1999

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

Arnold L. Rosenberg
University of Massachusetts - Amherst

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

(Extended Abstract)

Arnold L. Rosenberg*
University of Massachusetts at Amherst

Abstract

We derive efficient guidelines for scheduling data-parallel computations within a draconian mode of cycle-stealing in networks of workstations wherein an interruption by the owner of the "borrowed" workstation kills all jobs currently in progress. We derive both adaptive and non-adaptive scheduling guidelines that maximize, up to low-order additive terms, the amount of work that one is guaranteed *to accomplish during a cycle-stealing opportunity, no matter when the opportunity is interrupted—up to a prespecified number of times.*

1. Cycle-Stealing in Clusters

Many 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; see, e.g., [8]. Numerous sources describe systems that facilitate the mechanics of NOW-based computing, especially via *cycle stealing*—the use by one workstation of idle computing cycles of another. However, few sources study the problem of scheduling individual computations on NOWs; even fewer present models that facilitate such scheduling for broad classes of computations. In the current paper, we refine the model developed in [3] and derive guidelines for crafting cycle-stealing schedules for data-parallel computations, which approximately maximize the amount of work that one is *guaranteed* to accomplish during a cycle-stealing opportunity.

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

> Such draconian contracts are inevitable when workstation B is a laptop that can be unplugged from the network. Such contracts are popular because of the degradation in service that B's owner receives when A's jobs remain active, even with lowered priority.

This contract creates the following challenge. On the one hand, the threat of losing work in progress when an interrupt occurs recommends breaking each cycle-stealing opportunity into many short "periods," supplying small amounts of work to B each time. On the other hand, the expensive setup time for the inter-workstation communications that bracket each period—to supply work to B and to reclaim the results of that work—recommends breaking each opportunity into a small number of long periods, supplying large amounts of work to B each time. In order to schedule a cycle-stealing opportunity productively, one must balance these conflicting factors judiciously.

The model in [3] is two-faceted, comprising one submodel that focuses on the *expected* work production of a cycle-stealing opportunity and one that focuses on the *guaranteed* work production. Recognizing that cycle-stealing can accomplish productive work only if the "malicious adversary" is restrained from interrupting every period when B is doing work for A, just before B returns its results, both submodels assume some idealized knowledge that restrains the adversary. The *guaranteed-output* submodel—our focus here—assumes that the owner of A knows both the total amount of time that B will be available and an upper bound on the number of possible interruptions.

We derived in [3] exactly optimal schedules for a small number of specific scenarios under each of the two submodels, using techniques that were specific to each scenario. In the current paper and its companion [9] (which focuses on the expected-output submodel), we have sought broadly applicable guidelines that produce nearly optimal schedules for large classes of scenarios.

*This research was supported in part by NSF Grant CCR-97-10367.
Address: Dept. of Computer Science, Univ. Massachusetts, Amherst, MA 01003; rsnbrg@cs.umass.edu

1.2. Our Main Results. In Section 3, we present two computationally efficient sets of scheduling guidelines: One set produces optimal *non-adaptive* schedules—that do not change their strategy until all possible interrupts have occurred; one set produces *adaptive* schedules—that change strategy in response to each interrupt—which are optimal to within low-order additive terms.

1.3. Related Work. The few algorithmic studies of cycle-stealing in the literature approach the scheduling problem rather differently. In [1], a cycle-stealing schedule within a NOW is crafted by "auctioning off" large identical chunks of a compute-intensive task. The companion papers [5, 6] schedule directed acyclic graphs on a NOW in a way that optimizes system time and space requirements. In [2], cycle-stealing is but one application of a theory of how to make random decisions better than by random choices, within a logarithmic factor of optimally. Finally, in [3, 9] and the current paper, cycle-stealing is viewed as a game against an adversary who seeks to minimize the work production of a cycle-stealing opportunity by "maliciously" interrupting the borrowed workstation.

2. A Formal Model

2.1. The General Framework. Our schedules operate in an "architecture-independent" fashion [7]: the single parameter c represents the cost of setting up the paired communications in which workstation A sends work to workstation B and B returns the results of the work. We assume that: tasks are indivisible; task times may vary but are known perfectly; the time allotted to a task includes the marginal cost of transmitting its input and output data.

