A Parametric Worst Case Analysis for a Machine Scheduling Problem

Paul Mireault,
Rakesh Vohra
and
James B. Orlin

WP# 3603-93-MSA July 1993

A Parametric Worst Case Analysis for a Machine Scheduling Problem

Paul Mireault
University of Montreal

James B. Orlin
Rakesh V. Vohra*
Sloan School of Management
M.I.T.

July 1993

ABSTRACT

We consider the problem of minimizing the makespan when scheduling tasks on two uniform parallel machines, where one machine is q times as efficient on each task as is the other. We compute the maximum relative error of the LPT (largest processing time first) heuristic as a function of q. In the special case that the two machines are identical (q = 1), our problem and heuristic are identical to the problem and heuristic analyzed by Graham [1969].

* On leave from the department of Management Science, Ohio State University.

INTRODUCTION

Using the common notation of scheduling theory, we consider the problem $Q2||C_{MAX}$. This is the problem of scheduling each of n tasks on either of two uniform parallel machines, where the first machine is q times as efficient as the second, so as to minimize the makespan. The processing time for task j is p(j) on machine M1 and qp(j) on machine M2. This problem is well known to be NP-hard. As a consequence, much attention has focused on heuristics for solving this problem. The most well known of these is the Largest Processing Time first (LPT) heuristic. We describe it below.

LPT Heuristic

```
BEGIN
    order the tasks so that p(1) ≥ p(2) ≥.....≥ p(n);
    T(1) := 0;
    T(2) := 0;
    FOR j = 1 TO n DO
        BEGIN
            IF T(1) + p(j) < q(T(2) + p(j)) THEN T(1) := T(1) + p(j)
            ELSE T(2) := T(2) + p(j);
        END
END
```

LPT sequentially assigns the tasks in such a way that task j is assigned to the machine on which it would finish first.

An instance of our scheduling problem will be denoted by (P,q), where P is a vector of processing times and q is the efficiency factor. The makespan returned by LPT for the instance (P,q) will be denoted L(P,q) and it is equal to max{T(1), qT(2)}. Let OPT(P,q) denote the minimum possible makespan for instance (P,q). Then, the relative error for LPT with respect to an instance (P,q) is:

$$\text{REL}(P,q) \quad = \quad [L(P,q) - \text{OPT}(P,q)]/\text{OPT}(P,q).$$

The worst case relative error for LPT is R(q) = sup{REL(P,q): P}. Thus R(q) is the supremum of all relative errors for LPT for a fixed value of q. In this paper we show how to compute R(q).

RELATED RESULTS

In the last 20 years a number of papers have focused on guaranteed accuracy heuristics for machine scheduling problems. For a survey of such papers see Lawler et. al. [1993]. The analysis in this paper was motivated primarily by the classic paper by Graham [1969]. In that paper, Graham analyzed the special case in which q = 1 as generalized to the m machine case. Dobson [1984] and Friesen [1987] have studied the worst case relative error of LPT in the m machine case, where the machines have varying speeds. Their results do not subsume ours as they are concerned with the maximum relative error over all task sizes and machine efficiencies. Put another way, for the case of two machines, they compute $\sup\{R(q): q \geq 0\}$.

THE APPROACH

To understand our approach to computing R(q), fix a value of q. The standard method, first used by Graham (and everyone since then), is to start with a conjecture about the value of R(q). Usually, the conjecture is arrived at by experimenting with small problem instances. To prove the conjecture assume it is false. This implies the existence of an instance (P,q) that violates the conjecture. From among all such counter-examples pick the one with the fewest number of tasks. Identify some properties that this minimal counter-example must satisfy; such as the maximum number of tasks it can contain. On the basis of these properties show that such a minimal counter-example cannot exist. If one succeeds in this last venture, well and good; if not, back to the drawing board. Imagine now the difficulty involved in using this approach to compute R(q) for all $q > 0$.

The novelty of our approach is to use Linear Programming (LP) as part of a systematic way of generating (true) conjectures about the value of R(q) for each $q > 0$. We start by formulating an LP whose optimal objective function value provides an upper bound on R(q). This bound is then used to show that we can restrict the search for a minimal counter-example to instances involving no more than 5 jobs. We then use linear programming on these smaller instances to generate the instances that maximize REL(P,q). Subsequently we prove the correctness of the conjectures about R(q) generated by this LP approach.

Before proceeding, some words on the nature of R(q) are in order. First, it is not hard to see that as q gets very large, R(q) tends to zero. Thus one might be tempted to conclude that R(q) will be monotonically decreasing. Not so. It is not even unimodal. In fact its derivative changes sign in 7 places for $q \geq 1$. Finally, since $R(q) = R(1/q)$ for $q > 0$ we restrict our attention to cases where $q \geq 1$.

