T. Lang
E. B. Fernández

Improving the Computation of Lower Bounds for Optimal Schedules

Abstract: Ways of decreasing the number of operations needed to compute the lower bounds of optimal schedules, by reducing the number of time intervals that must be considered, are presented. The bounds apply to a system of identical processors executing a partially ordered set of tasks, with known execution times, using a non-preemptive scheduling strategy. In one approach we find that the required number of intervals depends on the graph. In our other approach, which subsumes the first, the number of intervals is decreased to at most $\min[D^2/2, n^2]$, where D is the deadline to complete the tasks and n is the number of tasks. The actual number of intervals for a particular graph can be considerably smaller than this worst case.

Introduction

In a previous paper [1] we discussed efficient ways of computing lower bounds for the optimal schedules presented by Fernández and Bussell [2]. The number of operations required is approximately $D^2/2$, where D is the deadline to complete the tasks, because one operation is performed for each interval $[t_1, t_2]$ for $0 \leq t_1$ and $t_2 \leq D$. Any reduction in this number of operations must result from a reduction in the number of intervals to be considered. It was shown in [1] that the number of intervals to consider can be reduced if the graph structure is of a specific type, such as a tree, a set of independent tasks, or a set of independent chains. In those cases it is possible to take into account only some specific intervals. Here we show that similar reductions in the number of intervals can be obtained by examining the particular time constraints of the tasks of the graph.

The importance of lower bounds for optimal scheduling has been discussed elsewhere [1–4]. Lower bounds are useful also for evaluating approximate scheduling methods [5]. Therefore, an effort to obtain accurate lower-bound expressions and to find efficient ways of calculating these expressions is well justified.

This paper should be considered a continuation of previous work [1], and familiarity with that work is an important requirement in understanding the ideas presented here. We use the model, concepts, and notation hitherto introduced in [1] (which we summarize in the second section), and only new concepts are defined in detail.

In the third section, a way of eliminating some intervals from consideration is presented. The improvement obtained depends on the structure of the graph, and a modification of the algorithm given previously [1] is presented to incorporate this saving. In the final section, reductions in the number of intervals are found by studying the regions along the time axis where $M(t_1, t_2)/(t_2 - t_1)$, the average number of tasks which have to be executed in interval $[t_1, t_2]$, changes monotonically. This reduction allows calculation of the lower bound on the number of processors, m_L, using at most only $\min[D^2/2, n^2]$ intervals, or $\min[D^2/2, 2n^2]$ intervals if an incremental method of computation is used. We show that for specific graphs the actual number of intervals required is considerably smaller.

Definitions and previous results

A set of *tasks* $T = \{T_1, T_2, \cdots, T_n\}$ is to be executed by a set of identical *processors* P_i, $i = 1, 2, \cdots, m$. A partial order $<$ is given on T, and a non-negative integer d_j represents the *duration of execution* of task T_j. The tasks are assigned to processors using a non-preemptive type of scheduling, and they must be completed within a *deadline* D. According to some schedule, for each T_j we have a completion time, C_j. The precedences of the partial ordering determine for a given T_j a minimum time in which this task can be finished, its earliest completion time, e_{cj}. The *latest completion time* of T_j, l_{cj}, indicates how long the completion time of this task can be delayed without exceeding the deadline. Similarly, a given schedule defines for T_j an initiation time, and we have an *earliest initiation time*, e_{ij}, and a latest initiation time, l_{ij}. If all the tasks are in their earliest possible positions, the number of units of task T_j that lie in the interval $[t_1, t_2]$ is denoted as $e_j(t_1, t_2)$. Similarly, if all tasks lie in