> Our framework: (a) keeps c independent of the amount of data transmitted; (b) is consistent with both "pull"- and "push"-oriented scheduling philosophies.

We characterize a cycle-stealing opportunity via two quantities:

1. the *usable lifespan* $U > 0$: the number of time units during which workstation B will be available to A;
2. an upper bound p on the *potential* number of times that B's owner may *interrupt* the usable lifespan.

Although A's owner knows p, U, s/he does *not* know either the *actual* number $0 \le a \le p$ of interrupts that will occur or their placements. We can thus view a cycle-stealing opportunity as a sequence of $a+1$ *episodes* of unknown durations, during which workstation A has access to workstation B, punctuated by the a actual interrupts by B's owner.

2.2. Schedules and Work Production. The owner of A partitions each episode into *periods*, each of which begins with A sending work to B and terminates with B returning the results of that work. Since A's discretionary power resides solely in deciding how much work to send in each period, we may view an *episode-schedule* as a sequence of period-lengths: an m-period schedule for an episode of length $\le L$ (the current residual lifespan) thus has the form $\mathcal{S} = t_1, t_2, \ldots, t_m$, where each $t_i > 0$, and $\sum_{i=1}^{m} t_i = L$.

> Both m and the t_i are determined completely by the current values of L and p; therefore, we adorn all relevant schedule parameters with them (unless they are clear from context), e.g., denoting m by $m^{(p)}[L]$.

Period k of (the episode scheduled by) $\mathcal{S}$ begins at time

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

A supplies B with[1] $t_k \ominus c$ units of work, where t_k is chosen to allow sufficient time for A to send the work to B, and for B to perform the work and return its results to A.

Say that an episode begins with residual lifespan L. If B *is not interrupted* during period k (i.e., by time $T_k = \tau_k + t_k$) then the amount of work done during this episode is augmented by $t_k \ominus c$; if B *is interrupted* during period k, say at time[2] $t \in [\tau_k,\ T_k)$, then the episode ends having accomplished $\mathcal{W}(\mathcal{S}) = \sum_{i=1}^{k-1}(t_i \ominus c)$ units of work, with the new residual lifespan $L - t$.

Since episode-lengths are dictated by interrupts, A's owner has the choice of scheduling a cycle-stealing opportunity either *adaptively* or *non-adaptively*.

When proceeding *non-adaptively*, A's owner crafts a single episode-schedule $\mathcal{S} = t_1, t_2, \ldots, t_m$. If a given period, say the ith, is interrupted, then upon regaining control of B, A's owner employs the "tail" $t_{i+1}, t_{i+2}, \ldots, t_m$ of $\mathcal{S}$ for the remainder of the opportunity. The only exception to this "oblivious" behavior is that after the pth interrupt, A's owner schedules the remainder of the opportunity as one long period. If interrupts occur at the ends of periods in the set $I = \{i_1, i_2, \ldots, i_p\}$, then $\mathcal{S}$ achieves work $\mathcal{W}(\mathcal{S}) = \sum_{k \notin I}(t_k \ominus c) + ((U - T_{i_p}) \ominus c)$; the last term represents the last, "long" period.

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

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

where $L_1, L_2, \ldots, L_a$ are the lengths of the a interrupted episodes. The work achieved under Σ is the aggregate work

[1] "$\ominus$" denotes *positive subtraction*: $x \ominus y \stackrel{\text{def}}{=} \max(0, x - y)$.

[2] As usual, the assertion "$a \in [b,\ c)$" means "$b \le a < c$".

achieved under its constituent episode-schedules:

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

We denote by $\mathcal{W}^{(p)}[L]$ the maximum amount of work achievable by any adaptive schedule in an opportunity with (residual) lifespan L and p potential interrupts.

3. Guidelines for Nearly Optimal Schedules

3.1. Non-Adaptive Schedules. The p-interrupt nonadaptive schedule $\mathcal{S}_{\text{na}}^{(p)}[U] = t_1^{(p)}[U],\ t_2^{(p)}[U], \ldots,\ t_m^{(p)}[U]$ is specified as follows.

