stichting
mathematisch
centrum

MC

AFDELING MATHEMATISCHE BESLISKUNDE (DEPARTMENT OF OPERATIONS RESEARCH) BW 150/81 NOVEMBER

C.N. POTTS

ANALYSIS OF HEURISTICS FOR TWO-MACHINE FLOW-SHOP SEQUENCING SUBJECT TO RELEASE DATES

Preprint

kruislaan 413 1098 SJ amsterdam

Printed at the Mathematical Centre, 413 Kruislaan, Amsterdam.

The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O.).

1980 Mathematics Subject Classification: 90B35, 68C25

Analysis of heuristics for two-machine flow-shop sequencing subject to release dates *)

by

C.N. Potts**)

ABSTRACT

The two-machine flow-shop problem is considered in which each job becomes available for processing at its release date after which it must be processed without interruption on the first machine and then on the second machine. The objective is to minimize the maximum completion time. Three heuristics are presented which each have a worst-case performance ratio of 2. One of these is modified to give an improved worst-case performance ratio of 5/3.

KEY WORDS AND PHRASES: *two-machine flow-shop, release dates, maximum completion time, heuristics, worst-case performance*

---

*) This report will be submitted for publication elsewhere.

**) University of Keele, England

1. INTRODUCTION

The problem may be stated as follows. Each of n jobs (numbered 1,...,n) must be processed without interruption firstly on machine A and then on machine B. Job i (i = 1,...,n) becomes available for processing at its non-negative release date r_i and requires a positive processing time of a_i and b_i on machines A and B respectively. At any time the machine can handle only one job and each job can be processed on only one machine. The objective is to schedule the jobs so that the maximum completion time C_{max} is minimized. It is well-known [1,8] that it is unnecessary to consider schedules in which the processing orders on the two machines are not identical.

An equivalent problem exists [8] in which job i (i = 1,...,n) has a zero release date and has a non-positive due date d_i. After the jobs are sequenced, the completion time C_i and the lateness $L_i = C_i - d_i$ of job i (i = 1,...,n) can be computed. The objective is to sequence the jobs so that the maximum lateness is minimized. However, the original problem of minimizing C_{max} when jobs have arbitrary release dates will be considered henceforth.

When all release dates are equal, the problem can be solved in O(n log n) steps by the algorithm of JOHNSON [5] in which those jobs with $a_i \le b_i$ are sequenced first in non-decreasing order of a_i followed by the remaining jobs (with $a_i > b_i$) sequenced in non-increasing order of b_i. For arbitrary release dates, LENSTRA et al. [6] have shown that the problem is NP-hard which indicates that the existence of a polynomial bounded algorithm to solve the problem is unlikely. Apart from the branch and bound algorithm proposed by GRABOWSKI [4], the problem has received little attention from researchers.

In this paper, we propose some heuristic methods to sequence the jobs. Suppose that C^*_{max} denotes the minimum value of the maximum completion time while C^H_{max} denotes the maximum completion time when the jobs are sequenced using a certain heuristic H. If, whatever the problem data, $C^H_{max} \le \rho C^*_{max} + \delta$ for specified constants ρ and δ, where ρ is as small as possible, then ρ is called the *worst-case performance ratio* of H. A survey and discussion of the worst-case analysis of heuristics are given by FISHER [2] and GAREY et al. [3].

In Section 2 four heursitics are given, one of which is shown to have a worst-case performance ratio of 3 while the other three are each shown to have a worst-case performance ratio of 2. Section 3 shows how the repeated application of one of these heuristics to a constrained version of the original problem leads to an improved worst-case performance ratio of 5/3. This is followed by some concluding remarks in Section 4.

2. ANALYSIS OF HEURISTICS

The four heursitics to be analyzed in this section are described now. The first is heuristic ARB in which the jobs are sequenced arbitrarily after which C_{max}^{ARB} is evaluated in O(n) steps. The second is heuristic R in which the jobs are sequenced in non-decreasing order of release dates in O(n log n) steps and the third is heuristic J in which the jobs are sequenced according to Johnson's rule (ignoring release dates) in O(n log n) steps. If heuristic R is adopted, there will be no unforced idle time on machine A. The fourth heuristic RJ is a variant of R which attempts to take advantage of J while retaining the absence of unforced idle time on machine A: whenever there is a choice of jobs for the first unfilled position in the sequence which preserves this absence of unforced idle time, one is chosen which would be sequenced first amongst these jobs according to Johnson's rule. A formal statement of this heuristic, which requires O(n log n) steps, is given below.

