SIAM J COMPUT
Vol 16, No 1, February 1987

ON STOCHASTIC SCHEDULING WITH IN-TREE PRECEDENCE CONSTRAINTS*

CHRISTOS H. PAPADIMITRIOU† AND JOHN N. TSITSIKLIS‡

Abstract. We consider the problem of optimal scheduling of a set of jobs obeying in-tree precedence constraints, when a number M of processors is available. It is assumed that the service times of different jobs are independent identically distributed random variables. Subject to a minor assumption on the service time distribution, we show that policies of the "Highest Level First" type are optimal asymptotically, as the number of jobs tends to infinity.

Key words. scheduling, stochastic, trees

AMS(MOS) subject classification. 90B35

1. Introduction. Scheduling jobs with equal execution times and in-tree precedence constraints to minimize makespan (latest finishing time) is a classical problem in scheduling [Co]. It is known that, for any number of processors, the most natural policy, namely the one that asigns to the next available processor a leaf that has the largest height (distance from the root), is optimal [Hu]. This policy is called *highest level first.*

An interesting twist on this theme is to allow the execution times of the tasks to be independent and identically distributed exponential random variables [CR]. The results here are less complete: It is known that the highest level first policy is optimal *only in the two-processor case* [CR]. In fact, for two processors this policy is optimal under a wide variety of criteria [BR]. Unfortunately, for three processors the highest level first policy is not optimal, and no tractable way to approach this problem is known. It was shown in [Pa] that a generalization of the two-processor problem in another direction (and-or precedences) is PSPACE-complete.

In this paper we show that for any number of processors the highest level first policy, although suboptimal, is not much worse than the optimum. In particular, we show that, for any in-tree, the cost associated with a highest level first policy is no larger than the optimal cost times a factor that goes to one as N increases to infinity. Moreover, this result is shown to be true for a fairly wide class of service time distributions, exponential distributions being a special case.

2. Problem definition. We are given a set of M processors and an in-tree G with N nodes. Each node represents a job that may be processed by any of the processors. We assume that the jobs have service times that are independent and identically distributed random variables, with known distributions. We assume that these random variables are positive, with probability one.

A *scheduling policy* is a rule that, at time $t=0$, assigns L of the processors to L leaves of the tree, where L is the minimum of M and the number of leaves of G. If at time t some processor terminates the processing of a job, we delete the corresponding

* Received by the editors June 24, 1985; accepted for publication (in revised form) April 24, 1986 This research was supported by a National Science Foundation grant, an IBM Faculty Development Award and U.S. Army Research Office contract DAAG-29-84-K-0005

† Departments of Computer Science and Operations Research, Stanford University, Stanford, California 94305.

‡ Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139.

leaf. Then, if there still exists a leaf with no processor assigned to it, the scheduling policy reassigns the free processor to one such leaf. The choice of the leaf to which a processor is assigned may depend on all events that have occurred up to that time. For example, it may depend on the amount of time that the other processors have been already processing the jobs to which they have been assigned.

In the above description of scheduling policies we have implicitly imposed the following restrictions:

(a) No processor remains unassigned as long as there exists a leaf with no processor assigned to it;

(b) Once a processor is assigned to a leaf, it remains assigned to it until the processing of the corresponding job is completed. (Thus, preemptive policies are not allowed.)

The above restrictions are only introduced for definiteness and to keep the description of scheduling policies simple. However, our result does not change if these restrictions are altered.

For any policy π and initial in-tree G, we define $T_\pi(G)$ to be the time that the processing of the root of G is completed. We are interested in policies that minimize $T_\pi(G)$, maybe approximately.

A special (and very easily implementable) class of policies is the class of highest level first policies. With such policies, processors are always assigned to leaves of maximum distance from the root. (Notice that there exists more than one highest level first policy; a particular one could be chosen by following a lexicographic tie-breaking rule, for example.)

It is known that highest-level-first policies are not optimal for $M>2$ [CR], even if the service time distribution is exponential. However, we will show that they come within an arbitrarily small percentage of the optimal, as the size of the original tree grows to infinity.

