Pergamon

PII: S0305-0548(96)00051-2

*Computers Ops Res.* Vol. 24, No. 2, pp. 169–178, 1997

0305-0548/97 $17.00+0.00

# SINGLE FACILITY SCHEDULING WITH MAJOR AND MINOR SETUPS

Ching-Jong Liao†‡ and Li-Man Liao§

Department of Industrial Management, National Taiwan Institute of Technology, 43 Keelung Road, Section 4, Taipei, Taiwan 106

*(Received August 1995; in revised form May 1996)*

**Scope and Purpose**—In many practical situations, it is frequently encountered that jobs can be grouped, based on similarities, into different classes and these classes can be further grouped into different families. A major setup and a minor setup are required when jobs are switched from one family to another while only a minor setup is required when jobs are switched from one class to another within the same family. These setup times depend only on the family or the class being switched to. In this article, a heuristic based on dynamic programming is developed to minimize the mean flow time of jobs.

**Abstract**—This article considers the problem of scheduling a given set of jobs at a single facility, where jobs can be grouped into different classes and these classes can be further grouped into different families. A major setup and a minor setup are required when jobs are switched from one family to another while only a minor setup is required when jobs are switched from one class to another within the same family. The minimum mean flow time schedule of this problem can be solved optimally by dynamic programming (DP), but the exponential behaviour of the DP solution precludes its use to solve problems with a large number of classes. An efficient heuristic is thus developed in which a sequence of two-class problems is solved. Computational results show that the heuristic produces solutions that deviate 0.118% on average from the optimum. Copyright © 1997 Elsevier Science Ltd

## INTRODUCTION

This article considers the problem of scheduling a given set of jobs at a single facility. Based on similarities, the jobs can be grouped into different classes and these classes can be further grouped into different families. For example, in the manufacture of aluminum press products [1], each product is pressed through a die which is located in a die carrier. Based on the size and shape of the dies, they can be grouped into several different families where dies in the same family share the same die carrier. In such situations, setup times may be incurred when a job (say A) changes to another job (say B). If jobs A and B are in the same family but in different classes, a minor setup time associated with the new class will be incurred. If jobs A and B are in different families, not only a minor setup time but a major setup time associated with the new family will be incurred. In the aluminum press example, the minor setup time may refer to the time for the change of die while the major setup time refers to the time for the change of die carrier. It was reported in a case study that 1 h was required for a change of die and 4 h for a change of die carrier [1].

In what follows, we briefly review the related research. The problem of minimizing mean flow time with job classes was probably first considered by Gupta [2]. He proposed a polynomically bounded algorithm for solving the problem with two job classes, but it has been recently shown by Potts [3] that his algorithm fails to generate the optimal schedule in some cases. For more than two classes, Gupta [4] developed a simple one-pass heuristic to minimize the mean flow time. Later, Ahn and Hyun [5] established the so-called intragroup shortest processing time property and modified Psaraftis's [6] DP algorithm to solve the same problem. As a result of the large storage requirement of the DP algorithm, they also proposed an efficient heuristic for solving large-sized problems.

For problems with more than one stage of setup, some relevant research can be found in the following publications. Using the criterion of minimum makespan, Tang [7] developed a heuristic that determines the quantity and the schedule of jobs on parallel machines. Leong and Oliff [8] developed a heuristic to

† To whom all correspondence should be addressed.

‡ Ching-Jong Liao is a Professor of Industrial Management at the National Taiwan Institute of Technology. He completed his PhD in Industrial Engineering in 1988 from the Pennsylvania State University. His current research interests include production scheduling and inventory control. His papers have appeared in Oprn. Res. Lett., IIE Transactions, J. Opl Res. Soc., Computers Ops Res., etc.

§ Li-Man Liao is a PhD student in Industrial Management at the National Taiwan Institute of Technology. Her research interest is in production scheduling.

reduce setups in the manufacture of fibreglass, where changes in fibre type are the most expensive setups and changes in weight are the next. They also considered other minor setups such as width changes and trim changes. Ozden *et al.* [9] analyzed a group technology environment where jobs switching from group to group require a major setup while switching within a group takes only a small setup time. They develop a DP formulation to minimize the makespan.

The problem considered in this article is a two-stage setup time problem with the criterion of minimum mean flow time, which has not been studied by other authors. Although this problem can be solved by the DP method in a way similar to Psaraftis [6] and Ahn and Hyun [5], but the exponential behavior of the DP solution precludes its use to solve problems with a large number of classes [10]. In this article, an efficient heuristic is proposed in which a sequence of two-class problems is solved.