Schedule-length: $m^{(p)}[U] = \lfloor\sqrt{pU/c}\rfloor$.

Period-lengths: Each $t_i^{(p)}[U] = \sqrt{cU/p}$.

Analysis. The adversary will kill the *last* p periods of $\mathcal{S}_{\text{na}}^{(p)}[U]$ (at their last instant), for this maximizes the effect of the parameter c in diminishing work production. Under this strategy, $\mathcal{W}\left(\mathcal{S}_{\text{na}}^{(p)}\right) = U - \sqrt{2pcU} + pc + O(1)$, which elementary calculus shows cannot be improved.

3.2. Adaptive Schedules. We now present and begin to analyze our adaptive opportunity-schedule $\Sigma_{\text{a}}^{(p)}[U]$; the development in Sections 4 and 5 is needed to complete the analysis. $\Sigma_{\text{a}}^{(p)}[U]$ is obtained by adaptively invoking the sequence of episode-schedules: $\mathcal{S}_{\text{a}}^{(p)}[U],\ \mathcal{S}_{\text{a}}^{(p-1)}[U], \ldots,\ \mathcal{S}_{\text{a}}^{(0)}[U]$. The p-interrupt episode-schedule $\mathcal{S}_{\text{a}}^{(p)}[U] = t_1^{(p)}[U],\ t_2^{(p)}[U], \ldots,\ t_m^{(p)}[U]$ is specified as follows. For $p = 0$, $\mathcal{S}_{\text{a}}^{(0)}[U]$ has one period, of length U. For $p > 0$:

Schedule-length: $m^{(p)}[U] = \left\lfloor 2^{p-1/2}\sqrt{U/c}\right\rfloor + p2^{2p-1}$.

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

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

Simple calculation verifies that the indicated period-lengths are consistent with the number of periods.

4. Theoretical Underpinnings

We view cycle-stealing as a game against a "malicious adversary" who seeks to use the p available interrupts to minimize the work production of our cycle-stealing opportunity, even as we seek to maximize this production. The first move is ours. Based on the current values of L and p, we invoke episode-schedule $\mathcal{S}^{(p)}[L]$. As long as the adversary still has available interrupts (i.e., $p > 0$), s/he will decide either to let the current episode play out without an interrupt or to interrupt some period, thereby nullifying some of our usable lifespan. If s/he does interrupt us, say at time t of the current episode, then when we regain control of B, we invoke episode-schedule $\mathcal{S}^{(p-1)}[L - t]$. The game continues until $p = 0$, at which point the adversary cannot prevent our working until the end of the residual lifespan.

We approach this "game" via bootstrapping, always assuming, when constructing a schedule for the current p, that, for each residual lifespan L, we inductively have access to an optimal $(p-1)$-interrupt schedule. Prop. 4.1(d) allows us to start up, with an optimal 0-interrupt schedule.

4.1. Underlying Observations. We begin with four simple, yet useful, results about $\mathcal{W}^{(p)}[U]$ as a function of p, U.

Proposition 4.1 **(a)** *For all p, the function $\mathcal{W}^{(p)}[U]$ is nondecreasing with increasing U.*
(b) *For all U, the function $\mathcal{W}^{(p)}[U]$ is nonincreasing with increasing p.*
(c) *If the lifespan $U \le (p+1)c$, then $\mathcal{W}^{(p)}[U] = 0$.*
(d) *The unique optimal schedule for the case $p = 0$ is the 1-period schedule $\mathcal{S}_{\text{opt}}^{(0)}[U] = U$ which achieves $\mathcal{W}^{(0)}[U] = U - c$ units of work.*

Proof Sketch. **(a)** One can always append new periods to an existing schedule. **(b)** The adversary need not use all available interrupts. **(c)** The adversary can kill every productive period of such a short lifespan. **(d)** Obvious. ■

We see now that only some of the adversary's apparent options concerning interrupts are actually viable. We begin with a result that has three significant consequences: (1) narrowing our search for optimal schedules; (2) allowing us to use ordinary, rather than positive, subtraction when computing an episode's potential work; (3) leading to two significant observations about the adversary's strategy. The result shows that we may restrict attention to schedules that are *productive*, in the sense of having all period-lengths, save perhaps the last in each episode, strictly exceed c.[3]

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

Proof Sketch. If the ith nonterminal period of the kth constituent episode-schedule $\mathcal{S}_k$ of Σ is nonproductive, then one alters Σ by combining periods i and $i+1$ of $\mathcal{S}_k$. ■

We actually concentrate on episode-schedules that are *fully productive*, in the sense of having *all* period-lengths

[3] An analogous result is proved in [3] for the expected-output submodel.