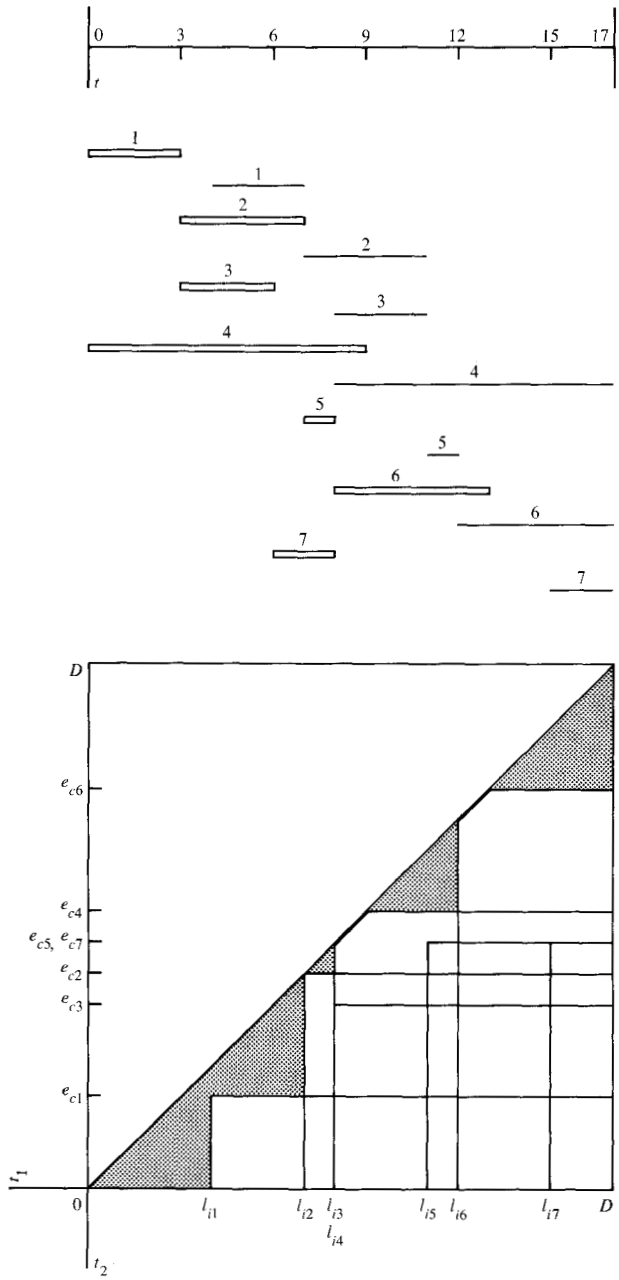

Figure 1 Example for Theorem 1.

their latest possible positions, the number of units of task T_j in this interval is called $l_j(t_1, t_2)$.

A lower bound on the minimum number of processors required to perform the computations of G in time D is given by

$$m_L = \left\lceil \max_{[t_1, t_2]} \left\{ \frac{1}{t_2 - t_1} \sum_{j=1}^{n} \min[e_j(t_1, t_2), l_j(t_1, t_2)] \right\} \right\rceil.$$

Because of its importance for computational purposes, we make the following definition:

$$M(t_1, t_2) = \sum_{j=1}^{n} \min[e_j(t_1, t_2), l_j(t_1, t_2)],$$

which allows rewriting m_L as

$$m_L = \left\lceil \max_{[t_1, t_2]} \frac{M(t_1, t_2)}{t_2 - t_1} \right\rceil.$$

Method to reduce the number of intervals

• *Intervals where $M(t_1, t_2)$ is zero*

The following theorem allows those intervals where $M(t_1, t_2)$ is zero to be eliminated from consideration.

Theorem 1

Assume that the n tasks are numbered such that their latest initiation times are in ascending order, i.e., we have

$$l_{ij} \leq l_{ik} \qquad \text{for } j < k, 0 \leq j, k \leq n.$$

Then

$$M(t_1, t_2) = 0$$

$$\text{for } (t_2 \leq l_{ij}) \wedge (t_1 \geq \max[\{e_{cp} | 1 \leq p \leq j-1\}]).$$

Proof

From our earlier work [1], by definition we have

$$M(t_1, t_2) = \sum_{q=1}^{n} \min(e_q(t_1, t_2), l_q(t_1, t_2)).$$

If $t_2 \leq l_{ij}$ then $l_q(t_1, t_2) = 0$ for $q \geq j$, because by hypothesis the latest initiation times are in ascending order. Also, if $t_1 \geq \max[\{e_{cp} | 1 \leq p \leq j-1\}]$, then $e_p(t_1, t_2) = 0$ for $p < j$. Consequently, $M(t_1, t_2) = 0$ for these values of t_1 and t_2.

Figure 1 illustrates the effects of this theorem. (In this figure, and in subsequent figures, the tasks' earliest positions are indicated by double lines.) The coordinate axes represent t_1 and t_2, respectively. The triangle $(0, D)$, (D, D), $(D, 0)$ is the limit of the intervals $[t_1, t_2]$ that can exist since, for any legal interval, $t_1 \leq t_2$. Therefore, all the legal values for pairs (t_1, t_2) lie within this triangle. The rectangles (perhaps with corners missing) correspond to the limits for the intervals $[t_1, t_2]$ where a given task makes a contribution to $M(t_1, t_2)$ [i.e., every

interval within the rectangle of task T_j has some contribution from T_j in its $M(t_1, t_2)$]. The shaded regions represent the intervals for which $M(t_1, t_2) = 0$ by Theorem 1. These intervals need not be considered in the computation of the lower bound.

• *Improved algorithm*

The algorithm previously given [1] to calculate $M(t_1, t_2)$ incrementally can now be modified to incorporate the results of Theorem 1. The improved algorithm uses the same arrays as the previous algorithm (plus two new arrays) and uses the procedure COMPUTE NEW F from that algorithm (this procedure is used here without showing its details). This algorithm requires the tasks to be numbered such that their latest initiation times are in ascending order. Also, this algorithm includes the calculation of $\max_{[t_1, t_2]}[M(t_1, t_2)/(t_2 - t_1)]$, which is given separately in [1, Section 3.4].