A SIMPLE UPPER BOUND

In order to simplify the search for worst case instances of (P,q), we make some simplifying assumptions. For a given instance of (P,q), let T(1,j) and T(2,j) denote the cumulative processing time of tasks allocated to machines M1 and M2 by LPT after task j has been assigned. (Thus the total processing time on M2 is qT(2,n)). These assumptions are as follows.

A1) $p(1) \geq p(2) \geq \ldots \geq p(n) > 0$.

A2) $OPT(P,q) = 1$.

A3) $T(2,n-1) \geq p(n)$.

A4) $T(1,n-1) \leq q(T(2,n-1) + p(n))$.

A5) $qT(2,n-1) \leq T(1,n-1) + p(n)$.

A6) $T(1,n-1) + T(2,n-1) + p(n) \leq 1 + 1/q$.

A1 is merely a restatement that the processing times are ordered from largest to smallest by LPT. A2 follows from the fact that R(q) is scale independent, i.e., if all processing times are multiplied by the same number, R(q) does not change. Notice that A2 implies that REL(P,q) = L(P,q) - 1. To see that A3 is valid, assume not. Then T(2,n-1) = 0. Hence, LPT assigns all of the first n - 1 jobs to M1. A simple interchange argument shows that L(P,q) = OPT(P,q) in this case. A4 and A5 follow from the fact that in an LPT schedule generating the worst case relative error, task n must be the last task to be completed. If not, we could delete task n without decreasing the relative error. Thus, any instance (P,q) that violates A4 or A5 can be transformed into another instance with at least as much relative error by eliminating task n.

To see A6, let OPT(i) denote the sum of the processing times of the tasks assigned to machine i in an optimal schedule. From A2 and the definition of OPT(P,q) we see that $OPT(1) \leq 1$ and $qOPT(2) \leq 1$. Combining these two inequalities and using the fact that OPT(1) + OPT(2) = T(1,n-1) + T(2,n-1) + p(n), we get A6.

In view of A1 - A6 it is not hard to see that the optimal objective function value of the following linear program (LP1) provides an upper bound on R(q) for all q > 0.

$$Z(q) = \max \quad H - 1 \tag{1.1}$$

subject to

$$H \leq T(1,n-1) + p(n) \tag{1.2}$$

$$H \leq q(T(2,n-1) + p(n)) \tag{1.3}$$

$$T(1,n-1) + T(2,n-1) + p(n) \leq 1 + 1/q \tag{1.4}$$

$$p(n) \leq T(2,n-1) \tag{1.5}$$

$$(n-1)\,p(n) \leq T(1,n-1) + T(2,n-1) \tag{1.6}$$

Proposition 1: $R(q) \leq 1/(2q + 1)$ for all $q \geq 1$.

Proof: If $Z(q) \leq 1/(2q+1)$, then we are done. Add 2q times (1.2) to (1.3), then add (1.5) and add (2q + 2) times (1.4) to produce:

$$\begin{aligned} (2q + 1)H &\leq (2q + 2) \\ \Rightarrow \quad H - 1 &\leq 1/(2q+1). \end{aligned}$$

Since this is true for any feasible H, $Z(q) \leq 1/(2q+1)$. Δ

It is interesting to note that this bound for Z(q) is tight for $q \geq (n - 1)/2$ as can be seen from the following feasible solution to LP1:

$$
\begin{aligned}
H &= 1 + 1/(2q+1),\\
T(1,n-1) &= (2q-1)(q+1)/[q(2q+1)],\\
T(2,n-1) &= (q+1)/[q(2q+1)],\\
p(n) &= (q+1)/[q(2q+1)].
\end{aligned}
$$

For $q < (n-1)/2$, the following is the optimal solution to LP1:

$$
\begin{aligned}
H &= 1 + 1/n,\\
T(1,n-1) &= 1 - 1/nq,\\
T(2,n-1) &= 1/q - 1/n,\\
p(n) &= 1/n.
\end{aligned}
$$

EVALUATING R(q)

In this section we show how to systematically generate conjectures about the value of R(q) for each q. Our method is implicit enumeration based on the following two observations.

FACT 1: If we knew both the LPT and Optimal assignments, we could solve for the n-vector P by linear programming.

FACT 2: The number of different combinations of LPT and optimal assignments for a schedule of n tasks is less than 2^{2n}.

To illustrate a linear program as mentioned in FACT 1, suppose that n = 4. Suppose also that LPT assigns tasks 1, 3 and 4 to M1 and task 2 to M2. Suppose further that the optimum schedule has tasks 2, 3 and 4 assigned to M1 and task 1 to M2.

Then we may find the worst case vector P as follows.