<u>Heuristic RJ</u>

<u>Step 1</u>. Let S be the set of all jobs, let k = 0 and find $T = \min_{j \in S}\{r_j\}$.
<u>Step 2</u>. Find the set $S' = \{j \mid j \in S,\ r_j \le T,\ a_j \le b_j\}$ and the set $S'' = \{j \mid j \in S,\ r_j \le T,\ a_j > b_j\}$. If $S' \neq \emptyset$, find a job i in S' with a_i as small as possible; if $S' = \emptyset$, find a job i in S" with b_i as large as possible.
<u>Step 3</u>. Set k = k+1, sequence job i in position k, set $T = T + a_i$ and set $S = S - \{i\}$.
<u>Step 4</u>. If $S = \emptyset$, then stop. Otherwise set $T = \max\{T, \min_{j \in S}\{r_j\}\}$ and go to Step 2.

If any heuristic H generates a sequence $(\sigma(1),\ldots,\sigma(n))$, the corresponding maximum completion time can be written as

$$(1)\qquad C^H_{max} = r_{\sigma(u)} + \sum_{i=u}^{v} a_{\sigma(i)} + \sum_{i=v}^{n} b_{\sigma(i)},$$

for some $u,v \in \{1,\ldots,n\}$, where $u \le v$ and where u is chosen as small as possible.

Some lower bounding schemes for C^*_{max}, which are needed in the subsequent analysis, are introduced. In general, each job i has a set of *predecessors* which are jobs that are known to be sequenced before job i in an optimum sequence and a set of *successors* which are jobs that are known to be sequenced after job i in an optimum sequence. For any subset S of jobs, the *machine-based bound* for machine A (or machine B) is the sum of the following:

(i) the minimum release date of jobs in S which have no predecessors;

(ii) the total processing time on machine A (or machine B) of the jobs in S.

For any subset S of jobs, the *job-based bound* centered about any job j in S is the sum of the following:

(i) the minimum release date of jobs in S which have no predecessors;

(ii) the total processing time on machine A of all predecessors of job j;

(iii) the total processing time on machine A of all jobs i in $S - \{j\}$ with $a_i \le b_i$ which are neither predecessors nor successors of job j;

(iv) the total processing time of job j;

(v) the total processing time on machine B of all successsors of job j;

(vi) the total processing time on machine B of all jobs i in $S - \{j\}$ with $a_i > b_i$ which are neither predecessors nor sucessors of job j.

We now proceed with the derivation of the worst-case performance ratio for our four heuristics.

<u>THEOREM 1</u>. *$C^{ARB}_{max}/C^*_{max} < 3$, $C^R_{max}/C^*_{max} < 2$, $C^J_{max}/C^*_{max} < 2$ and $C^{RJ}_{max}/C^*_{max} < 2$ and these bounds are the best possible.*

<u>PROOF</u>. We assume in each case that the sequence generated is $(\sigma(1),\ldots,\sigma(n))$ and the maximum completion time is given by (1).

Clearly, C^*_{max} is greater than any release date, so

$$(2)\qquad C^*_{max} > r_{\sigma(u)}.$$

The machine-based bound for machine A and jobs in $\{\sigma(u),\ldots,\sigma(v)\}$ yields

$$(3) \qquad C^*_{max} > \sum_{i=u}^{v} a_{\sigma(i)}$$

and the machine-based bound for machine B and jobs in $\{\sigma(v),\ldots,\sigma(n)\}$ yields

$$(4) \qquad C^*_{max} > \sum_{i=v}^{n} b_{\sigma(i)}.$$

For heuristic ARB, by adding (2), (3) and (4) we obtain $3C^*_{max} > C^{ARB}_{max}$ as required.

Under heuristic R and RJ the minimum release date of jobs $\{\sigma(u),\ldots,\sigma(v)\}$ is $r_{\sigma(u)}$. Applying the machine-based bound for machine A to this set yields

$$(5) \qquad C^*_{max} > r_{\sigma(u)} + \sum_{i=u}^{v} a_{\sigma(i)}.$$

Adding (4) and (5) yields $2C^*_{max} > C^{R}_{max}$ and $2C^*_{max} > C^{RJ}_{max}$ as required.