## PROBLEM SETTING

Consider the problem of scheduling $n$ jobs at a single facility where jobs are grouped into $K$ mutually exclusive classes. Let $n_k$ ($k=1,\ldots, K$) be the number of jobs in class $k$ and $W_k$ be its setup time. Then $n=\sum_{k=1}^{K} n_k$. For a given sequence of jobs in class $k$, let $J_{kj}$ be the $j$th job in the sequence and $p_{kj}$ be its processing time.

Let $H(k)$ be the family index of class $k$ and $S_{H(k)}$ be the setup time for the family. Given a job sequence, let $J_{[i]}$ be the job placed in the $i$th position, $G_{[i]}$ be the class index of $J_{[i]}$, and $p_{[i]}$ be its processing time. Associated with each job $J_{[i]}$ there is a completion time $C_{[i]}$, which is computed as follows:

$$C_{[i]}=C_{[i-1]}+S+W+p_{[i]}$$

where

$$W=\begin{cases} W_{G_{[i]}} & \text{if } G_{[i]}\neq H(G_{[i-1]}) \\ 0 & \text{otherwise} \end{cases}$$

and

$$S=\begin{cases} S_{H(G_{[i]})} & \text{if } H(G_{[i]})\neq H(G_{[i-1]}) \\ 0 & \text{otherwise} \end{cases}$$

Assuming that all jobs are ready to be processed at time zero, the flow time of $J_{[i]}$, denoted by $F_{[i]}$, is equal to its completion time. Letting $\bar{F}$ be the mean flow time, the objective of the problem can then be expressed as

$$\text{Minimize } \bar{F}=\frac{1}{n}\sum_{i=1}^{n} F_{[i]}$$

Since the number of jobs $n$ is a constant, minimizing the total flow time is equivalent to minimizing the mean flow time. For convenience, the total flow time will also be referred to as the criterion to be minimized in subsequent discussions.

In the following, we present three useful properties. Property 1 will be used both in the DP formulation and the heuristic. Properties 2 and 3 will be used only in the heuristic. These properties are stated without proof because Property 1 is a direct consequence from Ahn and Hyun [5] and Properties 2 and 3 are extensions of the classic result of the required strings of jobs [11]. The extensions are straightforward in that the setup time here is included in the total processing time of the connected string or class.

**Property 1.** Suppose that $P_{ki} < P_{kj}$. To minimize the mean flow time, $J_{ki}$ must precede $J_{kj}$ in the optimal sequence.

**Property 2.** Suppose that jobs in the same class have to be processed together. To minimize the mean flow time, the classes in the same family have to be sequenced in the non-decreasing order of the following ratio:

$$\frac{W_k+\sum_{j=1}^{n_k} p_{kj}}{n_k}$$

**Property 3.** Suppose that jobs in the same class have to be processed together and that classes in the same family have to be processed together. To minimize the mean flow time, the families have to be sequenced in the non-decreasing order of the following ratio:

$$\frac{S_h+\Sigma_{H(k)=h}[W_k+\Sigma_{j=1}^{n_k}p_{kj}]}{\Sigma_{H(k)=h}n_k}$$

## DYNAMIC PROGRAMMING FORMULATION

The DP formulation presented in this section serves two purposes. One is to obtain the optimal solution and the other is to be used as the base for the proposed heuristic.

According to Property 1, we pre-sort jobs in each class in the non- decreasing order of their processing times [5], and hence $p_{ki} \leq p_{kj}$ if $i<j$. Let $n(k)$ be the number of jobs scheduled from class $k$. Then we can define the optimal value function as $f(v,n(1),...,n(K))$=the minimum total flow time obtainable from the first $i$ ($i=\Sigma_{k=1}^{K}n(k)$) jobs in the sequence given that the last job is in class $v$.The recursive relation is

$$f(v,n(1),...,n(v),...,n(K))=\min\begin{cases} f(v,n(1),...,n(v)-1,...,n(K)) \\ \quad +(n-i+1)P_{v,n(v)} \\ f(u,n(1),...,n(v)-1,...,n(K)) \\ \quad +(n-i+1)(S+W_v+P_{v,n(v)}) \text{ if } u\neq v \end{cases}$$

where $i=\Sigma_{k=1}^{K}n(k)$ and

$$S=\begin{cases} S_{H(v)} & \text{if } H(u)\neq H(v) \\ 0 & \text{otherwise} \end{cases}$$

The boundary condition is

$$f(0,...,0)=0$$

The answer is

$$\min f(v,n_1,...,n_K) \text{ for } v=1,2,...,K$$

It is clear that this formulation is a forward dynamic program. We now show that the recursive relation is correctly given as above. We note that if we do not take setup times into consideration, the contribution of $J_{v,n(v)}$ to the total flow time is $(n-i+1)p_{v,n(v)}$. If the class of the $i$th job, $v$, is the same as the class of the $(i-1)$th job, no setup time will be incurred. Otherwise, there is a minor setup time $W_v$. In addition, a major setup time $S_{H(v)}$ will be incurred if these two classes belong to different families. The boundary condition and the answer are self-explanatory, which completes the verification of the formulation.