Arrays

OLI[1:n] contains latest initiation times in ascending order;

MM[1:d] contains max M for interval of length l.

Scalars

MEC partial maximum earliest completion time, i.e., $\max(e_{cj} | j \leq k)$.

```
BEGIN
  k = 1; MEC = OLI(1);
  MM = 0;
  FOR t1 = 0 UNTIL D − 1 DO
    BEGIN
      FOR j = 1 UNTIL n DO
        BEGIN 'COMPUTE NEW F';
          IF (t1 > EI(j))∧(t1 ≤ LI(j))∧(t1 ≤ EC(j))
          THEN
            BEGIN
              F(T(j)) = F(T(j)) − 1;
              T(j) = T(j) − 1;
            END;
        END;
      M(t1, OLI(k)) = 0;/ * by Theorem 1 * /
      FOR t2 = max(t1 + 1, OLI(k) + 1) UNTIL D DO
        BEGIN
          M(t1, t2) = M(t1, t2 − 1) + F(t2);
          MM(t2 − t1) = max(MM(t2 − t1), M(t1, t2));
        END;
      IF t1 = MEC − 1 THEN
        BEGIN
          WHILE (MEC ≥ EC(k)) ∧ (k ≠ n) DO
            k = k + 1;/ * keeps increasing k
            if previous ec is larger than ec
            of task under consideration * /
            MEC = EC[k];
```

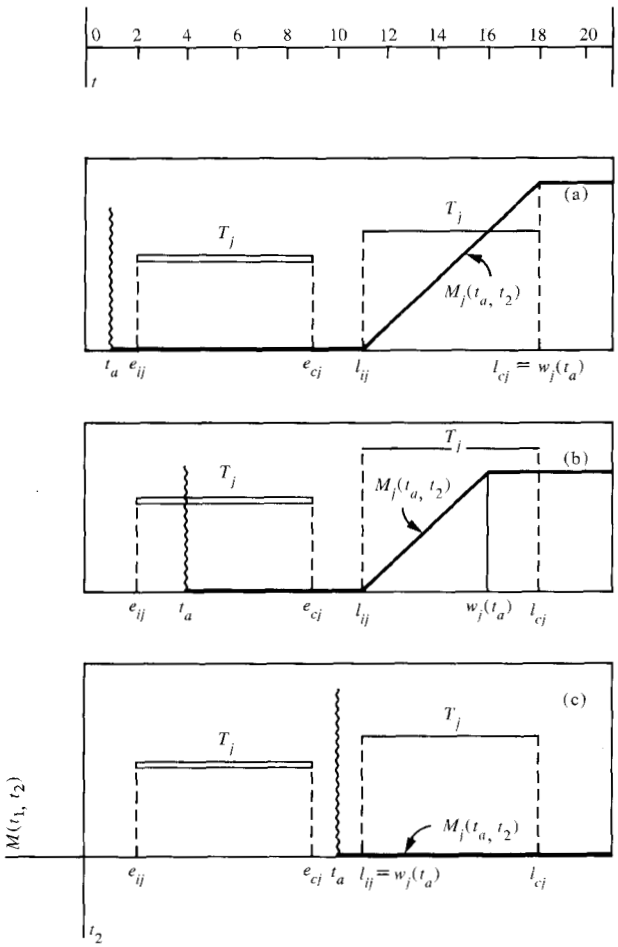

Figure 2 Graphs of $w_j(t_a)$ for different positions of t_a.

```
        END;
    END;
END.
```

Further reduction of the number of intervals

• *Shape of* $M(t_1, t_2)$

We denote as $M_j(t_1, t_2)$ the contribution to $M(t_1, t_2)$ of a specific task T_j. When the interval $[t_1, t_2]$ starts at a time t_a, the portions of task T_j that have to be done in the interval $[t_a, t_2]$ increase linearly from $t_2 = l_{ij}$ to $t_2 = w_j(t_a)$. This latter value, defining a knee-point in $M_j(t_a, t_2)$, depends on the relative position of t_a with respect to e_{ij} and e_{cj} as follows:

$$w_j(t_a) = \begin{cases} l_{cj} & \text{if } t_a \leq e_{ij}, \quad \text{Fig. 2(a)} \\ l_{cj} - (t_a - e_{ij}) & \text{if } e_{ij} < t_a < e_{cj}, \quad \text{Fig. 2(b)} \\ l_{ij} & \text{if } t_a \geq e_{cj}. \quad \text{Fig. 2(c)} \end{cases}$$