Lastly, under heuristic J the jobs in $\{\sigma(u),\ldots,\sigma(n)\}$ are sequenced according to Johnson's rule. Their maximum completion time, ignoring release dates, provides the lower bound

$$(6) \qquad C^*_{max} > \sum_{i=u}^{v} a_{\sigma(i)} + \sum_{i=v}^{n} b_{\sigma(i)}.$$

Adding (2) and (6) we obtain $2C^*_{max} > C^{J}_{max}$ as required.

To complete the proof, we present an example to show that the bounds of Theorem 1 are the best possible.

Consider the 3-job problem specified by the date in Table 1, where $0 < 8k < K$.

Table 1. Data for the first example

i	1	2	3
r_i	k	0	$K-3k$
a_i	$2k$	$K-6k$	k
b_i	$K-6k$	k	$2k$

Clearly, (1,2,3) is an optimum sequence with $C^*_{max} = K$. If the jobs are sequenced arbitrarily in the order (3,2,1), we have $C^{ARB}_{max} = 3K-12k$. Therefore $C^{ARB}_{max}/C^*_{max} = 3 - 12k/K$ which can be arbitrarily close to 3. If either heuristic R or heuristic RJ is applies, the sequence (2,1,3) results with $C^{R}_{max} = C^{RJ}_{max} = 2K - 8k$. Therefore, $C^{R}_{max}/C^*_{max} = C^{RJ}_{max}/C^*_{max} = 2 - 8k/K$ which can be arbitrarily close to 2. Finally, heuristic J generates the sequence (3,1,2) with $C^{J}_{max} = 2K - 5k$. Thus $C^{J}_{max}/C^*_{max} = 2 - 5k/K$ which can also be arbitrarily close to 2. □

Henceforth, we shall examine heuristic RJ in more detail and suggest a method of improving it. We start by presenting two special identifiable cases in which the maximum deviation from the optimum is less than 50%. As before, it is assumed that C^{RJ}_{max} is given by (1).

THEOREM 2. *If* $a_{\sigma(i)} \le b_{\sigma(i)}$ *for* $i = u,\ldots,v$ *or if* $a_{\sigma(i)} \ge b_{\sigma(i)}$ *for* $i = v,\ldots,n$, *then* $C^{RJ}_{max}/C^*_{max} < 3/2$. *In each case, this bound is the best possible.*

PROOF. The machine-based bounds for jobs in $\{\sigma(u),\ldots,\sigma(n)\}$ on machines A and B are respectively

$$(7) \qquad C^*_{max} > r_{\sigma(u)} + \sum_{i=u}^{n} a_{\sigma(i)}$$

and

$$(8) \qquad C^*_{max} > r_{\sigma(u)} + \sum_{i=u}^{n} b_{\sigma(i)}.$$

Subtracting (8) from (1) we obtain

$$(9) \qquad C^{RJ}_{max} - C^*_{max} < \sum_{i=u}^{v} a_{\sigma(i)} - \sum_{i=u}^{v-1} b_{\sigma(i)}.$$

If $a_{\sigma(i)} \le b_{\sigma(i)}$ for $i = u,\ldots,v$, it follows from (9) that

$$C^{RJ}_{max} - C^*_{max} < a_{\sigma(v)} \le \tfrac{1}{2}(a_{\sigma(v)} + b_{\sigma(v)}) \le \tfrac{1}{2}C^*_{max}$$

which implies that $C^{RJ}_{max}/C^*_{max} < 3/2$ for this first case.

Subtracting (7) from (1) we obtain

$$C_{max}^{RJ} - C_{max}^{*} < \sum_{i=v}^{n} b_{\sigma(i)} - \sum_{i=v+1}^{n} a_{\sigma(i)}. \tag{10}$$

If $a_{\sigma(i)} \geq b_{\sigma(i)}$ for $i = v,\ldots,n$, then (10) implies that

$$C_{max}^{RJ} - C_{max}^{*} < b_{\sigma(v)} \leq \tfrac{1}{2}(a_{\sigma(v)} + b_{\sigma(v)}) \leq \tfrac{1}{2}C_{max}^{*}.$$

Therefore $C_{max}^{RJ}/C_{max}^{*} < 3/2$ for the second case also.