$> c$. This focus makes sense heuristically (how can a short period help?) but we have not yet been able to justify it rigorously. This focus gives us easy access to the immediate consequences of each of the adversary's $m^{(p)} + 1$ apparent options, which are enumerated in Table 1.[4]

Observations. **(a)** *The adversary will always interrupt a period at its last instant, thereby nullifying a full $t_k^{(p)}[U]$ time units of usable lifespan.*
(b) *The adversary will always interrupt every episode of a cycle-stealing opportunity, as long as $p > 0$ and $U > c$.*
(c) *When the adversary has $p \geq 1$ potential interrupts left, s/he will always interrupt an episode of lifespan $U > (p+1)c$ during a period that begins at some time $t < U - pc$.*

In the light of Proposition 4.1: Observation (a) infers the global information in Table 1 from the local information in the table, thereby yielding Observation (b). The proviso "$p > 0$" in (b) means that the adversary has available interrupts; the proviso "$U > c$" means that the episode can achieve work, hence is worth interrupting. Finally, Observation (b) implies Observation (c).

We now establish the computational significance of the Observations. The m-period episode-schedule $\mathcal{S}^{(p)}[U]$ is *r-immune*, where $r \in \{0, 1, \ldots, m-1\}$, if the adversary will never interrupt a period whose index *exceeds* $m - r$.

Theorem 4.2 *For any m-period r-immune episode-schedule $\mathcal{S}^{(p)}[U]$, one can set each period-length $t_i^{(p)}[U]$ for $i \geq m - r$, within the range $(c,\ 2c]$ without decreasing the schedule's work-production.*

Proof Sketch. Splitting a long period $\ell \geq m - r$ into two equal-length periods can only increase work production.[5] ■

4.2. Characterizing Optimal Adaptivity. Once we understand the adversary's options, we counteract his/her "moves" by having $\mathcal{S}^{(p)}[U]$ *equalize* the impacts of all potential interruptions. This strategy yields the following partial specification of the period-lengths of the optimal episode-schedule.

Theorem 4.3 *The period-lengths of the optimal episode-schedule $\mathcal{S}_{\text{opt}}^{(p)}[U]$ for a cycle-stealing opportunity with usable lifespan U and $\leq p$ interrupts satisfy the following system of equalities. If $p = 0$, then $m^{(p)}[U] = 1$, and $t_1^{(p)}[U] = U$. If $p > 0$, then, letting ℓ_p be the smallest period-index k for which $U - T_{k-1}^{(p)} > pc$:*

- $t_k^{(p)} = c + \mathcal{W}^{(p-1)}[U - T_k^{(p)}] - \mathcal{W}^{(p-1)}[U - T_{k+1}^{(p)}]$ *for $1 \leq k \leq \ell_p - 2$;*
- $t_{\ell_p - 1}^{(p)} = c + \mathcal{W}^{(p-1)}[U - T_{\ell_p - 1}^{(p)}]$;
- $t_k^{(p)} = c + \alpha \in (c,\ 2c]$ *for* $\ell_p \leq k \leq m$.

[4] Tables appear at the end.

[5] Easily, this reasoning applies only to the *last* period of an episode.