Suppose the number of jobs in each class, $\bar{n}$, is the same for all classes. Then the time complexity of the DP solution is of the order of $O(K^2[1+\bar{n}]^K)$ since there are $K\times(1+\bar{n})^K$ possible states, each of which is determined from $K$ previous states. Thus, the DP solution is exponential with respect to the number of classes. Although the marginal running time of such a formulation is a linear function of $n$ [6], the storage requirements remain to be dealt with, especially for problems with a large number of classes. For example, in order to solve problems with 6 classes each consisting of 7 jobs, it requires a main storage memory of about 6M bytes ($6\times(1+7)^6\times4$) for array declarations of the optimal value function. (Note that four bytes are required for each long integer.) Suppose 5M bytes of memory space is available for solving our scheduling problem. Then the largest number of jobs in each class can be calculated as 22,11,6,4,3,2,2 for problems with 4,5,...,10 classes, respectively. For problems under these break-points, they can be satisfactorily solved by the DP approach. Otherwise, we have to resort to the heuristic approach, which will be developed in the next section.

## HEURISTIC APPROACH

As stated earlier, the above DP solution is exponential with respect to the number of classes. If the number of classes is too large to be solved optimally by DP, a heuristic approach can be adopted in which a sequence of two-class problems is solved. The heuristic consists of two parts, each of which produces

a sequence. The solution of the heuristic is obtained by selecting the one with the smaller total flow time.

*Part 1*

*Step* 1. Obtain an initial feasible sequence.

(a) For each class $k$, sequence jobs in the non-decreasing order of $p_{kj}$.
(b) For each family, sequence classes in the non-decreasing order of

$$\frac{W_k+\sum_{j=1}^{n_k} p_{kj}}{n_k}$$

(c) Sequence families in the non-decreasing order of

$$\frac{S_h+\sum_{H(k)=h}[W_k+\sum_{j=1}^{n_k} p_{kj}]}{\sum_{H(k)=h} n_k}$$

*Step* 2. Sequence jobs in each family. For each family, perform the following steps.

(a) Determine the sequence for jobs in the first two classes by using the DP formulation developed earlier, where n in the recursive relation represents the number of jobs in these two classes only. Treat the resulting sequence as a single class.
(b) Determine the sequence for jobs in the combined sequence in (a) and jobs in an added new class by using the two-class DP formulation.
(c) Perform (b) iteratively until all classes in the family are sequenced.

*Step* 3. Sequence jobs in all families. In this step, families are treated as classes in the DP formulation.

(a) Determine the sequence for jobs in the first two families by using the DP formulation. Treat the resulting sequence as a single family.
(b) Determine the sequence for jobs in the combined sequence in (a) and jobs in an added new family by using the two-class DP formulation.
(c) Perform (b) iteratively until all families are sequenced.

*Part 2*

Part 2 is the same as Part 1 except that the $n$ in the DP recursive relation denotes the total number of jobs in all the classes.

We now discuss the difference between Parts 1 and 2, which differ only in the definition of $n$ in the DP recursive relation. Recall that $(n-i+1)$ in the DP formulation is the weight for the contribution of $J_{v,n(v)}$ to the total flow time. If, as in Part 1, $n$ is defined as the number of jobs in the two concerned classes, then $(n-i+1)$ represents the number of unscheduled jobs in the two concerned classes only. If, as in Part 2, $n$ is defined as the number of jobs in all the classes, then $(n-i+1)$ represents the total number of unscheduled jobs.

It can be observed from the recursive relation that if the top line on the right is smaller, then the $i$th job is in the same class as its immediately previous job. However, if the second line is smaller, then the ith job is in the other class. When the weight is smaller (as in Part 1), the second line has a larger chance to yield the minimum, and hence the resulting sequence tends to have jobs from the two classes mixed together. However, when the weight is larger (as in Part 2), the top line has a larger chance to yield the minimum, and hence the resulting sequence tends to have jobs from the same class grouped together.

Suppose that the number of jobs in each class, $\bar{n}$, is the same for all classes and the number of classes in each family, $\bar{k}$, is the same for all families. Then the time complexity of the heuristic is of the order of $O(2^2[1+(K-\bar{k})\bar{n}][1+\bar{k}\bar{n}])$, or simply $O(\bar{k}\bar{n}^2(K-\bar{k}))$, since only two classes are considered at each iteration. Thus, the heuristic is polynomial with respect to the number of classes.