We assume that there exists some $\eta>0$ such that $E[e^{\eta X}]<\infty$, where X is sampled from the service time distribution. Notice that this assumption allows both discrete and continuous service time distributions; exponential distributions are covered in particular. Under the above assumption, we have the following result:

THEOREM. *There exists some function $\beta:\{1,2,\cdots\}\mapsto[0,\infty)$ such that $\lim_{N\to\infty}\beta(N)=0$ and such that for any in-tree G and any highest level first policy H we have*

$$E[T_H(G)]\leqq\inf_\pi E[T_\pi(G)](1+\beta(N)),$$

where $N=|G|$ and where the infimum is taken over all scheduling policies π.

We have not been able to prove that a bound of this form is tight, and we are tempted to conjecture the stronger result that there exists a constant A such that, for any in-tree G,

$$E[T_H(G)]\leqq\inf_\pi E[T_\pi(G)]+A.$$

3. Proof of the theorem. Let us denote by $\mathcal{U}$ the set of functions $f:[0,\infty)\mapsto[0,\infty)$ with the property that $\lim_{a\to\infty}a^kf(a)=0$, $\forall k\geqq0$. (Negative exponentials are prototype elements of this class.) We develop a few useful consequences of our assumption on the service time distribution.

We first notice that, by Jensen's inequality, $E[X]\leqq(1/\eta)\log E[e^{\eta X}]<\infty$. We will therefore assume, without any loss of generality, that the mean service time $E[X]$ equals 1. We also have

LEMMA 1. *For any $\varepsilon>0$ there exist some $h_1^\varepsilon,\cdots,h_4^\varepsilon\in\mathcal{U}$ such that*

(i) $P(|\sum_{i=1}^{N} X_i - N| \geqq \varepsilon D) \leqq h_1^{\varepsilon}(D) \quad \forall D \geqq N$,

(ii) $E[\sum_{i=1}^{N} X_i; \sum_{i=1}^{N} X_i \geqq N + \varepsilon D] \leqq h_2^{\varepsilon}(D) \quad \forall D \geqq N$,[1]

(iii) $P(\max_{1 \leqq i \leqq N} X_i \geqq \varepsilon N^{1/2}) \leqq h_3^{\varepsilon}(N) \quad \forall N$,

(iv) $E[\max_{1 \leqq i \leqq N} X_i] \leqq \varepsilon N + h_4^{\varepsilon}(N) \quad \forall N$,

where $X_1, \cdots, X_N$ are independent random variables sampled from the service time distribution.

Proof. (i) This a classical result, but for completeness we outline a proof. One first shows that, given any $\varepsilon > 0$, we may choose $\eta > 0$ small enough so that $E[\exp\{\eta(X_i - 1 - (\varepsilon/2))\}] \leqq 1$. (A proof may be found in [Ha], for example.) It then follows that $E[\exp\{\eta(\sum_{i=1}^{N} X_i - N(1+(\varepsilon/2))\}] \leqq 1$ and $P(\sum_{i=1}^{N} X_i \geqq N + \varepsilon D) \leqq P(\sum_{i=1}^{N} X_i \geqq N + (\varepsilon/2)N + (\varepsilon/2)D) \leqq \exp\{-D\eta(\varepsilon/2)\}$. The inequality for the lower bound on $\sum_{i=1}^{N} X_i$ is proved similarly.

(ii) We have that $E[\sum_{i=1}^{N} X_i; \sum_{i=1}^{N} X_i \geqq N + \varepsilon D] \leqq \sum_{k=1}^{\infty} (D + \varepsilon kD + 1) P(\sum_{i=1}^{N} X_i \geqq N + \varepsilon kD) \leqq \sum_{k=1}^{\infty} (2D + \varepsilon kD) h_1^{\varepsilon}(kD) = h_2^{\varepsilon}(D)$, where h_1^{ε} is the function of part (i). We now show that $h_2^{\varepsilon} \in \mathcal{U}$. For any $m > 0$, we can find some A such that $h_1^{\varepsilon}(kD) \leqq A/(kD)^{m+3}$, for all k, D, such that kD is large enough, because $h_1^{\varepsilon} \in \mathcal{U}$. Then, $D^m h_2^{\varepsilon}(D) \leqq A(1+\varepsilon) \sum_{k=1}^{\infty} (kD)^{-2} = A(2+\varepsilon)\pi^2/(6D^2)$, which goes to zero, as $D \to \infty$. which goes to zero, as $D \to \infty$.

