Optimal Schedules for Data-Parallel Cycle-Stealing in Networks of Workstations*

Arnold L. Rosenberg
Department of Computer Science
University of Massachusetts
Amherst, MA 01003, USA
rsnbrg@cs.umass.edu

April 17, 2000

Abstract

We refine the model underlying our prior work on scheduling cycle-stealing opportunities in networks of workstations [5, 16], obtaining a model wherein the scheduling guidelines of [16] produce optimal schedules for *every* such opportunity. We thereby render *pre*scriptive the *de*scriptive 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 always 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, 16]. Within our refined model, though, we derive computationally simple schedules for heavy-tailed opportunities, which can be "tuned" to have expected work-output that is arbitrarily close to optimal.

*A portion of this paper was presented at the *12th ACM Symp. on Parallel Algorithms and Architectures*, Bar Harbor, Me. (2000).

Keywords: Cycle-stealing, Data-parallel computations, 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 computation, the use of a network of workstations (NOW) as a parallel computer; cf. [1, 15]. Sources too numerous to list describe systems that facilitate the mechanics of NOW-based computing, often via the technique of *cycle-stealing*[1]—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 present 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 [16], in a way that allows one to devise schedules that maximize the expected work-output from *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 *pre*scriptive the *de*scriptive model of [5, 16]. We also consider the issue of deriving (nearly) optimal schedules efficiently.

1.1 Background

The model of [5, 16] 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. In this context, "relinquishing control immediately" implies killing any active job(s)—thereby losing all results since the last checkpoint.

Note. 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.

[1]As detailed imminently, we view cycle-stealing as an *adversarial* enterprise, wherein the "borrowed" workstation can be interrupted by the return of its owner. When implemented as a *cooperative* enterprise, cycle-stealing is often known as *work-sharing*.

This contract presents a challenging scheduling dilemma for the owner of workstation A. On the one hand, the typically large overhead required to set up an inter-workstation 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 for A, thereby killing all work done by B. The restraint studied in the *Known-Risk* model of [5, 16] 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* that embodies this probabilistic information—hence, governs the opportunity's duration—is "smooth." It is shown in [16] (cf. Theorem 2.1) that this simple model exposes constraints that any optimal[2] schedule must satisfy and that the guidelines that emerge from these constraints yield close-to-optimal schedules for large classes of cycle-stealing opportunities. Moreover, one sees from the examples in [5, 16] that one can often use situation-specific techniques to improve a guideline-prescribed schedule so that it is exactly optimal.

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 [16]. 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[3]—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.

[2] Throughout this study, a schedule's optimality is measured in terms of its expected work-output.

[3] [11, 12] discuss the occurrence of heavy-tailed distributions in important computational settings.

1.2 Our Main Results

In the present paper, we refine the Known-Risk model, obtaining a model within which the scheduling guidelines of [16] yield an optimal schedule for *every* cycle-stealing opportunity (Theorem 3.1). 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-optimizing 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).

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, 16, 17] 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-output. In [5, 16, 17], 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 work-output of a cycle-stealing opportunity. As noted earlier, the Known-Risk model of [5, 16] provides the starting point for our study; [17] develops the Guaranteed-Output model of [5], providing guidelines which optimize, to within low-order additive terms, the *guaranteed* work-output of 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, 18] 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 [18] 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. The study in [9] is concerned mainly with far-flung assemblages of NOWs, but its results are relevant to individual NOWs also. The study's focus, however, is on providing a "fair" allocation of resources to the members of its "Co-Op," (using a ticket-based resource-allocation scheme), rather than on optimizing either parallel speedup or work-output. Finally, one finds in [4] the beginnings of a model for scheduling collective communication in a heterogeneous NOW, similar to the communication-oriented portion of the model developed in [18]. Finally, the CILK system of [6, 7, 8] implements a work-stealing multithreading protocol wherein idle workstations borrow load asymptotically optimally, with respect to both speed of computation and space overhead.

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 [13] as an exemplar of the former type of study and to [19] as an exemplar of the latter.

2 Formal Background

2.1 The Known-Risk Model for Data-Parallel Cycle-Stealing

We focus on NOWs wherein a fixed overhead[4] c is incurred for setting up each pair of communications in which workstation A sends work to workstation B and B returns the results of that work to A. 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 schedules view tasks as indivisible; and, they assume 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).