MAXIMIZE H - 1

SUBJECT TO

$$
\begin{aligned}
T(1,1) &= p(1)\\
T(1,2) &= p(1)\\
T(1,3) &= p(1) + p(3)\\
T(1,4) &= p(1) + p(3) + p(4)\\
T(2,1) &= 0\\
T(2,2) &= p(2)\\
T(2,3) &= p(2)\\
T(2,4) &= p(2)\\
qT(2,2) &\le T(1,1) + p(2)\\
T(1,3) &\le qT(2,2) + qp(3)
\end{aligned}
$$

$$
\begin{aligned}
T(1,4) &\le qT(2,3) + qp(4) \\
p(2) + p(3) + p(4) &\le 1 \\
qp(1) &\le 1 \\
p(1) &\ge p(2) \\
p(2) &\ge p(3) \\
p(3) &\ge p(4) \\
p(4) &\ge 0.
\end{aligned}
$$

The variables and constraints of this linear program may be simplified; we list the entire set because it is the system we derive mechanically for use in a matrix generator.

Actually, we want to solve the above linear program for all values of $q \ge 1$. Thus, we want to solve a parametric linear program in which the constraint matrix varies linearly with q. In general, there is no simple approach to solving such parametric linear programs; however for the parametric linear programs that arise in this analysis , it is usually true that very simple methods can calculate the breakpoints; moreover, it is rare that there are more than four different optimal bases.

Our algorithm may now be described as follows.

```
BEGIN
      for all possible LPT schedule assignments x
      for all possible optimal schedule assignments y DO
    BEGIN
          find the worst case processing times P as a function of x, y, q
           by solving a parametric linear program for all q ≥ 1.
     END
END
```

Of course, the above search is overly exhaustive. In practice we can cut down on the computation time considerably. For example (as shown in the next section) we prove that it is sufficient to consider only the cases $3 \le n \le 5$. Further manipulations allow one to reduce even more the set of assignments to be considered.

The results of our search are included in Figure 1, the value of R(q) for $q \ge 1$. In the next section we prove the correctness of this curve. This curve is defined by the function f(q) where:

$$
\begin{aligned}
f(q) &= q/3(q+1) && 1 \le q < (1+\sqrt{37})/6 \\
&= q-1 && (1+\sqrt{37})/6 \le q < (1+\sqrt{17})/4 \\
&= (2-q)/2q && (1+\sqrt{17})/4 \le q < \sqrt{2} \\
&= 1/(2q+2) && \sqrt{2} \le q < (1+\sqrt{33})/4 \\
&= (q-1)/(q+2) && (1+\sqrt{33})/4 \le q < (1+\sqrt{7})/2 \\
&= (3-q)/3q && (1+\sqrt{7})/2 \le q < 2 \\
&= 1/(2q+2) && 2 \le q < q_1 \\
&= (2q^2-4q+1)/(4q-1) && q_1 \le q \le q_2 \\
&= 1/(2q+1) && q_2 \le q.
\end{aligned}
$$

Here q_1 is the largest real root of $4x^3 - 8x^2 - 10x + 1 = 0$ and q_2 is the unique real root of $2x^3 - 3x^2 - 3x + 1 = 0$. In the next section we prove that $R(q) = f(q)$ for all $q \ge 1$.

THE PROOF

The following simple lemma will prove quite useful.

Lemma 2: $L(P,q) - OPT(P,q) \le qp(n)/(q+1)$.

Proof: In LP1 add $q/(q+1)$ times (1.2) to $1/(q+1)$times (1.3) and add the sum to (1.4). This results in the following inequality:

$$H - 1 \quad \le \quad qp(n)/(q+1).$$

Using A2 and the fact that $L(P,q) \le H$ we obtain the result. Δ

Proposition 3: If $n \ge 6$, then $R(q) \le f(q)$.

Proof: If $n \ge 6$, then the optimal schedule assigns at least 3 tasks to M2 or 4 tasks to M1. In the first case, $1 = OPT(P,q) \ge 3qp(n)$. This implies, by Lemma 2, that $REL(P,q) \le 1/(3q+3) \le f(q)$ for all $q \ge 1$. In the second case, $OPT(P,q) \ge \max\{4p(n), 2qp(n)\}$ and so, $REL(P,q) \le \min\{q/(4q+4), 1/(2q+2)\} \le f(q)$ for all $q \ge 1$. Δ

Notice that the first part of the proof to Proposition 3 can be generalized to read:

Proposition 4: If an optimal schedule assigns at least 3 tasks to M2, then $R(q) \le f(q)$.

Proof: Obvious. Δ

Proposition 5: $R(q) \leq f(q)$ for $n \leq 2$.

Proof: For $n = 1, 2$, $L(P,q) = OPT(p,q)$. Δ

Propositions 3, 4 and 5 allow us to restrict our attention to instances involving between 3 and 5 tasks where the optimal schedule assigns at most two tasks to M2. In what follows, recall that we can assume that $p(n)$ is the last task to complete in an LPT schedule and $T(2,n-1) > 0$. Hence, if $T(1,n-1) + p(n) \leq q(T(2,n-1) + p(n))$, then, $L(P,q) = T(1,n-1) + p(n)$; otherwise, $L(P,q) = q(T(2,n-1) + p(n))$.