(iii) Let $C = E[\exp\{\eta X_i\}]$. Then, $P(X_i \geqq a) \leqq C e^{-\eta a}$, for all a and

$$P(\max_{1 \leqq i \leqq N} X_i \geqq a) \leqq NC e^{-\eta a}.$$

The result follows by taking $a = \varepsilon N^{1/2}$ and $h_3^{\varepsilon}(N) = CN \exp\{-\varepsilon\eta N^{1/2}\}$, which belongs to $\mathcal{U}$.

(iv) This follows from part (iii) in the same way that part (ii) followed from part (i). □

To simplify notation, we define $f_1^{\varepsilon} = \max_{1 \leqq i \leqq 4} \{h_i^{\varepsilon}\}$, which clearly belongs to $\mathcal{U}$.

We define the *depth* of a leaf of an in-tree to be the number of nodes on the path from the leaf to the root of the in-tree. We then define the depth of an in-tree as the maximum of the depths of all its leaves. We start by establishing the performance of scheduling policies for "thin" trees.

LEMMA 2. *For any $\varepsilon > 0$, there exists some $f_2^{\varepsilon} \in \mathcal{U}$ with the following property: Let G be any in-tree with M leaves and let $R_1, \cdots, R_M$ be the service times of its leaves. Then, $P(T_\pi(G) \geqq D + \varepsilon D' + \max_{1 \leqq i \leqq M} R_i | R_1, \cdots, R_M) \leqq f_2^{\varepsilon}(D')$, for all $D' \geqq D$, where D is the depth of G and π is a scheduling policy using M processors. (In fact there exists only a single scheduling policy in this case.) Also, $E[T_\pi(G) | R_1, \cdots, R_M] \leqq \max_{1 \leqq i \leqq M} R_i + (1+\varepsilon)D + f_2^{\varepsilon}(D)$.*

Proof. If $T_\pi(G) > D + \varepsilon D' + \max_{1 \leqq i \leqq M} R_i$, then there exists a path from some leaf to the root such that the sum of the service times of the jobs on this path (the leaf being excluded) is larger than or equal to $D + \varepsilon D'$. By Lemma 1(i), the probability of this event is bounded by $f_1^{\varepsilon}(D')$, for any particular path. There is a total of M candidate paths, which proves the first part of the lemma with $f_2^{\varepsilon}(D') = M f_1^{\varepsilon}(D')$. The second part of the lemma follows by appealing to Lemma 1(ii) to bound the contribution of the "unlikely" events to $E[T_\pi(G)]$. □

The main idea of the rest of the proof is the following: if the deepest part of an in-tree has very few leaves, then Lemma 2 will imply that a highest level first policy will be reducing its depth at approximately the largest possible rate of one unit per unit time, with high probability. If on the other hand the deepest part of an in-tree is

[1] If X is a random variable with distribution $P(\cdot)$ and A is an event, we use $E[X; A]$ to denote $\int_A X dP(X)$.

"thick," we will show that no policy can gain a substantial advantage over a highest level first policy. In order to put these two arguments together, we need to define a sequence of time intervals during which the deepest part of the remaining in-tree is either "thin" or "thick."

We will be assuming throughout that M is fixed and we introduce some new notation. Let us fix a highest level first policy H and another policy π. Given an initial in-tree G, we let $G_H(t)$, $G_\pi(t)$ be the (random) in-trees remaining at time t, if policy H, π, respectively, is followed. Let $D_H(t)$, $D_\pi(t)$ be the depth of $G_H(t)$, $G_\pi(t)$, respectively. Finally, let $d_H(t)$ be the smallest depth of any leaf to which a processor is assigned by policy H, at time t, in the graph $G_H(t)$. Notice that $G_H(t)$ has fewer than M leaves of depth larger than $d_H(t)$. (To resolve any ambiguity, we assume that all of the above introduced functions are right-continuous in t.) Also, for any processor k that is processing some job at time t, let $R_k(t)$ be the remaining time until the processing of that job is completed. Let $R^* = \max_{k,t} R_k(t)$. Notice that R^* equals the maximum processing time of any job. Thus, by virtue of Lemma 1 (iii)-(iv), we have $P(R^* \geqq \varepsilon N^{1/2}) \leqq f_1^\varepsilon(N)$ and $E[R^*] \leqq \varepsilon N + f_1^\varepsilon(N)$, for all $\varepsilon > 0$.

