Scheduling Rooted Forests with Communication Delays

Garth Isaak*

Abstract

We show that a greedy algorithm for scheduling unit time jobs on two machines with unit communication delays produces an optimal schedule when the precedence constraints are given by a rooted forest. We also give a min/max relationship for the length of such a schedule. The min/max result (for forests and two machines) shows that the addition of unit communication delays increases the optimal schedule length by at most one.

1 Introduction

We consider scheduling to minimize the maximum completion time with two machines, unit time jobs, unit communication delay and precedence constraints given by a rooted forest. The general problem of scheduling with communication delays arises in parallel processing. We will focus on proving a min/max result and simultaneously showing that a simple greedy algorithm produces an optimal schedule. The min/max result shows that the length of an optimal schedule is $\lceil (|F|+Q)/2 \rceil$ where Q maximizes over non-leaf jobs x, the number of jobs preceeding x plus 2 minus the number of jobs incomparable to x. (Q is 0 if this quantity is aways negative.) Using this, we show that if td is the length of an optimal schedule when communcication delays are present and tn is the length of an optimal schedule with no communication delays, then $td = tn$ or $tn + 1$.

*Department of Mathematics, Lehigh University, Bethlehem, PA 18015
The author thanks Ivan Rival for bringing this problem to his attention.
Partially supported by a grant from the Reidler Foundation

More formally, we view the problem as follows. An order $(F, \preceq)$, with diagram a disjoint union of rooted trees (a rooted forest), is given. Sometimes, precedence constraints are given as digraphs, in which case we would be dealing with out forests. The elements are to be labeled with an ordered pair $(m, t) \in \{1, 2\} \times \mathbf{N}$ such that

$$\text{if } x \neq y \text{ and } m(x) = m(y) \text{ then } t(x) \neq t(y) \tag{1}$$

$$\text{if } x \preceq y \text{ then } t(x) < t(y) \tag{2}$$

$$\text{if } x \preceq y \text{ and } m(x) \neq m(y) \text{ then } t(x) + 1 < t(y) \tag{3}$$

Such a labeling will be called a *schedule*. Its *length* is the maximum value of t assigned to an element.

In terms of scheduling, the elements of the order represent jobs, the first coordinate represents the machine on which the job is processed and the second the time of processing. Equation (1) ensures that two jobs cannot be processed at the same time on the same machine. Equation (2) gives the precedence constraints, if x precedes y then x is scheduled before y. Equation (3) adds communication delays to the precedence constraints. If x precedes y and x and y are scheduled on different machines, then x is scheduled at least two time units before y, to allow one time unit for communication. This arises if we consider the jobs being processed on a simple model of a parallel computer, if job x is executed on a different machine from y, then one time unit is needed for the machines to communicate.

Scheduling subject to the constraints (1) and (2) is well studied. See for example [3], [4], or [7], for surveys. When scheduling with communication delays, we add constraint (3). See [1] or [10] for surveys of this problem and its variants. Scheduling with delays, m machnes and an arbitrary precedence order is NP-complete (Rayward-Smith [8]). The problem remains NP-complete even if the precedence constraints are given by a rooted tree and the number of machines m is part of the input (Lenstra, Veldhorst, Veltman [5]). If the precedence constraints are a rooted forest $|F|$ and the number of machines m is fixed, Varvarigou, Roychowdhury and Kailath [9] give an $O(|F|^{2m})$ algorithm for scheduling with delays. With two machines and a general order the complexity remains open, in contrast to the case with no delays, where several efficient algorithms are known.

After completing the first version of this paper, we discovered that a number of other efficient algorithms for scheduling rooted forests with two machines and commmunication delays have been given. In addition to the algorithm of Varvarigou, Roychowdhury and Kailath [9] mentioned above, there is a quadratic algorithm by Picouleau [6] and linear (in $|F|$) algorithms by Lenstra, Veldhorst and Veldman [5] and Lawler [2]. It appears that our algorithm is different from these others and it has an advantage of simulataneously allowing proof of a min/max result. We will focus more on the min/max result and the relationship to scheduling without delays than on the details of our algorithm.