Proposition 6: For $n = 3$, $R(q) \leq f(q)$.

Proof: If the LPT schedule assigns tasks 1 and 2 to M1, then, $L(P,q) = OPT(P,q)$. Hence, we may assume that task 2 is assigned to M2 in an LPT schedule. If task 3 is assigned by LPT to M1, then, $L(P,q) = p(1) + p(3) \leq q(p(2) + p(3))$. If task 3 is assigned by LPT to M2, then, $L(P,q) = q(p(2) + p(3)) \leq p(1) + p(3)$. In all cases $L(P,q) = \min\{p(1) + p(3), q(p(2) + p(3))\}$.

If task 1 is assigned to M1 in an optimal schedule, then, $OPT(P,q) \geq p(1) + p(3)$ or $OPT(P,q) \geq q(p(2) + p(3))$. Thus, in either case, $OPT(P,q) \geq L(P,q)$. Hence we may assume that task 1 is assigned to M2 and tasks 2 and 3 are assigned to M1 in an optimal schedule, i.e., $OPT(P,q) = \max\{qp(1), p(2) + p(3)\}$. So,

$$\begin{aligned} L(P,q) - OPT(P,q) &\leq q(p(2) + p(3)) - (p(2) + p(3)) \\ &= (q-1)(p(2) + p(3)) \\ \Rightarrow \quad REL(P,q) &\leq q - 1. \end{aligned}$$

Also,

$$\begin{aligned} L(P,q) - OPT(P,q) &\leq p(1) + p(3) - [qp(1) + p(2) + p(3)]/2 \\ &\leq (1 - q/2)p(1) \\ \Rightarrow \quad REL(P,q) &\leq (2-q)/2q. \end{aligned}$$

Therefore, $REL(P,q) \leq \min\{q - 1, (2 - q)/2q\} \leq f(q)$ for all $q \geq 1$ and $q \leq 2$. When $q \geq 2$, $OPT(P,q) = \max\{qp(1), p(2) + p(3)\} \geq 2p(1) \geq p(1) + p(3) \geq \min\{p(1) + p(3), qp(2) + p(3)\} = L(P,q)$, a contradiction. Δ

Proposition 7: Let $n = 4$. If both LPT and the optimal schedule assign exactly two tasks to each machine, then, $R(q) \leq f(q)$.

Proof: If LPT assigns task 4 to M1, then, $L(P,q) = p(1) + p(4) \le OPT(P,q)$, and the LPT schedule is optimal. If LPT assigns both tasks 3 and 4 to M2, then, $LPT(P,q) = q(p(3) + p(4)) \le OPT(P,q)$, and the LPT schedule is optimal. We conclude that LPT assigns tasks 2 and 4 to M2, and $L(P,q) = q(p(2) + p(4))$.

Since the optimal schedule assigns two tasks to M2, it must assign tasks 3 and 4, otherwise the LPT schedule is optimal. Hence, $OPT(P,q) = \max\{p(1) + p(2), q(p(3) + p(4)\}$. Then,

$$\begin{aligned} L(P,q) - OPT(P,q) &\le q(p(2) + p(4)) - (p(2) + p(4)) \\ &\le (q-1)OPT(P,q). \end{aligned}$$

Thus, $REL(P,q) \le q - 1$. By Lemma 2, $L(P,q) - OPT(P,q) \le qp(4)/(q+1)$. Since, $OPT(P,q) \ge 2p(4)$, it follows that $REL(P,q) \le 1/(2q+2)$. Thus, $REL(P,q) \le \min\{q - 1, 1/(2q+2)\} \le f(q)$ for all $q \ge 1$.

Δ

Proposition 8: Let $n = 4$. If the optimal schedule assigns exactly two tasks to each machine and LPT assigns three tasks to one machine, then, $R(q) \le f(q)$.

Proof: We divide the proof into two cases.

Case 1 Task 4 is assigned to M1 by LPT.

We observe first that task 2 is assigned to M2 in an LPT schedule. Otherwise, $T(2,3) \le p(3)$ and so $L(P,q) \le q(p(3) + p(4)) \le OPT(P,q)$. Hence, $L(P,q) = p(1) + p(3) + p(4)$. If the optimal schedule assigns at least one of tasks 1 or 2 to M2, then, $OPT(P,q) \ge q(p(2) + p(4)) \ge L(P,q)$ by definition of LPT. Thus, in an optimal schedule tasks 1 and 2 are assigned to M1. We will show that $REL(P,q) \le \min\{q-1, 1/(2q+2)\}$.