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

[4]Our c plays the role of the communication-cost parameter of [14] and the "overhead" parameter o of [10].

an *(episode-)schedule* simply as a sequence of positive period-lengths: $\mathcal{S} = t_0, t_1, \ldots$. 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 $\mathcal{S}$ 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: (a) satisfies $\mathcal{P}(0) = 1$ (to indicate B's availability at the start of the episode); (b) when an upper bound L on the episode's lifespan ($\stackrel{\text{def}}{=}$ 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*:[5]

$$\text{MEAN-LIFESPAN}(\mathcal{P}) \ \stackrel{\text{def}}{=}\ -\int_0^U t\mathcal{P}'(t)\mathrm{d}t \ = \ \int_0^U \mathcal{P}(t)\mathrm{d}t. \qquad (2.1)$$

Note. The simplification of the integral in (2.1) results from the Known-Risk model's constraints on life functions, as just described.

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

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

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

$$T_i \ \stackrel{\text{def}}{=}\ t_0 + t_2 + \cdots + t_i;$$

the upper limit of the summation is the episode's lifespan L if it is finite, and is ∞ 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 summation (2.2) as one seeks good schedules.

[5] As usual, f' (resp., f'') denotes the first (resp., the second) derivative of the univariate function f.

Lemma 2.1 ([5, 16]) *One can effectively replace any schedule $\mathcal{S}$ for life function $\mathcal{P}$ by a schedule $\widehat{\mathcal{S}}$, each of whose periods—save the last, if $\widehat{\mathcal{S}}$ has finitely may periods—has length $> c$, such that* $\text{EXP-WORK}(\widehat{\mathcal{S}};\mathcal{P}) \geq \text{EXP-WORK}(\mathcal{S};\mathcal{P})$.

Proof Hint. One can never decrease the expected work-output of a schedule if one combines a "short" nonterminal period with its successor. ■

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

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

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

2.2 The Scheduling Guidelines of [16]

In [16], 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 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.

Henceforth, we say that a life function $\mathcal{P}$ is *concave* (resp., *convex*) if its second derivative is nonincreasing (resp., nondecreasing) for all t such that $\mathcal{P}(t) > 0$.

Theorem 2.1 ([16]) *If the productive schedule $\mathcal{S} = t_0, t_2, \ldots$ is optimal for the differentiable life function $\mathcal{P}$, then:*

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

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

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

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

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

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

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

Note 1. As is pointed out in [16], 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 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.

Note 2. It is shown in [16] that the guidelines systematically yield 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 [16] 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 3) opportunities whose durations are governed by the *heavy-tailed* life function

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

fall within this intransigent class.

Proposition 2.1 ([16]) *The heavy-tailed life function $\mathcal{P}_{(\mathrm{ht})}$ does not admit an optimal schedule.*

Proof Hint. One can always increase expected work-output by replacing any given schedule $\mathcal{S} = t_0, t_1, \ldots$ for $\mathcal{P}_{(\mathrm{ht})}$ by the schedule $\mathcal{S}^{(2)} \stackrel{\mathrm{def}}{=} 2t_0, 2t_1, \ldots$. ■

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

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

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

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. 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 $\mathcal{P}_{(\mathrm{ht})}$) convexity.

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

- For $\hat{L} \le L$, 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 $\mathcal{P}_{(\mathrm{ht})}(t)$ is

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

3.2 The Impact of BLAs on the Scheduling Guidelines of [16]

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 **(a)** *The difference in the system (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). The system thus becomes*

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

(b) *The recurrence of system (2.5) for the non-initial period-lengths of life function $\mathcal{P}^{(L)}$ is identical to the analogous recurrence for $\mathcal{P}$.*

(c) *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 (c), 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 (b) the term $(-\mathcal{P}(L))$ from the numerator of (3.1) also cancels out when instantiated in (2.5). ■

We now illustrate Proposition 3.1 by instantiating the guidelines of Theorem 2.1 for both $\mathcal{P}_{(\mathrm{ht})}$ and $\mathcal{P}^{(L)}_{(\mathrm{ht})}$. This is not an empty exercise, even though $\mathcal{P}_{(\mathrm{ht})}$ 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 $\mathcal{P}^{(L)}_{(\mathrm{ht})}$. Second, the "guidelines" for $\mathcal{P}_{(\mathrm{ht})}$ supply the inspiration for the computationally simple, provably good schedules for $\mathcal{P}^{(L)}_{(\mathrm{ht})}$ that we present in Section 4.2.