Proof Sketch. The high-index $t_k^{(p)}$ follow from Theorem 4.2; all others result from our strategy and Table 1. ■

5. From Underpinnings to Guidelines

We now analyze $\Sigma_{\text{a}}^{(p)}[U]$ and its constituent $\{\mathcal{S}_{\text{a}}^{(p)}[U]\}$: in an absolute sense—by estimating guaranteed work production—and in a relative sense—using the abstract guidelines implicit in Section 4 as our baseline. We construct the optimal episode-schedule $\mathcal{S}_{\text{opt}}^{(1)}[U]$ and compare $\mathcal{W}^{(1)}[U]$ with $\mathcal{W}(\Sigma_{\text{a}}^{(1)}[U])$ and with $\mathcal{W}(\Sigma_{\text{a}}^{(p)}[U])$, discovering thereby that $\mathcal{W}(\Sigma_{\text{a}}^{(p)}[U])$ deviates from optimality by only low-order additive terms.

5.1. The Guaranteed Work-Production of $\Sigma_{\text{a}}^{(p)}[U]$.

Theorem 5.1 *For all $p \geq 0$, schedule $\Sigma_{\text{a}}^{(p)}[U]$ accomplishes*

$$\mathcal{W}(\Sigma_{\text{a}}^{(p)}[U]) \ \geq\ U - (2 - 2^{1-p})\sqrt{2cU} - O\left(U^{1/4} + pc\right)$$

units of work, which is optimal to within low-order additive terms.

Proof Sketch. The bound on $\mathcal{W}(\Sigma_{\text{a}}^{(p)}[U])$ follows by induction; its near-optimality follows from Section 5.2. ■

5.2. Comparing $\Sigma_{\text{a}}^{(p)}[U]$ against Optimality. We apply the abstract guidelines of Section 4 to derive $\mathcal{S}_{\text{opt}}^{(1)}[U]$.[6] Importantly, this one case shows that all $\Sigma_{\text{a}}^{(p)}[U]$ are within a low-order additive term of optimality.

The period-lengths of $\mathcal{S}_{\text{opt}}^{(1)}[U]$. Since the case $p = 1$ is 0-immune, there exists $\alpha \in (0, 1]$ such that:

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

The optimal m. Since the t_i sum to L, we have $\alpha = (U - c)/mc - \frac{1}{2}(m-1)$. By the bounds on α and the integrality of m, therefore:

$$m^{(1)}[U] \ =\ \left\lceil \sqrt{\left(\frac{2U}{c} - \frac{7}{4}\right)} - \frac{1}{2} \right\rceil . \qquad (5.1)$$

Explicit (approximately) optimal parameters. We can use (5.1) to determine optimal values for α, the $t_k^{(1)}[U]$, and $\mathcal{W}^{(1)}[U]$. We must settle for approximate values because of the complicated form of $m^{(1)}[U]$ in (5.1), coupled with the

[6] A similar derivation appears in [3] using a rather different analysis.

broad range of relevant U, which precludes a simple asymptotic analysis. (U is commensurate with c toward the end of an episode.) These approximate values are summarized in Table 2, along with the analogous parameters of $\mathcal{S}_{\mathrm{a}}^{(1)}[U]$ for comparison.

References

[1] 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 Distr. Comput.*, 16:319–327, 1992.

[2] 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. *28th ACM Symp. on Theory of Computing*, pages 519–530, 1996.

[3] 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. Comp.*, 46:545–557, 1997.

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

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

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

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

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

[9] A.L. Rosenberg. Guidelines for data-parallel cycle-stealing in networks of workstations, I: on maximizing expected output. *12th IEEE Intl. Parallel Proc. Symp.*, 1998, 519–523.

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

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

Table 2. *Parameter values for the case* $p = 1$:

Parameter	Approximate Value for $\mathcal{S}_{\mathrm{opt}}^{(1)}$	Value for $\mathcal{S}_{\mathrm{a}}^{(1)}[U]$
$m^{(1)}[U]$	$\sqrt{2U/c - 7/4}$	$\lfloor\sqrt{2U/c} + 2\rfloor$
α	$1/2$	N/A
$t_k^{(1)}[U]\ (1 \le k \le m-2)$	$\sqrt{2cU} - kc$	$\sqrt{2cU} - (k - 7/2)c$
$t_m^{(1)}[U] \;=\; t_{m-1}^{(1)}[U]$	$3c/2$	$3c/2$
$\mathcal{W}^{(1)}[U]$	$U - \sqrt{2cU} - c/2$	$U - \sqrt{2cU} - O(U^{1/4} + c)$