Notice that $L(P,q) = p(1) + p(3) + p(4) \le q(p(2) + p(4))$ by LPT. Also, $OPT(P,q) \ge p(1) + p(2) \ge p(2) + p(4)$. Hence, $REL(P,q) \le q - 1$. To get the second bound we note that $OPT(P,q) \ge 2qp(4)$ and so by Lemma 2, $REL(P,q) \le 1/(2q+2)$.

Case 2 Task 4 is assigned to M2 by LPT

If tasks 1, 2 and 3 are assigned by LPT to M1, then the LPT schedule is optimal.

If LPT assigns task 1 to M1 and tasks 2, 3, and 4 to M2, then,

$$\begin{aligned} L(P,q) &= q(T(2, 3) + p(4)) \\ &\le T(1, 3) + p(4) \\ &= p(1) + p(4) \\ &\le OPT(P,q) \end{aligned}$$

because in an optimal schedule two tasks are assigned to each machine. Δ

Proposition 9: Let $n = 4$. If the optimal schedule assigns three tasks to one machine and LPT assigns two tasks to each machine, then, $R(q) \le f(q)$.

Proof: Clearly, the optimal schedule will assign the three tasks to M1.

Case 1 Task 4 is assigned to M1 by LPT

In this case $L(P,q) = p(1) + p(4)$. If task 1 is assigned to M1 in an optimal schedule, $OPT(P,q) \geq L(P,q)$, a contradiction. Hence, task 1 is assigned to M2 in an optimal schedule. Thus, $p(1) \leq OPT(P,q)/q$ and $p(4) \leq OPT(P,q)/3$. Hence,

$$\begin{aligned} L(P,q) - OPT(P,q) &\leq OPT(P,q)/q + OPT(P,q)/3 - OPT(P,q) \\ &\leq (1/q - 2/3)OPT(P,q) \\ \Rightarrow \quad REL(P,q) &\leq (1/q - 2/3). \end{aligned}$$

By, Lemma 2, $REL(P,q) \leq q/(3q+3)$. Therefore, $REL(P,q) \leq \min\{q/(3q+3), (1/q - 2/3)\} \leq f(q)$ for $q \geq 1.5$. When $1 \leq q \leq 1.5$, $REL(P,q) \leq q/(3q+3) \leq f(q)$.

Case 2 Task 4 is assigned to M2 by LPT

Suppose first that task 3 is assigned to M1 by LPT. Then, $L(P,q) = q(p(2) + p(4)) \leq p(1) + p(3) + p(4)$. If task 1 is assigned to M1 in an optimal schedule, then, $OPT(P,q) \geq L(P,q)$. Hence, $OPT(P,q) = \max\{qp(1), p(2) + p(3) + p(4)\}$. Thus,

$$\begin{aligned} L(P,q) - OPT(P,q) &\leq p(1) + p(3) + p(4) - OPT(P,q) \\ &\leq OPT(P,q)/q + 2OPT(P,q)/3 - OPT(P,q) \\ &= (3-q)OPT(P,q)/3q \\ \Rightarrow \quad REL(P,q) &\leq (3 - q)/3q. \end{aligned}$$

Lemma 2 implies that $REL(P,q) \leq q/(3q+3)$. Hence, $REL(P,q) \leq \min\{q/(3q+3), (3-q)/3q\} \leq f(q)$ for $3 \geq q \geq 1$. If $q > 3$, $OPT(P,q) \geq 3p(1) \geq p(1) + p(3) + p(4) \geq L(P,q)$, a contradiction.

Now suppose that task 2 is assigned to M1 by LPT. Then, $L(P,q) = q(p(3) + p(4))$. Since task 2 was assigned to M1, it follows that $qp(2) \geq p(1) + p(2) \Rightarrow q \geq 2$. We consider two subcases.

Subcase 1 Task 1 is assigned to M1 in an optimal schedule.

If task 2 is also assigned to M1 in an optimal schedule, then, $L(P,q) \leq OPT(P,q)$. Hence, $OPT(P,q) = \max\{qp(2), p(1) + p(3) + p(4)\}$. If $p(1) \geq OPT(P,q)/q$, then

$$\begin{aligned} L(P,q) &= q(p(3) + p(4)) \\ &\leq q(OPT(P,q) - p(1)) \\ &\leq (q - 1)OPT(P,q) \\ \Rightarrow \quad REL(P,q) &\leq q - 2 \leq f(q) \qquad \text{for } 2 \leq q \leq q_2. \end{aligned}$$

If $p(1) < OPT(P,q)/q$, $p(2) < OPT(P,q)/q$ as well. Hence, $OPT(P,q) = p(1) + p(3) + p(4)$. Since task 4 was assigned to M2 by LPT,

$$\begin{aligned} q(p(3)+p(4)) &\le p(1)+p(2)+p(4) \\ &\le 2p(1)+p(4). \end{aligned}$$