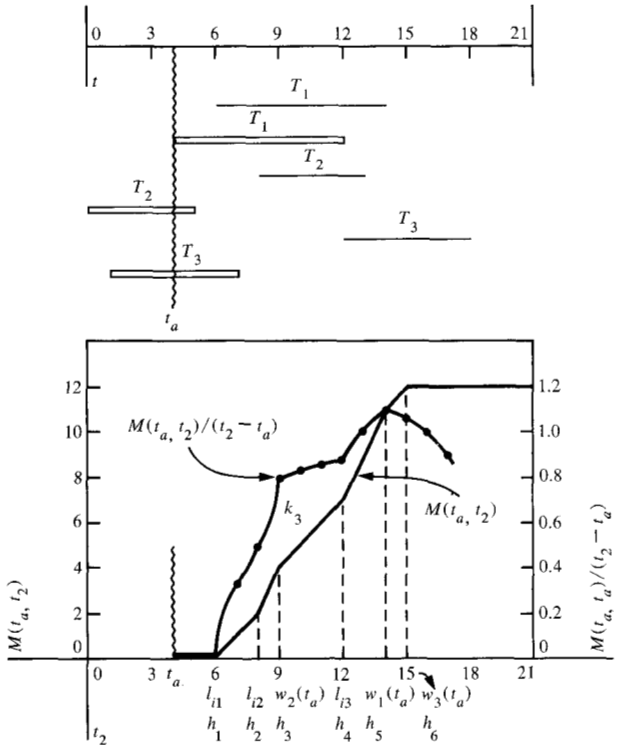

Figure 3 Superposition of tasks contributing to $M(t_a, t_2)$.

The total value of $M(t_a, t_2)$ can be obtained by superimposing the contributions of all the tasks in the graph, which is monotonically increasing with respect to t_2. Here $M(t_a, t_2)$ consists of linear segments of integer slope $0 \leq k \leq n$. The slope changes at points l_{ij} and $w_j(t_a)$. We denote these points as h_r, where r is an increasing index, $1 \leq r \leq n$. An example is given in Fig. 3, which also shows the corresponding $M(t_a, t_2)/(t_2 - t_a)$.

We now present two theorems that permit a further reduction on the number of intervals that have to be considered to compute the lower bound m_L, by defining regions where $M(t_1, t_2)/(t_2 - t_1)$ changes monotonically.

Theorem 2

In a region in which $M(t_a, t_2)$ has constant slope, $M(t_a, t_2)/(t_2 - t_a)$ is monotonic with respect to t_2.

Proof

For the region satisfying the hypothesis, we can write

$$M(t_a, t_2) = M_0 + k(t_2 - t_0),$$

and

$$\frac{M(t_a, t_2)}{t_2 - t_a} = \frac{M_0 + k(t_2 - t_0)}{t_2 - t_a},$$

where t_0 is the beginning time of the region, and k is the constant slope in that region. This function is monotonically increasing, constant, or monotonically decreasing depending on whether M_0 is less than, equal to, or greater than $k(t_0 - t_a)$, respectively.

Corollary

As $M(t_a, t_2)/(t_2 - t_a)$ is monotonic in each region of constant slope, to obtain $m_L = \max_{t_a}[\max_{t_2}[M(t_a, t_2)/(t_2 - t_a)]]$, it is sufficient to consider only those values of t_2 for which the slope changes, i.e., those that correspond to l_{ij} or $w_j(t_a)$.

A further elimination of t_2 values can be accomplished by proving that a maximum of the quotient $M(t_a, t_2)/(t_2 - t_a)$ cannot occur at some of the values of t_2 selected by the corollary of Theorem 2.

Theorem 3

Let k_{r-1} be the slope of $M(t_a, t_2)$ before time h_r, and let k_r be the slope after time h_r. Then, $\max_{t_2}[M(t_a, t_2)/(t_2 - t_a)]$ does not occur at $t_2 = h_r$ if $k_r > k_{r-1}$. That is, the maximum of $M(t_a, t_2)/(t_2 - t_a)$ cannot be at a point where the slope of $M(t_a, t_2)$ increases.

Proof

Let

$$Q_r = M(t_a, h_r)/(h_r - t_a).$$

We prove that 1) when $Q_r \leq k_r$, the value of this quotient at time $h_r + 1$, denoted as Q', is greater than or equal to Q_r; 2) when $Q_r > k_r$, the value of the quotient at time $h_r - 1$, denoted as Q'', is greater than or equal to Q_r.

Proof for $Q_r \leq k_r$

At time $h_r + 1$ we have

$$Q' = [M(t_a, h_r) + k_r]/(h_r - t_a + 1),$$

but $M(t_a, h_r) = (h_r - t_a)Q_r$; therefore,

$$\begin{aligned} Q' &= [(h_r - t_a)Q_r + k_r]/(h_r - t_a + 1) \\ &= [(h_r - t_a + 1)Q_r + k_r - Q_r]/(h_r - t_a + 1) \\ &= Q_r + (k_r - Q_r)/(h_r - t_a + 1). \end{aligned}$$

The second term in the right hand side expression is ≥ 0 when $Q_r \leq k_r$; therefore, $Q' \geq Q_r$.

