy-tailed distribution suffer from such complexity. We derived a parameterized family of easily computed schedules for heavy-tailed opportunities, which approach optimality asymptotically in the duration of the opportunity.
3. We performed a simulation-based study of heavy-tailed schedules, with the overall goal of garnering information that will help a potential cycle-stealer choose wisely those parameters of a cycle-stealing opportunity that one can control: the granularities of the tasks and the value of the parameter  $\alpha$  of the approximate schedules. We garnered this information via two experiments: 1) We determined how the quality of our schedules—both the optimal schedule  $\mathcal{S}_{(ht)}^{(L)}$  and the approximate, good ones  $\{\mathcal{S}_{(ht)}^{(L)}[\alpha]\}$ —depends on the variables that characterize the cycle-stealing opportunity: the communication overhead

of the host NOW, the granularities of the tasks that make up our bag-of-tasks workload, and the lifespan of the opportunity. 2) We determined the magnitude of the “eventually” promise that is inherent in asymptotic analyses: how quickly do all of our schedules approach ideal workload, and how quickly do our approximate schedules approach the productivity of the optimal schedule? We discovered that the optimal schedule produces an amount of work that is very close to ideal (within 10 percent) for all combinations of task granularities and lifespans that are reasonable in the light of the communication overhead that must be amortized. We discovered that we could make an only slightly weaker statement for appropriately chosen approximate schedules, although with a more restrictive definition of what is reasonable.

Several tempting challenges remain in this research area. How does one deal with task sizes that are known only approximately? How sensitive is an optimal schedule to the precise form of the life function  $\mathcal{P}$ ? Finally, how closely can one approximate an optimal schedule if one knows only one or two moments of the probability distribution of interrupts? The latter two questions are important since one would typically determine the “smooth” expression for  $\mathcal{P}$  by fitting a curve to sampled data.

## ACKNOWLEDGMENTS

This research was supported in part by US National Science Foundation Grants CCR-97-10367 and CCR-00-73401. A portion of this research was done while visiting the Department of Computer Science, The Technion; a portion of the writing was done while visiting the Laboratoire de Recherche en Informatique, Université Paris-Sud, under the support of a Fulbright Senior Scholar Grant and a grant from Centre Nationale de la Recherche Scientifique. It is a pleasure to thank: Lenny Heath and Ramesh Sitaraman for

helpful conversations about the mathematical development; Franck Cappello, Steve Dropsho, and Matt Thoennes for helpful conversations about parameter values in the simulations. A portion of this paper was presented at the 12th ACM Symposium on Parallel Algorithms and Architectures, Bar Harbor, Maine (2000).



**Arnold L. Rosenberg** received the AB in mathematics from Harvard College in 1962 and the AM and PhD in applied mathematics from Harvard University, in 1963 and 1966, respectively. He is a Distinguished University Professor of Computer Science at the University of Massachusetts (UMass) at Amherst, where he codirects the Theoretical Aspects of Parallel and Distributed Systems (TAPADS) Laboratory. Prior to joining UMass, he was a professor of

computer science at Duke University from 1981 to 1986 and a research staff member at the IBM Watson Research Center from 1965 to 1981. He has held visiting positions at Yale University and the University of Toronto; he was a Lady Davis Visiting Professor at the Technion (Israel Institute of Technology) in 1994 and a Fulbright Research Scholar at the University of Paris-South in 2000. His research focuses on theoretical aspects of parallel computing systems and communication networks, with emphasis on developing algorithmic techniques for designing better networks and architectures and using them more efficiently. He is the author of more than 135 technical papers on these and other topics in theoretical computer science and discrete mathematics. He is the coauthor of the new book, *Graph Separators, with Applications*, Kluwer Academic/Plenum Publishers. He is a fellow of the ACM, a fellow of the IEEE, a Golden Core member of the IEEE Computer Society, and a member of SIAM. He is the current conference chair of the ACM Symposium on Parallel Algorithms and Architectures and serves on several journal editorial boards. Information on his publications and other activities can be found at <http://www.cs.umass.edu/~rsnbrg/>.