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

(a) *Letting $\widetilde{t}_i$ (resp., $\widetilde{T}_i$) ambiguously denote t_i and $t_i^{(L)}$ (resp., 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*

$$\widetilde{t}_{k+1} \;=\; (\widetilde{t}_k - c)\frac{\widetilde{T}_k + 1}{\widetilde{T}_{k-1} + c + 1}. \tag{3.4}$$

(b) *The initial period-length t_0 for $\mathcal{S}$ would be bounded as follows.*

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

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

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

3.3 The Universal Optimizability of BLA-Governed Opportunities

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 system (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-output that one can achieve during a finite-lifespan episode.

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

$$\text{EXP-WORK}(\mathcal{S}; \mathcal{P}) \ \le \ \text{MEAN-LIFESPAN}(\mathcal{P}).$$

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

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 $\mathcal{S} = t_0, t_1, \ldots, t_{m-1}$, 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 $m \in \{2, 3, \ldots, \lceil L/c \rceil\}$, the m-variable formal[6] work-function for $\mathcal{P}^{(L)}$:

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

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

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

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 $\langle \tau_0, \tau_1, \ldots, \tau_{m-1} \rangle$ defined by the three constraints:

- $\tau_{m-1} \geq 0$
- each other $\tau_i \geq c$ (3.6)
- $\tau_0 + \tau_1 + \cdots + \tau_{m-1} = L$,

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

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

which simultaneously

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

[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.

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:

$$\mathcal{W}_p^{(L)}(t_0^\star, t_1^\star, \ldots, t_{p-1}^\star) \;=\; \max_{m \in \{2,3,\ldots,\lceil L/c \rceil\}} \{\mathcal{W}_m^{(L)}(t_0, t_1, \ldots, t_{m-1}) \mid \text{ constraints (3.6) hold}\}.$$

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 $\widehat{\mathcal{S}^\star}$ whose expected work-output matches $\mathcal{S}^\star$'s. (Of course, $\widehat{\mathcal{S}^\star}$ may have fewer than p periods.)

Computing a productive optimal schedule. Finally, we invoke Theorem 2.1 to complete the proof. ■

We remarked in [16] 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 noncomprehensive 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 (Almost) Optimal 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 to optimal in expected work-output as desired.

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

(a) *for each nonterminal period-index i, $t_i \leq t_{i-1} - c$;*

(b) *the life function $\mathcal{P}$ has a bounded lifespan $L_{\mathcal{P}}$;*

(c) *schedule $\mathcal{S}$ has fewer than $\left\lceil \sqrt{\frac{2L_{\mathcal{P}}}{c} + \frac{1}{4}} + \frac{1}{2} \right\rceil$ periods.*

Note 1. The reader can easily adapt the proof of Lemma 4.1 to prove that, when $\mathcal{P}$ is *convex*, then each nonterminal $t_i \geq t_{i-1} - c$.

Note 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].

Note 3. The quantitative claims of Lemma 4.1 cannot be improved in general: the unique optimal schedule $\mathcal{S} = t_0, t_1, \ldots, t_{m-1}$ for the lifespan-L uniform-risk life function has $m = \left\lfloor \sqrt{\frac{2L_{\mathcal{P}}}{c} + \frac{1}{4}} + \frac{1}{2} \right\rfloor$ periods, and, for each nonterminal period-index i, $t_i = t_{i-1} - c$ [5].

Note 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. **(a)** We exploit the optimality of $\mathcal{S}$ only to infer that it is at least as productive as any of its *δ-perturbations*, $\mathcal{S}^{[i,\delta]} \stackrel{\text{def}}{=} t_0, t_1, \ldots, t_{i-1}, t_i+\delta, t_{i+1}-\delta, t_{i+2}, \ldots$. In other words, for every nonterminal period-index i and every real $\delta > 0$, the following difference is nonnegative:

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

After some calculation, we infer from this nonnegativity that:

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

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

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

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

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

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

(b) 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 (a)) with the fact that all of $\mathcal{S}$'s periods, save the last, have length $> c$ (Lemma 2.1).

(c) Part (a) implies that schedule $\mathcal{S}$ has some finite number $m \leq t_0/c$ periods. If we look at parts (a) and (b) "from the vantage point of t_{m-1}," we find that

$$L_{\mathcal{P}} \;=\; t_0 + t_1 + \cdots + t_{m-2} + t_{m-1} \;\geq\; m t_{m-1} + \binom{m}{2} c \;>\; \binom{m}{2} c.$$

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

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 BLAs' work-outputs 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 system (2.4).