Proof for $Q_r > k_r$

At time $h_r - 1$ we have

$$Q'' = [M(t_a, h_r) - k_r]/(h_r - t_a - 1),$$

and replacing as in the previous case we get

$$Q'' = Q_r + (Q_r - k_r)/(h_r - t_a - 1).$$

Again the second term is ≥ 0 and $Q'' \geq Q_r$.

Therefore, the maximum of Q along t_2 occurs either before h_r or after h_r, but not at h_r, which proves the theorem.

Combining the Corollary of Theorem 2 with Theorem 3 we see that to calculate m_L it is sufficient to consider,

for a given $t_1 = t_a$, only the intervals that end at values of t_2 at which the slope of $M(t_1, t_2)$ decreases. This happens at all times when the number of tasks that have $w_j(t_a) = t_2$ exceeds the number of tasks that have $l_{ij} = t_2$. In this way the number of intervals required in the worst case is reduced from $\frac{1}{2}D^2$ to nD. The number of intervals for a particular case is smaller because 1) not all the $w_j(t_a)$ are different; 2) when t_1 is increased more and more tasks are eliminated from consideration (tasks with e_c less than the current t_1 do not participate in the calculation); and 3) for a given $w_j(t_a)$ more tasks can begin than end, i.e., the slope of $M(t_a, t_2)$ increases.

• *Elimination of t_1 values*

Theorems 2 and 3 permit a reduction in the number of values of t_2 that have to be considered for a given t_1. We now obtain a further reduction by showing that it is not necessary to consider all values of t_1 either.

Theorem 4

$M(t_a, w_k(t_a))/(w_k(t_a) - t_a)$ is monotonic as a function of t_a, in a region in which t_a is between two successive earliest times (initiation or completion), corresponding to tasks T_j and T_k.

Proof

We compare $M(t_a + 1, w_k(t_a + 1))$ with $M(t_a, w_k(t_a))$, i.e., we consider how the value of M changes when the initiation of the interval changes. When we move the initiation of the interval from t_a to $t_a + 1$ and the termination of the interval from $w_k(t_a)$ to $w_k(t_a + 1)$, the contribution of task T_j to $M(t_a, w_k(t_a))$ is reduced by 1 if one of the following conditions is satisfied:

1. $(t_a < e_{ij}) \wedge (l_{cj} \geq w_k(t_a)) \wedge (e_{ik} \leq t_a < e_{ck})$;
2. $(e_{ij} \leq t_a < e_{cj}) \wedge (w_j(t_a) \leq w_k(t_a))$;
3. $(e_{ij} \leq t_a < e_{cj}) \wedge (w_j(t_a) > w_k(t_a))$
   $\wedge (e_{ik} \leq t_a < e_{ck})$;

This contribution is not changed otherwise.

To visualize why this is true we consider all the possible positions for t_a and its corresponding $w_k(t_a)$, and analyze the effect of increasing t_a by 1. First, we notice from Fig. 2 that

$$w_k(t_a + 1) = w_k(t_a) \quad \text{if } t_a < e_{ik}$$

and

$$w_k(t_a + 1) = w_k(t_a) - 1 \quad \text{if } e_{ik} \leq t_a < e_{ck}.$$

Then, the effect on $M_j(t_a, w_k(t_a))$ is as follows:

$t_a < e_{ij}$ {

- $w_k(t_a) > l_{cj} \rightarrow$ no change; — Fig. 4(a)
- $w_k(t_a) \leq l_{cj}$ and $t_a < e_{ik} \rightarrow$ no change; — Fig. 4(b)
- $w_k(t_a) \leq l_{cj}$ and $e_{ik} \leq t_a < e_{ck} \rightarrow$ change; — Fig. 4(c)