2 Min/Max result and the algorithm

We will use the notion of idle machine/time units. For a given schedule S with length t, the idle time $idle(S)$ is the number of pairs (i, j) with $j \leq t$ *not* assigned to some element of the tree. This sums the idle time over both machines. Note that if there are n jobs, then the length of an optimal schedule S is $(n+idle(S))/2$.

For an element x in F let

$$P_F(x) = \{y \in F | y \preceq x\} \text{ and } I_F = \{y \in F | y \sim x\} \text{ and } S(x) = \{y \in F | x \preceq y\}.$$

Here $\sim$ denotes incomparability ($x \sim y$ iff neither $x \preceq y$ nor $y \preceq x$). So $I(x)$ is the set of elements with no precedence relation to x, $P(x)$ the elements preceding x (plus x) and $S(x)$ the elements succeeding x (plus x). The subscript F will be used only when necessary to distinguish between several forests. We will call the unique immediate predecessor of x in the forest the parent of x.

Let

$$Q_F(x) = |P_F(x)| + 1 - |I_F(x)|.$$

A non-leaf element x is one for which some elements follow it, i.e., $|S(x)| \geq 2$. Let Q_F be the maximum value of $Q_F(x)$ over non-leaf elements x if this is positive and let $Q_F = 0$ otherwise.

If the path to x is scheduled consecutively on one machine, and elements incomparable to x are scheduled on the other, and there are more elements on the path than incomparable to x, then $|P(x)| - |I(x)|$ units of idle time are forced. $Q(x)$

represents a measure of this idle time. If this quantity is positive and x is non-leaf then an additional idle unit appears during the succeeding time period since at most one successor of x can be scheduled. Hence the $+1$ term in the definition of $Q(x)$. If $Q(x)$ is maximum, any schedule which does not first schedule the path to x will have even more idle time, so $Q(x)$ represents the maximum amount of idle time. This is the idea behind the lower bound of the min/max theorem. We will use the algorithm to show that there is a schedule achieving this bound.

In the example of figure 1, observe that in the tree (without the isolated elements), Q is three, from the element scheduled at time 6 by the 'box' machine. Thus any schedule will have length at least $\lceil (12+3)/2 \rceil = 8$ and this is attained by the schedule shown in the figure. The extra isolated elements in the figure can be considered dummy elements to fill the idle time as will be done in the proofs that follow.

Theorem 1 *Let $(F, \preceq)$ be an order with diagram a rooted forest. The minimum length of a schedule on F with two machines and unit jobs and unit communication delays is equal to* $\left\lceil \dfrac{|F| + Q_F}{2} \right\rceil$.

Greedy Algorithm for Scheduling

Input: A rooted forest F.

Starting with time $t = 1$ repeat the following to select elements to be scheduled at time t. Select x and y that maximize $|S(x)| + |S(y)|$ among all pairs x, y that can be simultaneously scheduled at time t subject to the precedence and communication constraints.

Observe that this greedy procedure will pick x and y with two largest $|S|$'s unless both have the same parent and that parent was scheduled at time $t-1$, i.e., both must be scheduled on the same machine if scheduled at time t. It is not difficult to see that this algorithm can be implemented in $O(|F|)$ time if F is represented by the forest of covering relations.

Theorem 2 *Scheduling a rooted forest by the greedy algorithm produces a schedule with minimum length.*

To prove both Theorem 1 and Theorem 2, we need to prove that the bound $\left\lceil \frac{|F| + Q_F}{2} \right\rceil$ of Theorem 1 is indeed a lower bound on the schedule length and that the greedy algorithm produces a schedule attaining this bound.