Given some $\varepsilon > 0$, we define a finite sequence $t_0 \leqq t_1 \leqq \cdots \leqq t_K$ of random times, (where K is a random integer to be defined below), and a corresponding sequence of random variables χ_k (for $k < K$), as follows. Let $t_0 = 0$ and suppose that t_k has been already defined. If $D_H(t_k) < N^{1/2}$ or if $G_H(t_k)$ has no more than M leaves, we let $K = k$ and the construction is completed. Otherwise, we consider two cases: (i) If $D_H(t_k) - d_H(t_k) \geqq N^{1/2}$, we let $\chi_k = 1$ and $t_{k+1} = t_k + (1+\varepsilon)N^{1/2}$. (ii) If $D_H(t_k) - d_H(t_k) < N^{1/2}$, we let $\chi_k = 0$ and we define t_{k+1} as the first time $t > t_k$ that $D_H(t) < N^{1/2}$, or the number of leaves of $G_H(t)$ is no larger than M or $D_H(t) - d_H(t) \geqq N^{1/2}$, whichever comes first. We define, for $k \leqq K$, $\Delta_k = D_H(t_k) - D_\pi(t_k)$. (Notice that the above introduced variables all depend on ε, even though this is not explicit in our notation.) We finally define t^* as the first time that $G_H(t)$ has M or less leaves.

It will be convenient to assume that both stochastic processes $G_H(t)$ and $G_\pi(t)$ are defined on a common probability space as follows. We start with a collection $\mathscr{X} = \{X_n^i : n = 1, 2, \cdots; i = 1, \cdots, M\}$ of independent random variables drawn from the service time distribution. We interpret X_n^i as the time spent by processor i to process the nth job ever assigned to that processor. With this definition, each of $G_H(t)$, $G_\pi(t)$ is a well-defined functional of the collection $\mathscr{X}$ of random variables. With this correspondence we have the following useful properties.

LEMMA 3. *If $G_H(s)$ has at least M leaves for all times $s < t$ (and in particular if $t \leqq t^*$) then*

(i) $|G_H(t)| \leqq |G_\pi(t)|$,

(ii) $D_\pi(t) \geqq d_H(t)$.

Proof. (i) Simply notice that each time that a node is deleted from G_π, a node is also deleted from G_H, provided that G_H had at least M leaves just before time t, which we are assuming.

(ii) Let A be the number of nodes of $G(0)$ at depth smaller than $D_H(t)$. If $D_\pi(t) < d_H(t)$, then $|G_\pi(t)| \leqq A$. On the other hand, under policy H, none of the jobs at depth smaller than $d_H(t)$ has been processed and there exists at least one unprocessed job at depth $d_H(t)$. Thus, $|G_H(t)| > A$ and the result follows from part (i). □

LEMMA 4. *There exists some $f_4 \in \mathscr{U}$ such that $P(T_H(G) > 2N) \leqq f_4(N)$.*

Proof. $P(T_H(G) > 2N) \leqq P(\sum_{i=1}^N X_i \geqq 2N) \leqq f_1^1(N)$, where each X_i is sampled independently from the service time distribution and where $f_1^1 \in \mathscr{U}$ is the function of Lemma 1. □

LEMMA 5. *For any $\varepsilon > 0$, there exists some $f_5^\varepsilon \in \mathscr{U}$ such that $P(K > 5N^{1/2}) \leqq f_5^\varepsilon(N)$.*

Proof. Notice that if $K > k+2$, then $t_{k+2} - t_k \geqq N^{1/2}$. Thus, if $K > 5N^{1/2}$, then $T_H(G) \geqq t_K \geqq 2N$, and the result follows from Lemma 4. □

LEMMA 6. *For any $\varepsilon > 0$, there exists some $f_6^\varepsilon \in \mathscr{U}$ such that $P(D_H(t_k) - D_H(t_{k+1}) \leqq N^{1/2}$ and $\chi_k = 1) \leqq f_6^\varepsilon(N)$ for all k.*