Now, $p(1) = OPT(P,q) - p(3) - p(4) \Rightarrow$

$$\begin{aligned} q(p(3)+p(4)) &\le 2(OPT(P,q) - p(3) - p(4)) + p(4) \\ \Rightarrow \quad (q+2)p(3) + (q+1)p(4) &\le 2OPT(P,q). \end{aligned}$$

Also, $p(3) \ge p(4) \Rightarrow$

$$(q+1)p(3) + (q+2)p(4) \le 2OPT(P,q).$$

Adding these last two inequalities we obtain:

$$\begin{aligned} (2q+3)(p(3)+p(4)) &\le 4OPT(P,q) \\ \Rightarrow \quad L(P,q) &\le 4qOPT(P,q)/(2q+3) \\ \Rightarrow \quad REL(P,q) &\le (2q-3)/(2q+3) \le 1/(2q+2) \le f(q) \qquad \text{for } 2 \le q \le q_1. \end{aligned}$$

To obtain the bound in the region $q_1 \le q \le q_2$, we observe that $qp(2) \le p(1) + p(2)$, since task 2 was assigned by LPT to M1. Hence:

$$\begin{aligned} q(p(3)+p(4)) &\le p(1)+p(2)+p(4) \\ &\le qp(1)+p(4) \\ &\le q(OPT(P,q) - p(3) - p(4)) + p(4) \\ \Rightarrow \quad 2qp(3) + (2q-1)p(4) &\le qOPT(P,q). \end{aligned}$$

As $p(3) \ge p(4)$, we also get:

$$(2q-1)p(3) + 2qp(4) \le qOPT(P,q).$$

Adding these last two inequalities produces:

$$\begin{aligned} (4q-1)(p(3)+p(4)) &\le 2qOPT(P,q) \\ \Rightarrow \quad L(P,q) &\le 2q^2OPT(P,q)/(4q-1) \\ \Rightarrow \quad REL(P,q) &\le (2q^2-4q+1)/(4q-1). \end{aligned}$$

<u>Subcase 2</u> Task 1 is assigned to M2 in an optimal schedule.
If so, $qp(1) \le OPT(P,q)$ and the previous argument applies. Δ

Proposition 10: Let $n = 4$. If the optimal and LPT schedule each assign three tasks to a single machine, then, $R(q) \le f(q)$.

Proof: By proposition 4 and the hypothesis of the proposition, in an optimal schedule, M2 is never assigned more than one task.

Case 1 $T(1, 3) \le qT(2, 3)$

Thus $L(P,q) = T(1, 3) + p(4)$. By A6, $OPT(P,q) \ge q(T(1, 3) + T(2, 3) + p(4))/(q+1)$. Hence,

$$\begin{aligned} L(P,q) - OPT(p,q) &\le (T(1, 3) + p(4))/(q+1) - qT(2, 3)/(q+1) \\ &\le p(4)/(q+1). \end{aligned}$$

Since, $OPT(P,q) \ge 3p(4)$, it follows that $REL(P,q) \le 1/(3q+3) \le f(q)$ for all $q \ge 1$.

Case 2 $T(1, 3) > qT(2, 3)$

We further divide this case into a number of subcases.

Subcase 1 $T(1, 3) + p(4) \le q(T(2, 3) + p(4))$

In this case, LPT assigns task 4 to M1. Suppose first that LPT assigns task 3 to M1 and task 2 to M2, i.e., $L(P,q) = p(1) + p(3) + p(4)$. If task 1 is assigned to M1 in an optimal schedule then, $OPT(P,q) \ge p(1) + p(3) + p(4) = L(P,q)$. So, we may assume that $OPT(P,q) = \max\{qp(1), p(2) + p(3) + p(4)\}$. We will now show that $R(q) \le \min\{q-1, 2(q-1)/q^2\}$ which is $\le f(q)$ for $q \ge 1$.

Observe first that $L(P,q) - OPT(P,q) \le p(1) - p(2)$. Since task 2 was assigned to M2 and task 3 to M1 by LPT, $p(1) + p(3) \le q(p(2) + p(3))$. Hence,

$$\begin{aligned} p(1) - p(2) &= p(1) + p(3) - p(2) - p(3) \\ &\le (q-1)(p(2) + p(3)) \\ \Rightarrow \quad REL(P,q) &\le (q-1). \end{aligned}$$

Secondly, as task 4 was assigned by LPT to M1,

$$\begin{aligned} q(p(2) + p(4)) &\ge p(1) + p(3) + p(4) \\ \Rightarrow \quad p(2) &\ge (p(1) + p(3))/q + (1/q - 1)p(4). \end{aligned}$$

So,

$$\begin{aligned} L(P,q) - OPT(P,q) &\le p(1) - p(2) \\ &\le (1 - 1/q)(p(1) + p(4)) - p(3)/q \\ &\le 2(1 - 1/q)p(1). \end{aligned}$$

Hence, $REL(P,q) \leq 2(1 - 1/q)/q^2$.