Proof of the lower bound for Theorem 1:

Let S be any schedule for F. If $Q_F = 0$ the bound is immediate. Let x be a non-leaf element with $Q(x) > 0$. Any element succeeding x must be scheduled after x. Since a path of length $|P(x)|-1$ precedes x, the earliest x can be scheduled is at time $|P(x)|$. So $t(x) \geq |P(x)|$. At most $|P(x)|+|I(x)|$ elements are scheduled by time $t(x)$. There are $2t(x)$ machine/time units during this period. So $idle(S) \geq 2t(x) - |P(x)| - |I(x)| \geq 2|P(x)| - |P(x)| - |I(x)| = |P(x)| - |I(x)| = Q(x) - 1$. Additionally, if all of $I(x)$ is scheduled before time $t(x) + 1$, then one machine is idle at time $t(x) + 1$ due to communication delays (since only one successor of x can be scheduled at time $t(x)+1$ in this case). If some element in $I(x)$ is scheduled at time $t(x) + 1$, then there is an extra unit of idle time before $t(x) + 1$ as at most $|I(x)| - 1$ elements are scheduled by $t(x)$ in this case. In either case, there is another unit of idle time and $idle(S) \geq Q(x)-1+1 = Q(x)$. So the bound follows. □

For an instance of the algorithm running on a forest F, let F_i be the rooted forest of elements scheduled at time i or later. So $F = F_1$. Also, for $x \in F_j$, let $I_j(x)$ denote $I_F(x)$ restricted to F_j.

Lemma 1 *Let $S(x)$, with root x be a component of F_j, $j \geq 2$. If $|S(x)| \geq 2 + |I_j(x)|$, then the parent p of x was scheduled at time $j - 1$ by the greedy algorithm. Furthermore, $S(p)$ is a component of F_{j-1} with $|S(p)| \geq 2 + |I_{j-1}(p)|$.*

Proof: Assume that p is scheduled earlier, in F_{j-1} there is a component C at least as large as $S(x)$, whose root gets scheduled at time $j - 1$ instead of x. Only the root of this component can be scheduled at time $j - 1$ so $|C| - 1$ elements remain

in F_j. Thus, $|S(x)| - 1 \leq |C| - 1 \leq I_j(x)$, a contradiction. So p is scheduled at time $j-1$ and $S(p)$ is in F_{j-1}. At most one other element is also scheduled at time $j-1$, so $|I_{j-1}(p)| \leq |I_j(x)| + 1$. Here we have used $y \sim p \Rightarrow y \sim x$ for p the parent of x. Since also $|S(p)| \geq |S(x)| + 1$, $|S(p)| \geq 2 + |I_{j-1}(p)|$ follows. $\square$

Proof that the greedy algorithm produces a schedule with length $\left\lceil \dfrac{|F| + Q_F}{2} \right\rceil$*:*

It suffices to prove that the algorithm works when $Q_F = 0$. If $Q_F \geq 1$, add Q_F isolated elements (roots) to form a new forest F'. The new isolated elements are leaves, hence do not affect $Q_{F'}$. (Note it is also this assumption that an isolated element is a root as well as a leaf that allows $Q_F = 0$ in the case that F consists of one or two isolated elements.) If $x \in F$ is not a leaf, then $Q_{F'}(x) = Q_F(x) - Q_F \leq 0$. So $Q_{F'} = 0$. Then F' is scheduled with minimum length

$$\left\lceil \frac{|F'| + Q_{F'}}{2} \right\rceil = \left\lceil \frac{|F'|}{2} \right\rceil = \left\lceil \frac{|F| + Q_F}{2} \right\rceil.$$

If an element of F and a new element are both eligible to be scheduled when applying the greedy algorithm to F', both are isolated in the forest of unscheduled elements. Select the element from F in this case. Then the schedule for F' restricted to F is the same as that produced by the greedy algorithm applied to F.

Assume $Q_F = 0$. We need to show that at each time, except possibly the last, $\lceil |F|/2 \rceil$, there are at least two distinct elements that can be scheduled. There are at least two roots in the forest, otherwise the single root r has $Q(r) = 2$, contradicting $Q_F = 0$. So two elements are scheduled at time $t = 1$.