Proof. If $\chi_k = 1$, then the portion of $G_H(t_k)$ which lies at depth larger than $D_H(t_k) - N^{1/2}$ has at most M leaves. Therefore, for any time $t \geqq t_k$, a processor is always assigned to each leaf at depth larger than $D_H(t_k) - N^{1/2}$. Hence, by Lemma 2, this portion of $G_H(t_k)$ will be processed until time $t_k + (1 + (\varepsilon/2))N^{1/2} + R^*$, with probability no smaller than $1 - f_2^{\varepsilon/2}(N^{1/2})$. Furthermore, the probability that R^* is larger than $(\varepsilon/2)N^{1/2}$ is bounded by $f_1^{\varepsilon/2}(N)$. The result follows with $f_6^\varepsilon(N) = f_1^{\varepsilon/2}(N) + f_2^{\varepsilon/2}(N^{1/2})$, which belongs to $\mathscr{U}$. □

The following lemma is a straightforward consequence of Lemma 1(i) and the proof is omitted.

LEMMA 7. *For any $\varepsilon > 0$ there exists some $f_7^\varepsilon \in \mathscr{U}$ such that*

(i) $P(D_\pi(t_k) - D_\pi(t_{k+1}) > (1+2\varepsilon)N^{1/2}$ *and* $\chi_k = 1) \leqq f_7^\varepsilon(N) \quad \forall k$;

(ii) $P(D_H(t_k) - D_H(t_{k+1}) > 2N^{1/2}$ *and* $\chi_k = 1) \leqq f_7^\varepsilon(N) \quad \forall k$.

LEMMA 8. *For any $\varepsilon > 0$ there exists some $f_8^\varepsilon \in \mathscr{U}$ such that $P(k \leqq K$ and $\Delta_k \geqq N^{1/2} + 2\varepsilon k N^{1/2} + 1) \leqq k f_8^\varepsilon(N)$ for all k.*

Proof. The proof is by induction on k and with $f_8^\varepsilon = f_6^\varepsilon + f_7^\varepsilon$. The result is true for $k = 0$, since $\Delta_0 = 0$. Suppose it is true up to some k. In order to prove the result for $k+1$, we condition on the event $K > k$. If $\chi_k = 0$ and $k < K$, then, by definition, $D_H(t_{k+1} - \delta) - d_H(t_{k+1} - \delta) < N^{1/2}$, for sufficiently small $\delta > 0$. Moreover, $G_H(t_{k+1} - \delta)$ has more than M leaves. Thus, by Lemma 3(ii), $D_H(t_{k+1} - \delta) - D_\pi(t_{k+1} - \delta) < N^{1/2}$, for sufficiently small δ. At time t_{k+1} each processor may complete at most one job (because we have assumed that the service times are positive with probability one). Thus, $D_\pi(t_{k+1} - \delta) - D_\pi(t_{k+1}) \leqq 1$. Consequently, $D_H(t_{k+1}) - D_\pi(t_{k+1}) < N^{1/2} + 1$.

The above demonstrates that we only need to consider the case $\chi_k = 1$. Notice that $\Delta_{k+1} = \Delta_k + (D_\pi(t_k) - D_\pi(t_{k+1})) - (D_H(t_k) - D_H(t_{k+1}))$. We argue by contradiction. If $\Delta_{k+1} > 1 + N^{1/2} + 2\varepsilon(k+1)N^{1/2}$, then one of the following three must be happening:

(i) $\Delta_k > N^{1/2} + 2\varepsilon k N^{1/2} + 1$, which is the case with probability bounded by $k f_8^\varepsilon(N)$, using the induction hypothesis;

(ii) $D_\pi(t_k) - D_\pi(t_{k+1}) > N^{1/2} + 2\varepsilon N^{1/2}$, which is the case with probability bounded by $f_7^\varepsilon(N)$, because of Lemma 7(i);

(iii) $D_H(t_k) - D_H(t_{k+1}) < N^{1/2}$, which is the case with probability bounded by $f_6^\varepsilon(N)$, because of Lemma 6. Therefore, the induction step goes through with our choice of f_8^ε. □

LEMMA 9. *For any $\varepsilon > 0$ there exists some $f_9^\varepsilon \in \mathscr{U}$ such that $P(\Delta_K \geqq N^{1/2} + 10\varepsilon N + 1) \leqq f_9^\varepsilon(N)$.*

Proof. This is a trivial consequence of Lemmas 5 and 8. □

LEMMA 10. *For any $\varepsilon > 0$ there exists some $f_{10}^\varepsilon \in \mathscr{U}$ such that $P(D_H(t^*) - D_\pi(t^*) > 4N^{1/2} + 10\varepsilon N) \leqq f_{10}^\varepsilon(N)$.*