To complete the proof, we present examples to show that in each case the bound of 3/2 is the best possible.

Consider the 3-job problem specified by the data in Table 2, where $0 < 5k < K$.

Table 2. Data for the second example

i	1	2	3
r_i	k	0	0
a_i	$\frac{1}{2}k$	$\frac{1}{2}K-k$	$\frac{1}{2}K-k$
b_i	$\frac{1}{2}K-k$	$\frac{1}{2}K-k$	$\frac{1}{2}k$

Clearly (1,2,3) is an optimum sequence with $C_{max}^{*} = K$. When heuristic RJ is applied, the sequence $\sigma = (2,1,3)$ is generated with $C_{max}^{RJ} = r_{\sigma(1)} + a_{\sigma(1)} + b_{\sigma(1)} + b_{\sigma(2)} + b_{\sigma(3)} = 3K/2 - 5k/2$. Thus we have $u = v = 1$ with $a_{\sigma(1)} \leq b_{\sigma(1)}$. Furthermore, $C_{max}^{RJ}/C_{max}^{*} = 3/2 - 5k/(2K)$ which can be arbitrarily close to 3/2.

Consider now another 3-job problem with $r_1 = 0$ and $r_2 = 3k/2$ and where the other data are given in Table 2. We have that (1,2,3) is again an optimum sequence with $C_{max}^{*} = K$. When heuristic RJ is applied, the sequence $\sigma = (1,3,2)$ is generated with $C_{max}^{RJ} = r_{\sigma(1)} + a_{\sigma(1)} + a_{\sigma(2)} + a_{\sigma(3)} + b_{\sigma(3)} = 3K/2 - 5k/2$. Thus we have $v = n = 3$ with $a_{\sigma(3)} \geq b_{\sigma(3)}$. Furthermore, $C_{max}^{RJ}/C_{max}^{*} = 3/2 - 5k/(2K)$ which can be arbitrarily close to 3/2. □

When the conditions of Theorem 2 are satisfied heuristic RJ has a satisfactory worst-case performance. When the conditions are not satisfied, a method by which RJ can be imporved is proposed in the next section.

3. THE IMPROVED HEURISTIC

Before proceeding, some notation is introduced. Let

$$S_1 = \{\sigma(i) \mid i \in \{u,\ldots,v\},\ a_{\sigma(i)} \leq b_{\sigma(i)}\},$$
$$S_2 = \{\sigma(i) \mid i \in \{u,\ldots,v\},\ a_{\sigma(i)} > b_{\sigma(i)}\},$$
$$S_3 = \{\sigma(i) \mid i \in \{v,\ldots,n\},\ a_{\sigma(i)} \leq b_{\sigma(i)}\},$$
$$S_4 = \{\sigma(i) \mid i \in \{v,\ldots,n\},\ a_{\sigma(i)} > b_{\sigma(i)}\},$$

so we can write

$$(11) \qquad C_{max}^{RJ} = r_{\sigma(u)} + \sum_{i \in S_1 \cup S_2} a_i + \sum_{i \in S_3 \cup S_4} b_i.$$

We also define $S_i' = S_i - \{\sigma(v)\}$ $(i = 1,2,3,4)$.

The improved heuristic RJ' which, at each iteration, applies heuristic RJ and increases one release date is described now. The first step is to apply heuristic RJ and find the sets S_1, S_2, S_3 and S_4. If $S_2 = \emptyset$ or if $S_3 = \emptyset$, then computation is terminated. Otherwise we find a *changeover job* $\sigma(t)$ with $\sigma(t) \in S_2$ and with t chosen as large as possible and constrain it to be sequenced after at least one job in S_3 in each subsequent application of heuristic RJ. This constraint is implemented by setting $r_{\sigma(t)} = \min_{i \in S_3}\{r_i + p_i\}$. This process is repeated until S_2 or S_3 is empty at which stage $C_{max}^{RJ'}$ is chosen to be the maximum completion time of the best schedule generated. A formal statement of the heuristic is given below.

<u>Heuristic RJ'</u>

<u>Step 1</u>. Let $j = 1$ and let $C_{max}^{RJ'} = \infty$.