Let t be the first time (if any) during which at most one element is scheduled. Assume for contradiction that $t < \lceil |F|/2 \rceil$. One of the elements scheduled at time $t-1$ is a leaf, since if both have sucessors, two sucessors will be scheduled at time t. Let p be the non-leaf element. Then F_{t-1} is the disjoint union of $S(p)$ and an isolated element. There can be no other elements in F_{t-1} since they would be eligible to be scheduled along with a successor of p at time t. Note that there are at least two elements scheduled at time t or later since $t < \lceil |F|/2 \rceil$. So $|S(p)| \geq 3$.

Since $|S(p)| \geq 3$ and $|I_{t-1}(p)| = 1$, we may apply Lemma 1. By repeated application of this lemma, we see that there is a path $v_1, v_2, \ldots, v_{t-1} = p$ in F with v_i scheduled at time i. Furthermore, there are at most $t-1$ other elements in F that are not in F_t (i.e., the elements scheduled by time $t-1$). Then, $Q_F(p) = |P_F(p)| + 1 - |I_F(p)| \geq (t-1) + 1 - (t-1) = 1$, a contradiciton to the assumption that $Q_F = 0$. $\square$

There are number of other structural result about Q and F that are not difficult to prove. For example, if x is a non-leaf element such that $Q_F(x) = Q_F$ then for every other element y (including leaves) with $Q_F(y) = Q_F$, either x is on the path from a root to y or vice-versa. From this, one can see that the algorithm schedules during times $1, 2, \ldots |P(x)|$ the path to x on one machine and the remaining elements not below x on the second machine. For any element x, it is easy to see that if $S = S(x)$ and y is a successor of x then $Q_S(y) + Q_F(x) - 2 = Q_F(y)$. With such straightforward observations variations on the greedy algorithm (for example a recursive algorithm) can easily be given.

3 Comparison to Non-delay Schedules

In this section we use Theorem 1 to show that the addition of communication delays increases schedule length by at most one (for the case of rooted forests and two machines).

There is a known min/max result for schedule length, m machines and rooted trees when no communication delays are present. See for example Poguntke [7]. We will state the case when $m = 2$ using the notation of this paper. Let $\tilde{Q}(x) = |P(x)| - |I(x)|$ and let $\tilde{Q}_F$ be the maximum value of $\tilde{Q}(x)$ over non-leaf elements x if this is positive and let $\tilde{Q}_F = 0$ otherwise. So, $\tilde{Q}_F = Q_F - 1$ whenever $Q_F > 0$ and $\tilde{Q}_F = 0$ when $Q_F = 0$. Then the minimum length of a schedule subject to conditions (1) and (2) is $\lceil (|F| + \tilde{Q})/2 \rceil$. It is straightforward to check that this is just a restatement of the known min-max result as presented in Poguntke [7], viewed from the perspective of idle times rather than levels. Rather than introducing the notation of Poguntke to show this, we sketch a proof which is nearly identical to

that given in Section 2.

From the proof of the lower bound for Theorem 1, delete the extra idle time at $t+1$ due to communication delays. Then $idle(S) \geq Q(x) - 1 = \tilde{Q}(x)$. If x is scheduled at time j and $|S(x)| \geq 1 + |I_j(x)|$ then the parent p of x is scheduled at time $j-1$ and $|S(p)| \geq |I_{j-1}(p)| + 1$. This is just a non-delay version of Lemma 1. The proof is analogous, except there must be at least two components C_1 and C_2 with size at least $|S(x)|$ in F_{j-1}. If $|S(x)| = 1$ the result is trivial to check. Otherwise, $|S(x)| \leq (|C_1| - 1) + (|C_2| - 1) \leq |I_j(x)|$. The rest is the same as the proof of Lemma 1. Finally, if t is the first time when one element p is scheduled, then F_t consists of the single component $S(p)$. Applying the non-delay version of Lemma 1 stated above, we get $|P(p)| = t$ and at most $t-1$ other elements. So $|I(p)| \leq t-1$ and we get the contradiction $Q(p) \geq |P(p)| - |I(p)| \geq t - (t-1) = 1$.