Efficiently computing optimal schedules. Let $\mathcal{P}$ be an arbitrary concave life function, and let $\mathcal{S}^\star = t_0^\star, t_1^\star, \ldots, t_{m-1}^\star$ 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 $\mathcal{S}^\star$ satisfies system (2.4) if, and only if, every first partial derivative

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

of $\mathcal{W}_m$ *vanishes* at the point $\vec{t^\star} \stackrel{\text{def}}{=} \langle t_0^\star, t_1^\star, \ldots, t_{m-1}^\star \rangle$. Since $\mathcal{S}^\star$ has maximum expected work-output over all schedules for $\mathcal{P}$, we expect all of the second partial derivatives of $\mathcal{W}_m$ to be negative at point $\vec{t^\star}$. 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 $k < i$:

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

- For $k > i$:

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

- For $k = i$:

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

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 $\vec{t^\star}$, and since its second derivatives are always negative, we infer that $\vec{t^\star}$ is the *unique* maximum of $\mathcal{W}_m$. We now exploit this uniqueness to compute the point $\vec{t^\star}$, hence the desired optimal schedule $\mathcal{S}^\star$.

1. We invoke the inter-period dependencies for optimal schedules specified by system (2.5) to convert $\mathcal{W}_m(\tau_0, \tau_1, \ldots, \tau_{m-1})$ into a (formally, rather complex) function $\overline{\mathcal{W}}_m$ of the single variable τ_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 τ_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 $t_0^\star$ of schedule $\mathcal{S}^\star$.

4. We invoke system (2.5) again to determine all subsequent period-lengths of schedule $\mathcal{S}^\star$.

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 $\mathcal{S}^\star$. ■

4.2 Provably Good Schedules for Heavy-Tailed BLAs

This final 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-output 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 $\mathcal{P}^{(L)}_{(\mathrm{ht})}$, then by system (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 inter-period ratio $\alpha > 1$, would have quite good expected work-output—and would be computationally very simple. We now craft a family of such schedules, parameterized by the ratio α, that verify this conjecture. (Implicit in our using a single inter-period 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 α progressively closer to 1, one obtains schedules with progressively greater expected work-outputs.

We begin by instantiating Lemma 3.1 for $\mathcal{P}^{(L)}_{(\mathrm{ht})}$, thereby obtaining an upper bound on the maximum possible expected work-output of any schedule for $\mathcal{P}^{(L)}_{(\mathrm{ht})}$.

Proposition 4.1 *For any schedule $\mathcal{S}$ for the heavy-tailed BLA $\mathcal{P}^{(L)}_{(\mathrm{ht})}$,*[7]

$$\textsc{Exp-Work}\left(\mathcal{S}; \mathcal{P}^{(L)}_{(\mathrm{ht})}\right) \; < \; \int_0^L \mathcal{P}^{(L)}_{(\mathrm{ht})}(t)\mathrm{d}t \; = \; \left(1+\frac{1}{L}\right)\ln(L+1) - 1.$$

Next, we define our family of schedules. For any lifespan L and inter-period ratio $\alpha > 1$, the m_L-period schedule $\mathcal{S}^{(L)}_{(\mathrm{ht})}[\alpha] \; = \; t_0, t_1, \ldots, t_{m_L-1}$, where

$$\left(m_L \; \stackrel{\mathrm{def}}{=} \; \left\lfloor \log_\alpha \left(\frac{\alpha-1}{c}L + \alpha\right)\right\rfloor\right),$$

is defined as follows.

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

Note that, 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 $\textsc{Exp-Work}\left(\mathcal{S}^{(L)}_{(\mathrm{ht})}[\alpha]; \; \mathcal{P}^{(L)}_{(\mathrm{ht})}\right)$ by splitting the schedule's last period. Even without this improvement, though, schedule $\mathcal{S}^{(L)}_{(\mathrm{ht})}[\alpha]$ has good expected work-output.