<u>Step 2</u>. Apply heuristic RJ to obtain a sequence σ_j with maximum completion time $C_{max}(\sigma_j)$. If $C_{max}(\sigma_j) < C_{max}^{RJ'}$, then set $C_{max}^{RJ'} = C_{max}(\sigma_j)$.

<u>Step 3</u>.Find S_2 and S_3. If $S_2 = \emptyset$ or if $S_3 = \emptyset$, then stop having found a sequence with maximum completion time $C_{max}^{RJ'}$. Otherwise find the changeover job $\sigma_j(t)$, set $r_{\sigma_j}(t) = \min_{i \in S_3}\{r_i + p_i\}$, set $j = j+1$ and go to Step 2.

Since there are in general $O(n)$ jobs i with $a_i \leq b_i$ and $O(n)$ jobs i with $a_i > b_i$, it may be necessary to impose $O(n^2)$ constraints before guaranteeing that S_2 or S_3 is empty. Each time a constraint is added heuristic RJ, which requires $O(n \log n)$ steps, is applied. Thus, heuristic RJ' requires $O(n^3 \log n)$ steps. However, it is expected that for most problems the heuristic will terminate in less than $O(n^2)$ iterations. Computation can be reduced by using the observation that those jobs sequenced in the first $t-1$ positions of σ_j before the changeover job $\sigma_j(t)$ are also sequenced, in the same order, in the first $t-1$ positions of σ_{j+1}.

We now prove that heuristic RJ' has a worst-case performance ratio of 5/3.

THEOREM 3. $C_{max}^{RJ'}/C_{max}^{*} < 5/3$ *and, for arbitrary* n, *this bound is the best possible.*

PROOF. Suppose that, after each increase in release date, the minimum value of the maximum completion time for that currect problem is equal to C_{max}^{*}. Then, at termination when S_2 or S_3 is empty, Theorem 2 is applied yielding $C_{max}^{RJ'}/C_{max}^{*} < 3/2$.

Alternatively, at some iteration, increasing a release date yields a current problem for which the minimum value of the maximum completion time exceeds C_{max}^{*}. Suppose that the first such increase in release date is derived from the sequence σ. Suppose also that the maximum completion time C_{max}^{RJ} for the sequence σ is given by (11) and that $\sigma(t)$ is the changeover job. Any sequence in which job $\sigma(t)$ is forced to be sequenced after at least one job in S_3 has a maximum completion which exceeds C_{max}^{*}. Therefore, in any optimum sequence, $\sigma(t)$ is sequenced before all jobs in S_3. We prove that $C_{max}^{RJ}/C_{max}^{*} < 5/3$ by using this requirement.

Some lower bounds on C_{max}^{*} which are used throughout the proof are given first. The machine-based bound for the jobs in $\{\sigma(u),\ldots,\sigma(n)\}$ on machine A is

$$C_{max}^{*} > r_{\sigma(u)} + \sum_{i \in S_1 \cup S_2} a_i + \sum_{i \in S_s' \cup S_4'} a_i,$$

which implies that

$$(12)\qquad C^*_{max} > r_{\sigma(u)} + \sum_{i\in S_1 \cup S_2} a_i + \sum_{i\in S'_4} b_i.$$

To derive our next lower bound, we observe that the time at which the processing of job $\sigma(t)$ commences is less than the release date of all jobs in S_3: if it were not, heuristic RJ would sequence a job in S_3 in preference to $\sigma(t)$. Assume that in an optimum sequence some job $\sigma(w)$ is sequenced first amongst jobs in S_3. Then the job-based bound centred about job $\sigma(w)$ for the jobs in S_3 is

$$(13)\qquad C^*_{max} > r_{\sigma(u)} + \sum_{i\in S_2} a_i - a_{\sigma(t)} + a_{\sigma(w)} + \sum_{i\in S_3} b_i.$$

The case that job $\sigma(v)$ is the changeover job and the case that it is not are considered separately.

<u>Case 1</u>. $a_{\sigma(v)} > b_{\sigma(v)}$ (implying $\sigma(v) \in S_2$ and $\sigma(v) \in S_4$).