## REFERENCES

- [1] T.E. Anderson, D.E. Culler, D.A. Patterson, and the NOW Team, "A Case for NOW (Networks of Workstations)," *IEEE Micro*, vol. 15, pp. 54-64, 1995.
- [2] M.J. Atallah, C.L. Black, D.C. Marinescu, H.J. Siegel, and T.L. Casavant, "Models and Algorithms for Coscheduling Compute-Intensive Tasks on a Network of Workstations," *J. Parallel Distributed Computing*, vol. 16, pp. 319-327, 1992.
- [3] B. Awerbuch, Y. Azar, A. Fiat, and F.T. Leighton, "Making Commitments in the Face of Uncertainty: How to Pick a Winner Almost Every Time," *Proc. 28th ACM Symp. Theory of Computing*, pp. 519-530, 1996.
- [4] M. Banikazemi and D.K. Panda, "Efficient Collective Communication on Heterogeneous Networks of Workstations," technical report, Ohio State Univ., 2000.
- [5] S.N. Bhatt, F.R.K. Chung, F.T. Leighton, and A.L. Rosenberg, "On Optimal Strategies for Cycle-Stealing in Networks of Workstations," *IEEE Trans. Computers*, vol. 46, pp. 545-557, 1997.
- [6] R. Blumofe and C.E. Leiserson, "Space-Efficient Scheduling of Multithreaded Computations," *Proc. 25th ACM Symp. Theory of Computing*, pp. 362-371, 1993.
- [7] R. Blumofe and C.E. Leiserson, "Scheduling Multithreaded Computations by Work Stealing," *Proc. 35th IEEE Symp. Foundations of Computer Science*, pp. 356-368, 1994.
- [8] R. Blumofe and D.S. Park, "Scheduling Large-Scale Parallel Computations on Networks of Workstations," *Proc. Third Int'l Symp. High-Performance Distributed Computing*, pp. 96-105, 1994.
- [9] F. Cappello, P. Fraigniaud, B. Mans, and A.L. Rosenberg, "HiHCoHP—Toward a Realistic Communication Model for Hierarchical Hyperclusters of Heterogeneous Processors," *Proc. Int'l Parallel and Distributed Processing Symp. (IPDPS '01)*, 2001.
- [10] W. Cirne and K. Marzullo, "The Computational Co-Op: Gathering Clusters into a Metacomputer," *Proc. 13th Int'l Parallel Processing Symp.*, pp. 160-166, 1999.
- [11] D. Culler, R.M. Karp, D. Patterson, A. Sahay, K.E. Schauser, E. Santos, R. Subramonian, T. von Eicken, "LogP: Towards a Realistic Model of Parallel Computation," *Comm. ACM*, vol. 39, pp. 78-85, 1996.
- [12] M. Harchol-Balter and A. Downey, "Exploiting Process Lifetime Distributions for Dynamic Load Balancing," *Conf. Measurement and Modeling of Computer Systems (SIGMETRICS '96)*, pp. 13-24, 1996.
- [13] W.E. Leland and T.J. Ott, "Load-Balancing Heuristics and Process Behavior," *Proc. Performance and ACM SIGMETRICS*, vol. 14, pp. 54-69, 1986.
- [14] M. Litzkow, M. Livny, and M.W. Mutka, "Condor—A Hunter of Idle Workstations," *Proc. Eighth Int'l Conf. Distributed Computing Systems*, pp. 104-111, 1988.
- [15] C.H. Papadimitriou and M. Yannakakis, "Towards an Architecture-Independent Analysis of Parallel Algorithms," *SIAM J. Computing*, vol. 19, pp. 322-328, 1990.
- [16] G.F. Pfister, *In Search of Clusters*. Prentice-Hall, 1995.
- [17] A.L. Rosenberg, "Guidelines for Data-Parallel Cycle-Stealing in Networks of Workstations, I: On Maximizing Expected Output," *J. Parallel and Distributed Computing*, vol. 59, pp. 31-53, 1999.
- [18] A.L. Rosenberg, "Guidelines for Data-Parallel Cycle-Stealing in Networks of Workstations, II: On Maximizing Guaranteed Output," *Int'l J. Foundations of Computer Science*, vol. 11, pp. 183-204, 2000.
- [19] A.L. Rosenberg, "Sharing Partitionable Workloads in Heterogeneous NOWs: Greedier Is not Better," *Proc. IEEE Int'l Conf. Cluster Computing*, pp. 124-131, 2001.
- [20] S.W. White and D.C. Torney, "Use of a Workstation Cluster for the Physical Mapping of Chromosomes," *SIAM NEWS*, pp. 14-17, Mar. 1993.

► For more information on this or any computing topic, please visit our Digital Library at <http://computer.org/publications/dlib>.