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

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

[7] $\ln x$ denotes the natural logarithm of x.

$$
\begin{aligned}
&\text{Exp-Work}\left(\mathcal{S}^{(L)}_{(\mathrm{ht})}[\alpha];\ \mathcal{P}^{(L)}_{(\mathrm{ht})}\right) \\
&\quad = \frac{L+1}{L} \sum_{k=0}^{m_L-2} (\alpha^{k+1}-1)\left(\frac{(\alpha-1)c}{\alpha^{k+2}c-\alpha c+\alpha-1} - \frac{c}{L+1}\right) \\
&\quad = \frac{L+1}{L} \sum_{k=0}^{m_L-2} \left(\frac{\alpha-1}{\alpha} - O(\alpha^{-k}) - \frac{\alpha^{k+1}c-c}{L+1}\right) \qquad (4.4) \\
&\quad \geq \frac{\alpha-1}{\alpha}\cdot\frac{L+1}{L}\cdot\log_\alpha L - \log_\alpha c - O(1).
\end{aligned}
$$

When we write the final inequality in the chain (4.4) in the more perspicuous form

$$
\text{Exp-Work}\left(\mathcal{S}^{(L)}_{(\mathrm{ht})}[\alpha];\ \mathcal{P}^{(L)}_{(\mathrm{ht})}\right) \ \geq\ \frac{\alpha-1}{\alpha\ln\alpha}\left(1+\frac{1}{L}\right)\ln L - \log_\alpha c - O(1),
$$

it becomes clear that we can make $\text{Exp-Work}\left(\mathcal{S}^{(L)}_{(\mathrm{ht})}[\alpha];\ \mathcal{P}^{(L)}_{(\mathrm{ht})}\right)$ arbitrarily close to an additive constant away from the upper bound of Proposition 4.1 by choosing α appropriately close to 1. For instance, when $\alpha = 1.015$, $(\alpha-1)/(\alpha\ln\alpha) \approx 0.993$, so that, for sufficiently large L, $\text{Exp-Work}\left(\mathcal{S}^{(L)}_{(\mathrm{ht})}[\alpha];\ \mathcal{P}^{(L)}_{(\mathrm{ht})}\right)$ is within 1% of the optimal expected work-output of any schedule for $\mathcal{P}^{(L)}_{(\mathrm{ht})}$. ■

Acknowledgments. This research was supported in part by NSF Grant #CCR-97-10367. A portion of this research was done while visiting the Dept. of Computer Science, The Technion; a portion of the writing was done while visiting the Lab. de Recherche en Informatique, Univ. Paris-Sud, under the support of a Fulbright Senior Scholar Grant and a grant from CNRS. It is a pleasure to thank Lenny Heath and Ramesh Sitaraman for helpful conversations.

References

[1] T.E. Anderson, D.E. Culler, D.A.Patterson, and the NOW Team (1995): A case for NOW (networks of workstations). *IEEE Micro 15*, 54–64.

[2] 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.

[3] 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.

[4] M. Banikazemi and D.K. Panda (2000): Efficient collective communication on heterogeneous networks of workstations. Tech. Rpt., Ohio State Univ.

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

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

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

[8] 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.

[9] W. Cirne and K. Marzullo (1999): The Computational Co-Op: gathering clusters into a metacomputer. *13th Intl. Parallel Processing Symp.*, 160–166.

[10] D. Culler, R.M. Karp, D. Patterson, A. Sahay, K.E. Schauser, E. Santos, R. Subramonian, T. von Eicken (1996): LogP: towards a realistic model of parallel computation. *C. ACM 39*, 78–85.

[11] M. Harchol-Balter and A. Downey (1996): Exploiting process lifetime distributions for dynamic load balancing. *SIGMETRICS'96*, 13–24.

[12] W.E. Leland and T.J. Ott (1986): Load-balancing heuristics and process behavior. *Performance and ACM SIGMETRICS 14*, 54–69.

[13] M. Litzkow, M. Livny, M.W. Mutka (1988): Condor – A hunter of idle workstations. *8th Intl. Conf. on Distr. Computing Systs.*, 104–111.

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

[15] G.F. Pfister (1995): *In Search of Clusters*. Prentice-Hall.

[16] A.L. Rosenberg (1999): Guidelines for data-parallel cycle-stealing in networks of workstations, I: on maximizing expected output. *J. Parallel and Distr. Comput. 59*, 31–53.

[17] A.L. Rosenberg (2000): Guidelines for data-parallel cycle-stealing in networks of workstations, II: on maximizing guaranteed output. *Intl. J. Foundations of Computer Science*, to appear.

[18] A.L. Rosenberg (2000): Optimal sharing of partitionable workloads in heterogeneous networks of workstations. *Intl. Wkshp. on Cluster Computing – Technologies, Environments, and Applications (CC-TEA'2000).*

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