Suppose now that LPT assigns task 2 to M1 and task 3 to M2. Then, $L(P,q) = p(1) + p(2) + p(4)$. If task 1 is assigned to M2 in an optimal schedule, then, by an argument similar to the one above we deduce that $R(q) \leq \min\{q-1, 2(q-1)/q^2\}$. If task 1 and task 2 are assigned to M1, then $OPT(P,q) \geq L(P,q)$. Hence, $OPT(P,q) = \max\{qp(2), p(1) + p(3) + p(4)\}$. Then, $L(P,q) - OPT(P,q) \leq p(2) - p(3)$.

Since task 3 was assigned to M2 by LPT, $qT(2, 3) = qp(3) \geq T(1, 3) + p(4) = p(1) + p(2) + p(3) \geq p(2)$. Hence, $p(2) - p(3) \leq (q-1)p(3)$, i.e, $REL(P,q) \leq q - 1$. As task 4 was assigned to M1 by LPT,

$$\begin{aligned} q(p(3) + p(4)) &\geq p(1) + p(2) + p(4) \\ \Rightarrow \quad p(3) &\geq (p(1) + p(2))/q + (1/q - 1)p(4). \end{aligned}$$

Hence, $L(P,q) - OPT(P,q) \leq p(2) - p(3) \leq 2(1 - 1/q)p(2) \Rightarrow REL(P,q) \leq 2(1-1/q)/q^2$.

<u>Subcase 2</u> $T(1, 3) + p(4) > q(T(2, 3) + p(4))$

In this case task 4 will be assigned to M2 by LPT. If task 4 is the only task on M2, $L(P,q) = OPT(P,q)$. Hence, LPT must assign three tasks to M2, i.e., $L(P,q) = q(p(2) + p(3) + p(4))$. Since LPT assigns task's 2 and 3 to M2,

$$\begin{aligned} qp(2) &\leq p(1) + p(2), \\ q(p(3) + p(2)) &\leq p(1) + p(3). \end{aligned}$$

Suppose that task 2 is the unique task assigned to M2 in an optimal schedule. Then we can reduce the makespan of the optimal schedule by moving task 4 to M2, because,

$$\begin{aligned} q(p(2) + p(4)) &\leq q(p(2) + p(3)) \\ &\leq p(1) + p(3). \end{aligned}$$

The same argument applies if task 3 or 4 is the only task on M2 in an optimal schedule. Hence, we may assume that $OPT(P,q) = \max\{qp(1), p(2) + p(3) + p(4)\}$. Thus, $L(P,q) - OPT(P,q) \leq (q-1)$. Notice also, that, $L(P,q) \leq p(1) + p(4) \leq 2p(1)$. So, $L(P,q) - OPT(P,q) \leq (2-q)p(1)$. Thus, $R(q) \leq \min\{q-1, (2-q)/q\} \leq f(q)$ for $2 \geq q \geq 1$. For $q \geq 2$, $OPT(P,q) \geq qp(1) \geq 2p(1) \geq p(1) + p(4) \geq L(P,q)$, contradiction.
Δ

Proposition 11: If $n = 5$, then, $R(q) \leq f(q)$.

Proof: By proposition 4 M1 has at least three tasks assigned to it in an optimal schedule. So, $OPT(P,q) \geq 3p(5)$. Hence, by Lemma 2, $R(q) \leq q/(3q+3)$. Notice also, that in an optimal schedule, either two tasks are assigned to M2 or four tasks to M1. If two tasks are assigned to M2, $OPT(P,q) \geq 2qp(5)$. By lemma 2, $R(q) \leq 1/(2q+2)$. So, for this case, $R(q) \leq \min\{q/(3q+3), 1/(2q+2)\} \leq f(q)$ for $q \geq 1$. Thus, we restrict our attention to the case where an optimal schedule assigns four tasks to M1. In this case

$OPT(P,q) \geq 4p(5)$, i.e, $R(q) \leq \min\{q/(4q+1), 1/(2q+1)\} \leq f(q)$ for $1 \leq q \leq 2$. Hence, we need only consider the case when $2 \leq q \leq q_2$. We will assume that the processing times are normalized so that $OPT(P,q) = 1$.

Since there are four tasks assigned to M1 in an optimal schedule, $1 \geq p(2) + 3p(5) \Rightarrow p(5) \leq [1 - p(2)]/3$. If $p(2) > 1 - 3/2q$, then, $p(5) \leq 1/2q$. So, by Lemma 2, $REL(P,q) \leq 1/(2q+2)$. Therefore we may assume that $p(2) \leq 1 - 3/2q$.