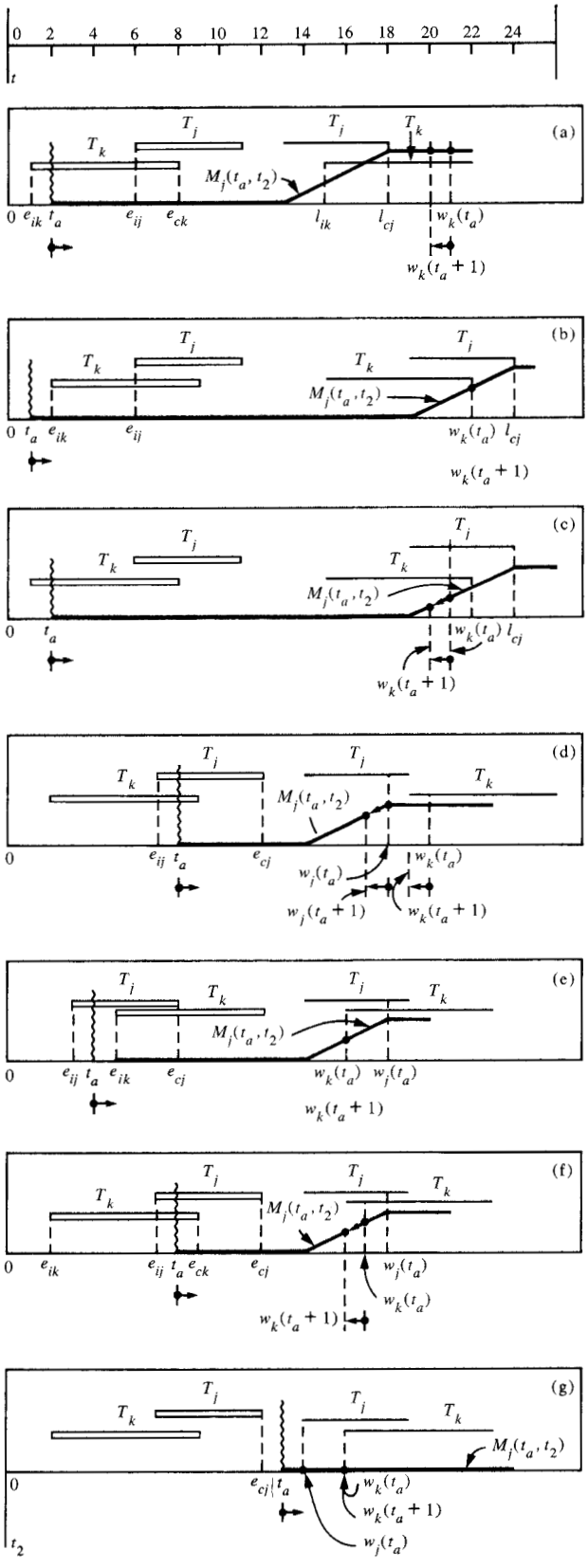

Figure 4 Effect on $M_j(t_a, w_k(t_a))$ of different positions of t_a.

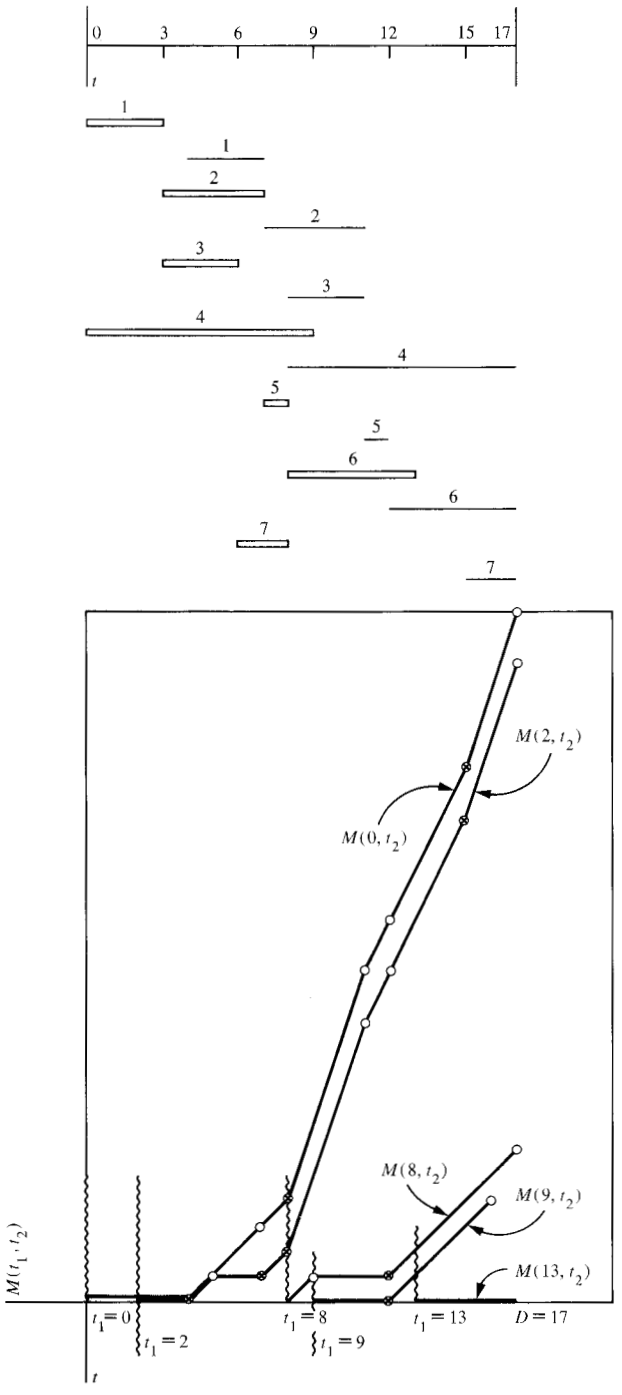

Figure 5 Example of Theorems 2, 3, and 4 using the tasks of Figure 1.