Let $td(F)$ denote the minimum schedule length (with two machines) subject to (1), (2) and (3), i.e., with communication delays and $tn(F)$ the minimum length of a schedule subject to (1) and (2), i.e., with no delays.

Corollary 1 *Let F be a rooted forest. Then $td(F) = tn(F)$ or $tn(F) + 1$. The second case occurs only if for x such that $Q_F(x) = Q_F > 0$, $|S(x)|$ is odd.*

Proof: Any schedule for the problem with delays is also feasible for the problem with no delays. Hence $tn(F) \leq td(F)$.

If $Q_F = 0$ then $\tilde{Q}_F = 0$ and both schedules have length $\lceil |F|/2 \rceil$. Assume $Q_F > 0$. Then $Q_F - 1 = \tilde{Q}_F$. By Theorem 1 and the observations above

$$td(F) = \left\lceil \frac{|F| + Q_F}{2} \right\rceil = \left\lceil \frac{|F| + \tilde{Q}_F + 1}{2} \right\rceil \leq tn(F) + 1.$$

Additionally, $tn(F) + 1 = td(F)$ exactly when $|F| + Q_F$ is odd.

$$|F| + Q_F = (|P(x)| + |S(x)| + |I(x)| - 1) + (|P(x)| + 1 - |I(x)|) = 2|P(x)| + |S(x)|.$$

This is odd exactly when $|S(x)|$ is odd. □

We conclude by asking about a bound for m machine scheduling with delays in terms of non-delay schedule lengths. Let $td_m(F)$ and $tn_m(F)$ denote the m machine analogs of $tn(F)$ and $td(F)$. It seems reasonable to conjecture that $td_m(F) \leq$

$tn_m(F) + \lfloor \log_2 m \rfloor$. Corollary 1 is a special case. One reason for this is to look at complete binary trees as a potential 'worst case'. Let B_h denote height h order with the complete binary tree as diagram (i.e., 2^h leaves and$2^{h+1}-1$ elements). Let $h = \lfloor \log_2 m \rfloor$. Then it is easy to check that $td_m(B_h) = 2h+1$ and $tn_m(B_h) = h+1$. Adding a chain of arbitrary length above the root or equal length chains below each leaf gives the bound for arbitrary size forests.

4 References

1. Chretienne, P., and C. Picouleau, Scheduling with communication delays: A survey, in Proc. Summer School on Scheduling Theory and its Applications, 1992, to appear.

2. Lawler, E.L., Sceduling trees on multiprocessors with unit communication delays, manuscript, 1993.

3. Lawler, E.L. and J.K. Lenstra, Machine scheduling with precedence constraints, in I. Rival (ed.) Ordered sets, Reidel, 1982, 655–675.

4. Lawler, E.L. and J.K. Lenstra, A.H.J. Rinooy Kan, and D. Shmoys, Sequencing and scheduling: algorithms and complexity, in S.C. Graves et al. ed., Handook in operations research and management science, vol. 4; Logistics of production and inventory, Elsevier, 1993, 445–522.

5. Lenstra, J.K., M. Veldhorst and B. Veltman, The complexity of scheduling trees with communication delays, manuscript, preliminary version in T. Lengauer ed., Algorithms - ESA '93, Lecture notes in Computer Science 726, Springer-Verlag, 1993, 284–294.

6. Picouleau, C., Etude de problemes d'optimisation dans les systemes distribues, Ph.D. thesis, Univ. Pierre et Marie Curie, Paris, 1992.

7. Pogunkte, W., Order-theoretic aspects of scheduling, in Combinatorics and Ordered Sets, in I. Rival ed., AMS Contemp. Math. vol 57, 1986, 1–31.

8. Rayward-Smith, V.J., UET scheduling with unit interprocessor communication delays, Disc. App. Math. **18** (1987) 55–71.

9. Varvarigou, T.A., V.P. Roychowdhury and T. Kailath, Scheduling in and out forests in the presence of communication delays, manuscript, 1992.

10. Veltman, B., B.J. Lageway and J.K. Lenstra, Multiprocessor scheduling with communication delays, Parallel Computing **16** (1990) 173–182.