In this case, job $\sigma(v)$ is the changeover job, i.e. $t = v$. The job-based bound centered about job $\sigma(v)$ for the jobs in $\{\sigma(u),...,\sigma(n)\}$ is

$$(14)\qquad C^*_{max} \geq r_{\sigma(u)} + \sum_{i\in S_1} a_i + a_{\sigma(v)} + b_{\sigma(v)} + \sum_{i\in S'_2 \cup S_3 \cup S'_4} b_i,$$

since the job of S_3 are known to be successors of job $\sigma(v)$ in an optimum sequence. If $S'_2 = \emptyset$, then (14) implies that $C^{RJ}_{max} = C^*_{max}$. If $S'_2 \neq \emptyset$, we compute (1/2)((12 + (13) + (14)) to obtain

$$(15)\qquad \begin{aligned} \frac{3}{2} C^*_{max} &> \frac{3}{2} r_{\sigma(u)} + \sum_{i\in S_1 \cup S_2} a_i + \sum_{i\in S_3 \cup S_4} b_i \\ &+ \frac{1}{2} (a_{\sigma(w)} + \sum_{i\in S'_2} b_i - b_{\sigma(v)}). \end{aligned}$$

Now, if S'_2 contains a job $\sigma(s)$ with $b_{\sigma(s)} \geq b_{\sigma(v)}$, then (15) implies that $C^{RJ}_{max}/C^*_{max} < 3/2$. Alternatively, if S'_2 contains no such job, then we may assume that in the sequence σ job $\sigma(s)$, with $b_{\sigma(s)} < b_{\sigma(v)}$, is sequenced last amongst jobs in S'_2. The time at which the processing of job $\sigma(s)$ commences is less than the release date of job $\sigma(v)$ due to the construction of σ by heuristic RJ. Thus, the job-based bound centred about $\sigma(v)$ for the jobs in $S_3 \cup \{\sigma(v)\}$ is

(16) $$C^*_{max} > r_{\sigma(u)} + \sum_{i \in S_2'} a_i - a_{\sigma(s)} + a_{\sigma(v)} + b_{\sigma(v)} + \sum_{i \in S_3} b_i,$$

since the jobs of S_3 are known to be successors of job $\sigma(v)$ in an optimum sequence.

Firstly, suppose that in an optimum sequence job $\sigma(s)$ is sequenced before job $\sigma(v)$. Then the job-based bound centred about job $\sigma(v)$ for the jobs in $S_1 \cup S_3 \cup S_4 \cup \{\sigma(s)\}$ is

(17) $$C^*_{max} > r_{\sigma(u)} + \sum_{i \in S_1} a_i + a_{\sigma(s)} + a_{\sigma(v)} + b_{\sigma(v)} + \sum_{i \in S_3 \cup S_4'} b_i.$$

Computing $(\frac{1}{2})((12) + (16) + (17))$ yields

$$\frac{3}{2} C^*_{max} > \frac{3}{2} r_{\sigma(u)} + \sum_{i \in S_1 \cup S_2} a_i + \sum_{i \in S_3 \cup S_4} b_i + \frac{1}{2} a_{\sigma(v)},$$

which implies that $C^{RJ}_{max}/C^*_{max} < 3/2$.

Secondly, suppose that in an optimum sequence job $\sigma(s)$ is sequenced after job $\sigma(v)$ but before any job in S_3. Recalling that $r_{\sigma(v)} > r_{\sigma(u)} + \sum_{i \in S_2'} a_i - a_{\sigma(s)}$, the job-based bound centred about job $\sigma(s)$ for the jobs in $S_3 \cup \{\sigma(s), \sigma(v)\}$ is

(18) $$C^*_{max} > r_{\sigma(u)} + \sum_{i \in S_2'} a_i + a_{\sigma(v)} + b_{\sigma(s)} + \sum_{i \in S_3} b_i.$$

Computing $(1/2)((12) + (14) + (18))$ yields

$$\frac{3}{2} C^*_{max} > \frac{3}{2} r_{\sigma(u)} + \sum_{i \in S_1 \cup S_2} a_i + \sum_{i \in S_3 \cup S_4} b_i + \frac{1}{2}\Big(\sum_{i \in S_2'} b_i + b_{\sigma(s)} + a_{\sigma(v)} - b_{\sigma(v)}\Big),$$

which implies that $C^{RJ}_{max}/C^*_{max} < 3/2$ because $a_{\sigma(v)} > b_{\sigma(v)}$.