Proof. We distinguish three cases: (i) If $t^* = t_K$, then $D_H(t^*) - D_\pi(t^*) = \Delta_K$ and we use the bound of Lemma 9. (ii) If $t^* > t_K$, the definition of t_K implies that $D_H(t^*) - D_\pi(t^*) \leqq D_H(t^*) \leqq D_H(t_K) < N^{1/2}$ and there is nothing to prove. (iii) If $t^* < t_K$, then we have $t_{K-1} < t^* \leqq t_{K-1} + (1+\varepsilon)N^{1/2} = t_K$ and $\chi_{K-1} = 1$. Hence, $D_H(t^*) - D_\pi(t^*) \leqq D_H(t_{K-1}) - D_\pi(t_K) = \Delta_K + D_H(t_{K-1}) - D_H(t_K)$. For Δ_K we use the bound of Lemma 9. For $D_H(t_{K-1}) - D_H(t_K)$ we have $P(D_H(t_{K-1}) - D_H(t_K) > 2N^{1/2}$ and $\chi_{K-1} = 1) \leqq P(K > 5N^{1/2}) + \sum_{k \leqq 5N^{1/2}} P(D_H(t_{k-1}) - D_H(t_k) > 2N^{1/2}$ and $\chi_{k-1} = 1) \leqq$

$f_5^\varepsilon(N)+5N^{1/2}f_7^\varepsilon(N)$. By combining the above observations, we obtain the desired result. □

We can now complete the proof of the theorem. We have the obvious bound $E[T_\pi(G)] \geqq E[t^*]+E[D_\pi(t^*)]-1$. Moreover, using Lemma 2, $E[T_H(G)-t^*|G_H(t^*), R_1(t^*), \cdots, R_M(t^*)] \leqq R^*+D_H(t^*)+\varepsilon N+f_2^\varepsilon(N)$. We now use the fact $E[R^*] \leqq \varepsilon N+f_1^\varepsilon(N)$ and combine the above inequalities to obtain $E[T_H(G)-T_\pi(G)] \leqq E[D_H(t^*)-D_\pi(t^*)]+\varepsilon N+f_1^\varepsilon(N)+\varepsilon N+f_2^\varepsilon(N)+1$. We now use Lemma 10, together with the fact $D_H(t^*)-D_\pi(t^*) \leqq N$, to obtain the bound $E[D_H(t^*)-D_\pi(t^*)] \leqq 4N^{1/2}+10\varepsilon N+Nf_{10}^\varepsilon(N)$. Using the obvious fact $E[T_\pi(G)] \geqq N/M$, we finally obtain $E[T_H(G)-T_\pi(G)] \leqq E[T_\pi(G)](12M\varepsilon+\beta^\varepsilon(N))$, where $\beta^\varepsilon(N)$ satisfies $\lim_{N\to\infty}\beta^\varepsilon(N)=0$, for all $\varepsilon>0$. Recall now that ε was arbitrary and the result follows with $\beta(N)=\min_{\varepsilon>0}(11M\varepsilon+\beta^\varepsilon(N))$. □

Acknowledgments. We would like to thank the referees for a variety of suggestions on improving the style of the paper.

REFERENCES

[BR] J. BRUNO, *On scheduling tasks with exponential service times and in-tree precedence constraints*, Acta Inform., 22 (1985), pp. 139-148.

[Co] E. G. COFFMAN, JR., *Computer and Job Shop Scheduling Theory*, Prentice-Hall, Englewood Cliffs; NJ, 1976.

[Hu] T. C. HU, *Parallel sequencing and assembly-line problems*, Oper. Res., 9 (1961), pp. 841-848.

[Pa] C. H. PAPADIMITRIOU, *Games against nature*, Proc. 24th FOCS, 1983, pp. 446-450.

[CR] K. M. CHANDY AND P. F. REYNOLDS, *Scheduling partially-ordered tasks with probabilistic execution times*, Proc. of the Fifth Symposium on Operating Systems Principles, 1975, pp. 169-177.

[Ha] B. HAJEK, *Hitting-time and occupation-time bounds implied by drift analysis with applications*, Adv. in Appl. Probab., 14 (1982), pp. 502-525.