Of the first four tasks, at most one will be assigned to M2 by LPT. Therefore, $L(P,q) \leq q(p(2) + p(5))$ and so

$$\begin{aligned} L(P,q) - 1 &\leq q(1 - 3/2q + 1/4) - 1 \\ &= (5q - 10)/4 \\ &\leq f(q) \qquad\qquad \text{for } 2 \leq q \leq q_2 . \quad \Delta \end{aligned}$$

To complete the proof it sufficient to exhibit an instance (P,q) for each q, such that $REL(P,q) = f(q)$. This we now do; note that these examples are not scaled so that $OPT(P,q) = 1$.

$\underline{1 \leq q < (1 + \sqrt{37})/6}$

$p(1) = (2q+1)/(q+1)$

$p(2) = (3 + 2q - 2q^2)/q(q+1)$

$p(3) = p(4) = p(5) = 1$

$\underline{(1 + \sqrt{37})/6 \leq q < \sqrt{2}}$

$p(1) = 2/q$

$p(2) = p(3) = 1$

$\underline{\sqrt{2} \leq q < (1 + \sqrt{33})/4}$

$p(1) = (2q^2 + q - 2)/(q+1)$

$p(2) = (q+2)/(q+1)$

$p(3) = p(4) = 1$

$\underline{(1 + \sqrt{33})/4 \leq q < (1 + \sqrt{7})/2}$

$p(1) = (q+2)/(q^2-1)$

$p(2) = (2+2q-q^2)/(q^2-1)$

$p(3) = p(4)$

$\underline{(1 + \sqrt{7})/2 \leq q < 2}$

$p(1) = 3/q$

$p(2) = p(3) = p(4) = 1$

$\underline{2 \le q < q_1}$

$p(1) = (2q^2+q-2)/(q+1)$

$p(2) = (q+2)/(q+1)$

$p(3) = p(4) = 1$

$\underline{q_1 \le q < q_2}$

$p(1) = p(2) = (2q-1)/(2q+3)$

$p(3) = p(4) = 2/(2q+3)$

$\underline{q_2 \le q < (3+\sqrt{33})/4}$

$p(1) = (2q+1)/(q+1)$

$p(2) = (2q^2-q-2)/(q+1)$

$p(3) = p(4) = 1$

$\underline{(3+\sqrt{33})/4 \le q}$

$p(1) = (2q^2-q-2)/(q+1)$

$p(2) = (2q+1)/(q+1)$

$p(3) = p(4) = 1$

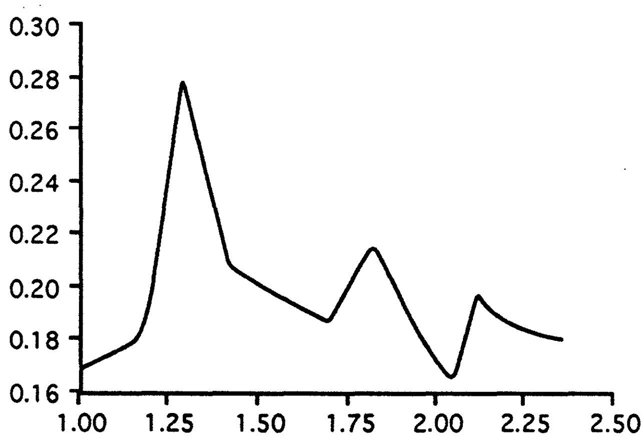

Figure 1. A graph of f(q).

EXTENSIONS

This methodology can be extended to find worst case examples for other parametric and non-parametric scheduling problems, especially those satisfying the following conditions:

1) The worst case examples are small, preferably with fewer than 7 tasks.

2) For each possible heuristic assignment x and for each parameter selection q, the set of processing times P giving the heuristic assignment is polyhedral.

3) The relative error for a fixed schedule is piecewise linear in P.
Some examples would include LPT heuristics for : $Q3||C_{max}$, $Q2|prec.|C_{max}$, $Q2||\Sigma T_j$, and many more.

ACKNOWLEDGMENTS

The authors wish to thank Richard Burns, Alexander Rinnooy Kan and Ronald Graham for their help and encouragement during early stages of the research. They also acknowledge partial support through NSF grant 84151517-ECS. The research of the third author was supported in part by a Dean's Summer Fellowship from the College of Business of the Ohio State University.

REFERENCES

Dobson, G., "Scheduling Independent Tasks on Uniform Processors", SIAM J. Computing, vol. 13, 705-716, 1984.

Friesen, D., "Tighter Bounds for LPT Scheduling on Uniform Processors", SIAM J. Computing, vol. 16, 554-560, 1987.

Graham, R., "Bounds on Multiprocessing Time Anomalies", SIAM J. Applied Math., vol. 17, 416-429, 1969.

Lawler, E., J. Lenstra, A. H. G. Rinnooy Kan, D. Shmoys, "Scheduling", in the Handbook of Operations Research, vol. 3, North Holland, Amsterdam, 1993.