Thirdly and lastly, suppose that in an optimum sequence job $\sigma(s)$ is sequenced after at least one job of S_3. The machine-based bound for the jobs in $S_3 \cup \{\sigma(s)\}$ on machine A is

(19) $$C^*_{max} > r_{\sigma(u)} + \sum_{i\in S_2'} a_i + \sum_{i\in S_3} a_i + a_{\sigma(s)}.$$

Computing (1/5)(2 × (12) + (13) + 3 × (14) + (16) + (19)) yields

$$\frac{8}{5} C_{max} > \frac{8}{5} r_{\sigma(u)} + \sum_{i\in S_1 \cup S_2} a_i + \sum_{i\in S_3 \cup S_4} b_i +$$
$$\frac{1}{5} (a_{\sigma(v)} - b_{\sigma(v)} + 3 \sum_{i\in S_2'} b_i + \sum_{i\in S_3} a_i + a_{\sigma(w)}),$$

which implies that $C^{JR}_{max}/C^*_{max} < 8/5$ because $a_{\sigma(v)} > b_{\sigma(v)}$.
This completes the proof of Case 1.

Case 2. $a_{\sigma(v)} \le b_{\sigma(v)}$ (implying $\sigma(v) \in S_1$ and $\sigma(v) \in S_3$)

In this case, for the sequence σ, the changeover job $\sigma(t)$ is sequenced before job $\sigma(v)$. Recall that $\sigma(w)$ is sequenced first amongst jobs in S_3 in an optimum sequence.

Firstly suppose that $a_{\sigma(w)} \ge a_{\sigma(v)}$. The job-based bound centred about job $\sigma(w)$ for the jobs in $S_1 \cup S_3 \cup S_4 \cup \{\sigma(t)\}$ is

(20) $$C^*_{max} > r_{\sigma(u)} + \sum_{i\in S_1'} a_i + a_{\sigma(t)} + a_{\sigma(w)} + \sum_{i\in S_3 \cup S_4} b_i,$$

since job $\sigma(t)$ is sequenced before job $\sigma(w)$. Computing (1/2)((12) + (13) + + (20)) yields

$$\frac{3}{2} C^*_{max} > \frac{3}{2} r_{\sigma(u)} + \sum_{i\in S_1 \cup S_2} a_i + \sum_{i\in S_3 \cup S_4} b_i + a_{\sigma(w)} - \frac{1}{2} a_{\sigma(v)}$$

which implies that $C^*_{max}/C^*_{max} < 3/2$ because $a_{\sigma(w)} \ge a_{\sigma(v)}$.

Secondly and lastly, suppose that $a_{\sigma(w)} < a_{\sigma(v)}$. The job-based bound centred about job $\sigma(v)$ for jobs in $S_1 \cup S_4$ is

(21) $$C^*_{max} > r_{\sigma(u)} + \sum_{i\in S_1} a_i + b_{\sigma(v)} + \sum_{i\in S_4} b_i.$$

The machine-based bound for jobs in $S_3 \cup S_4$ on machine B is

(22) $$C^*_{max} > r_{\sigma(u)} + \sum_{i\in S_3 \cup S_4} b_i.$$

The time at which the processing of job $\sigma(v)$ commences is less than $r_{\sigma(w)}$: if it were not, then heuristic RJ would sequence job $\sigma(w)$ in preferance to job $\sigma(v)$. Thus, the machine-based bound for jobs in S_3 on machine B is

$$C^*_{max} > r_{\sigma(u)} + \sum_{i\in S_1' \cup S_2} a_i + \sum_{i\in S_3} b_i. \tag{23}$$

Computing (1/3)((12) + (21) + (22) + 2 × (23)) yields

$$\frac{5}{3} C^*_{max} > \frac{5}{3} r_{\sigma(u)} + \sum_{i\in S_1 \cup S_2} a_i + \sum_{i\in S_3 \cup S_4} b_i$$
$$+ \frac{1}{3} \Big(\sum_{i\in S_1'} a_i + b_{\sigma(v)} - a_{\sigma(v)} \Big)$$

which implies that $C^{RJ}_{max}/C^*_{max} < 5/3$ since $b_{\sigma(v)} \geq a_{\sigma(v)}$.

To complete the proof, we present an example to show that, for arbitrary n, the bound of 5/3 is the best possible. Consider the n-job problem ($n \geq 5$) specified by the data in Table 3, where $0 < n^2 \varepsilon < k$ and where $k = K/(3n-8)$.