$$e_{ij} \leq t_a < e_{cj} \begin{cases} w_k(t_a) \geq w_j(t_a) \rightarrow \text{change}; & \text{Fig. 4(d)} \\ w_k(t_a) < w_j(t_a) \text{ and } t_a < e_{ik} \rightarrow \text{no change}; & \text{Fig. 4(e)} \\ w_k(t_a) < w_j(t_a) \text{ and } e_{ik} \leq t_a < e_{ck} \rightarrow \text{change}; & \text{Fig. 4(f)} \end{cases}$$

$t_a \geq e_{cj} \rightarrow$ no change. Fig. 4(g)

Figure 4(c) corresponds to condition 1, Fig. 4(d) to condition 2, and Fig. 4(f) to condition 3.

It is possible to see that the cardinality of the set of tasks that satisfy one of these conditions is constant in the regions specified by the hypothesis (i.e., between two earliest times). Therefore,

$$M(t_a, w_k(t_a)) = M_0 - (t_a - t_0)\alpha,$$

where t_0 is the smallest beginning time of the interval, and $M_0 = M(t_0, w_k(t_0))$.

The denominator $(w_k(t_a) - t_a)$ can be found by noting that

$$w_k(t_a) = w_k(t_0) - (t_a - t_0)\beta,$$

where β is 1 if $e_{ik} \leq t_a < e_{ck}$ and 0 otherwise (it is constant in the region). Therefore,

$$\begin{aligned} w_k(t_a) - t_a &= w_k(t_0) - (t_a - t_0)\beta - t_a \\ &= (w_k(t_0) - t_0) - (\beta + 1)(t_a - t_0), \end{aligned}$$

and

$$\begin{aligned} &M(t_a, w_k(t_a))/(w_k(t_a) - t_a) \\ &= [M_0 - (t_a - t_0)\alpha]/[w_k(t_0) - t_0 - (\beta + 1)(t_a - t_0)]; \end{aligned}$$

this is a monotonic function of t_a.

Corollary

$M(t_a, w_k(t_a))/(w_k(t_a) - t_a)$ is monotonic as a function of t_a in the region between two earliest times. Therefore, to obtain m_L it is sufficient to consider only the intervals that start at times corresponding to the earliest initiation or completion times of all the tasks.

It is clear that we now need to consider only $2n$ distinct values for t_a (instead of D as before). Furthermore, since whenever one task has its earliest completion time, another task has to start (i.e., it has its earliest initiation time), at most n distinct values for t_a are needed. Combining Theorems 2, 3, and 4, we see that the total number of intervals to be considered in the worst case is n^2. Notice that the number of intervals is always smaller than $D^2/2$. Therefore, the effective number of intervals to consider in the worst case is $\min[D^2/2, n^2]$. For specific cases, the required number of intervals can be considerably smaller for the same reasons as in Theorems 2 and 3.

• *Incremental calculation*

$M(t_1, t_2)$ can be calculated incrementally in a manner similar to previous work [1]. The set of the l_{ij} and $w_j(t_1)$ are first reordered and then relabeled according to their position along time. Let $h_r(t_1)$ represent either l_{ij} or $w_j(t_1)$, and let the index r represent the ordering along time (Fig. 3). Then, since $M(t_1, t_2)$ is composed of linear segments of slope k_r, as indicated in Fig. 2, it can be calculated by means of the recurrence relations:

$$M(t_1, h_{r+1}(t_1)) = M(t_1, h_r(t_1)) + k_r[h_{r+1}(t_1) - h_r(t_1)],$$

$$k_{r+1} = k_r + N_r,$$

where $N_r = S_r - E_r$, S_r is the number of tasks that have $l_{ij} = h_r$, E_r is the number of tasks for which $w_j(t_a) = h_r$, and k_r is the slope of $M(t_1, t_2)$ at time h_r^+.

The initial conditions for the recurrence relations are

$$M(t_1, h_1(t_1)) = 0,$$

$$k_0 = 0.$$

Notice that for intervals which have $t_1 = 0$, $\{h_r\} = \{l_{ij}\} \cup \{l_{cj}\}$; i.e., the list of h_r contains only latest initiation and completion times. For each $t_1 > 0$, the l_{ij} remain in the list but the l_{cj} are replaced by the $w_j(t_1)$. For every new t_1, the $w_j(t_1)$ and N_r must be recalculated and the ordered list of h_r must be set up. It is possible to organize the list of h_r in such a way that these changes are made efficiently, since only a few specific changes occur for a new t_1. For example, the w_j can be calculated in the following way.