Table 3. Data for the third example

i	1		n-3	n-2	n-1	n
r_i	ε	. . .	ε	$\frac{1}{3}(K+2k+3\varepsilon)$	$\frac{1}{3}K$	0
a_i	k	. . .	k	ε	$\frac{1}{3}(K-k-6\varepsilon)$	$\frac{1}{3}(K-k-3\varepsilon)$
b_i	$k-\varepsilon$	. . .	$k-\varepsilon$	$\frac{1}{3}(K-k-6\varepsilon)$	$\frac{1}{3}(K-k-3\varepsilon)$	ε

Clearly, (1,...,n) is an optimum sequence with $C^*_{max} = K$. The first n-3 applications of heuristic RJ produce sequences (n,i,n-1,n-2,1,...,i-1, i+1,...,n-3) for i=1,...,n-3, each with $C^{RJ}_{max} = 5(K-k)/3 - (n+2)\varepsilon$. Job i is the changeover job and we set $r_i = r_{n-2} + p_{n-2} = 1/(3(K+2k+6\varepsilon))$. Application n-2 of heuristic RJ produces the sequence (n,n-1,n-2,1,...,n-3) with $C^{RJ}_{max} = (5K-4k)/3 - (n+2)\varepsilon$. At this stage there is no changeover job and the algorithm terminates with $C^{RJ'}_{max}/C^*_{max} = 5/3 - 5k/(3K) - (n+2)\varepsilon/K$ which can be arbitrarily close to $5/3 - 5k/(3K)$ or $5/3 - 5/(3(3n-8))$. This in turn can be arbitrarily close to 5/3 when n is arbitrary. □

It is, perhaps, rather surprising that for arbitrary n the bound of 5/3 is the best possible, since it might be expected that one of the other sequences generated by heuristic RJ' would give a lower value of the maximum completion time than the value C^{RJ}_{max} which is used in the proof of Theorem 3. However, the example demonstrates that this is not the case.

If n is fixed, there is a difference between the upper bound of 5/3 for $C^{RJ'}_{max}/C^{*}_{max}$ and its lower bound of $5/3 - 5/(3(3n-8))$. Further research is required to resolve this difference.

4. CONCLUDING REMARKS

We have constructed a heuristic method of sequencing the job producing a maximum completion time which lies within two thirds of the value of the optimum. As is usual for most heuristics, the average performance is likely to be considerably better than the worst-case performance.

The method of repeatedly applying a simple heuristic to an increasingly constrained version of the original problem was also used in [7] for a single machine sequencing problem with release dates. It seems likely that simple heuristics for other scheduling problems can be improved using this technique.

ACKNOWLEDGEMENT

The author is grateful to the Mathematisch Centrum, Amsterdam for helping to finance a visit to the Mathematisch Centrum where this research was undertaken. The author is also grateful to J.K. Lenstra for his comments on a preliminary version of this paper.

REFERENCES

[1] CONWAY, R.W., W.L. MAXWELL & L.W. MILLER, *Theory of Scheduling*, Addison-Wesley, Reading, MA (1967).

[2] FISHER, M.L. *Worst-case analysis of heuristic algorithms*, Management Sci. 26, 1-17 (1980).

[3] GAREY, M.R., R.L. GRAHAM & D.S. JOHNSON, *Performance guarantees for scheduling algorithms*, Operations Res. 26, 3-21 (1978).

[4] GRABOWSKI, J., *On two-machine scheduling with release dates to minimize maximum lateness*, Opsearch 17, 133-154 (1980).

[5] JOHNSON, S.M., *Optimal two- and three-stage production schedules with setup times included*, Naval Res. Logist. Quart. 1, 61-68 (1954).

[6] LENSTRA, J.K., A.H.G. RINNOOY KAN & P. BRUCKER, *Complexity of machine scheduling problems*, Ann. Discrete Math. 1, 343-362 (1977).

[7] POTTS, C.N., *Analysis of a heuristic for one machine sequencing with release dates and delivery times*, Operations Res. 28, 1436-1441 (1980).

[8] RINNOOY KAN, A.H.G., *Machine Scheduling Problems: Classification, Complexity and Computations*, Nijhoff, The Hague (1976).