Let t'_a be the new value for t_a; then

if $t'_a \leq e_{ij}$, $w_j(t'_a) = l_{cj}$;

if $e_{ij} \leq t'_a < e_{cj}$ $\begin{cases} t_a \leq e_{ij}, \; w_j(t'_a) = w_j(t_a) - (t'_a - e_{ij}); \\ t_a > e_{ij}, \; w_j(t'_a) = w_j(t_a) - (t'_a - t_a); \end{cases}$

if $t'_a \geq e_{cj}$, $w_j(t'_a) = l_{ij}$.

Notice that by using incremental calculation, the savings of Theorem 3 cannot be obtained, since the l_{ij} are needed as intermediate values in the calculation. Nevertheless, the incremental method may be preferred because the number of operations per interval is reduced.

• *Example*

We now present an example to illustrate the reductions due to Theorems 2, 3, and 4. Consider the graph of tasks in Fig. 1; Fig. 5 displays these reductions. The circles (empty or crossed) indicate the values of t_2 that must be considered for three specific values of t_1 ($t_1 = 0$, 2, and 8) according to Theorem 2; i.e., they correspond either to some l_{ij} or to some $w_j(t_1)$. The crossed circles identify the l_{ij} that can be eliminated from consideration according to Theorem 3 (unless an incremental method is used as in the preceding section). According to the Corollary of Theorem 4, all the intervals starting at $t_1 = 2$ can also be eliminated. The only values of t_1 which must be used for this case are 0, 3, 6, 7, 8, 9, and 13.

For this particular case, Table 1 summarizes the intervals required in the calculation of m_L, according to 1) Theorems 2, 3, and 4 combined, and 2) the incremental method, Theorems 2 and 4. The required values of t_1 according to Theorem 4 are the rows of the table, and the required values of t_2 are listed for each t_1. The values of t_2 that can be eliminated by Theorem 3 are circled. The maximum savings, obtained using Theorems 2, 3, and 4, require a total of 19 intervals, while the incremental method needs 31 intervals. These values should be compared to the worst case predictions, which are

$$\min[\lceil D^2/2 \rceil, n^2] = \min[145, 49] = 49$$

for the combination of Theorems 2, 3, and 4, and

$$\min[\lceil D^2/2 \rceil, 2n^2] = \min[145, 98] = 98$$

for the incremental method. The method described in [1] requires the consideration of $\lceil D^2/2 \rceil = 145$ intervals.

Table 1 Intervals required in the calculation of m_L.

t_1	t_2	*Number of intervals*	
		Theorems 2, 3, 4	*Incremental method*
0	④ 7 ⑧11 12 ⑮17	4	7
3	7 ⑧11 12 ⑮17	4	6
6	⑦ 8 11 12 ⑮17	4	6
7	⑧10 ⑪12 ⑮16 17	4	7
8	9 ⑫17	2	3
9	⑫16	1	2
13		0	0
	Total	19	31

Summary

Four theorems have been presented which allow more efficient calculation of the lower bound expression presented by Fernández and Bussell [2]. These results improve on the computation methods already given in [1]. Theorem 1 allows a reduction in the number of intervals to be considered based on the determination of the intervals for which $M(t_1, t_2) = 0$. A small modification of the algorithm given in [1] is required to take advantage of this result. Theorems 2 and 4 can be combined to provide a significant reduction—from $D^2/2$ intervals to $\min[D^2/2, 2n^2]$ intervals. If direct calculation is used, instead of incremental computation as proposed in [1], Theorem 3 allows a further reduction to $\min[D^2/2, n^2]$ intervals. However, the operations needed in this case to keep track of the graph's structural properties make the savings less significant except for graphs in which D is large compared to n.

Acknowledgment

The authors thank the reviewers for detailed comments that have helped to improve the presentation of this paper.

References

1. E. B. Fernández and T. Lang, "Computation of Lower Bounds for Multiprocessor Schedules," *IBM J. Res. Develop.* **19,** 435 (1975).
2. E. B. Fernández and B. Bussell, "Bounds on the Number of Processors and Time for Multiprocessor Optimal Schedules," *IEEE Trans. Comput.* **C-22,** 745 (1973).
3. E. B. Fernández and T. Lang, "Scheduling as a Graph Transformation," presented at the ORSA/TIMS National Meeting, Las Vegas, Nevada, October 1975; *IBM J. Res. Develop*, **20,** 551 (1976).
4. C. V. Ramamoorthy, K. M. Chandy, and M. J. González, "Optimal Scheduling Strategies in a Multiprocessor System," *IEEE Trans. Comput.* **C-21,** 137 (1972).
5. T. L. Adam, K. M. Chandy, and J. R. Dickson, "A Comparison of List Schedules for Parallel Processing Systems," *Commun. ACM* **17,** 685 (1974).

Received May 14, 1976

T. Lang is with the Computer Science Department, University of California, Los Angeles, California 90024; E. B. Fernández is located at the IBM Data Processing Division Scientific Center, 9045 Lincoln Boulevard, Los Angeles, California 90045.