## NUMERICAL EXAMPLE

A numerical example with two families and four classes is given below to illustrate the proposed heuristic. (The jobs in each class have been pre-sorted in the non-decreasing order of their processing

times.)

| Family | Class | Processing time | Class setup time | Family setup time |
|---|---|---|---|---|
| 1 | 1 | $p_{11}=12, p_{12}=17, p_{13}=21$ | 9 | 24 |
| | 2 | $p_{21}=11, p_{22}=28$ | 5 | |
| 2 | 3 | $p_{31}=10, p_{32}=19, p_{33}=28$ | 5 | 27 |
| | 4 | $p_{41}=14$ | 6 | |

*Part 1.*

*Step* 1.

(a) The jobs in each class have already been sequenced in the non-decreasing order of their processing times.
(b) The ratio for class 1 is calculated as follows:

$$\frac{9+(12+17+21)}{3}=19\frac{2}{3}$$

The ratio for class 2 is calculated to be 22, so the class sequence for family 1 is (class 1, class 2). Similarly, the class sequence for family 2 is (class 4, class 3).
(c) The ratio for family 1 is calculated as follows:

$$\frac{24+[(9+12+17+21)+(5+11+28)]}{(3+2)}=25\frac{2}{5}$$

Similarly, the ratio for family 2 is found to be 27.25, so the family sequence for the problem is (family 1, family 2).

*Step* 2. Applying DP, we obtain job sequences ($J_{21}$,$J_{11}$,$J_{12}$,$J_{13}$,$J_{22}$) and ($J_{31}$,$J_{32}$,$J_{41}$,$J_{33}$) for families 1 and 2, respectively. As an illustration of the DP procedure, the calculations for family 1 are given in Table 1. The $q$ in the table is the class of the next-to-the-last job of the sequence resulting in $f$. For example, $f(1, 2, 1)$ is calculated as follows:

Table 1. Calculations of step 2 in part 1 for family 1

| $n(1)+n(2)$ | $(n(1),n(2))$ | $v$ | $f$ | $q$ |
|---|---|---|---|---|
| 1 | (1,0) | 1 | 105 | - |
| | (0,1) | 2 | 80 | - |
| 2 | (2,0) | 1 | 173 | 1 |
| | (1,1) | 1 | 164 | 2 |
| | | 2 | 169 | 1 |
| | (0,2) | 2 | 192 | 2 |
| 3 | (3,0) | 1 | 236 | 1 |
| | (2,1) | 1 | 215 | 1 |
| | | 2 | 221 | 1 |
| | (1,2) | 1 | 255 | 2 |
| | | 2 | 253 | 2 |
| 4 | (3,1) | 1 | 257 | 1 |
| | | 2 | 268 | 1 |
| | (2,2) | 1 | 289 | 1 |
| | | 2 | 277 | 2 |
| 5 | (3,2) | 1 | 307 | 2 |
| | | 2 | 290 | 1 |

$$f(1,2,1)=\min\begin{cases} f(1,1,1)+3\times 17 \\ f(2,1,1)+3\times(9+17) \end{cases}$$

$$=\min\begin{cases} 164+51 \\ 169+78 \end{cases}$$

$$=215$$

So, $q=1$.

*Step* 3. Applying DP and treating families as classes, we obtain job sequences $(J_{21},J_{11},J_{12},J_{13},J_{22},J_{31},J_{32},J_{41},J_{33})$ with a total flow time of 1236 for the problem.

*Part 2.*

Step 1 is the same as in Part 1. In Step 2, we use the total number of unscheduled jobs as the weight and obtain sequences $(J_{11},J_{12},J_{13},J_{21},J_{22})$ and $(J_{31},J_{32},J_{41},J_{33})$ for the two families. In Step 3, we obtain job sequences $(J_{11},J_{12},J_{13},J_{21},J_{22},J_{31},J_{32},J_{41},J_{33})$ with a total flow time of 1222 for the problem.

Table 2. Computional results of the heuristic for two families

| No. of classes | No. of Jobs | % Errors | | | | | | | % Optimum of Part 1+2 |
|---|---|---|---|---|---|---|---|---|---|
| | | Step 1 | Part 1 | Part 2 | Part 1+2 | | | | |
| | | Mean | Mean | Mean | Mean | Min. | Max. | Std. | |
| 4 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 |
| | 18 | 0.205 | 0.156 | 0 | 0 | 0 | 0 | 0 | 100 |
| | 24 | 0.188 | 0.316 | 0.043 | 0.043 | 0 | 0.395 | 0.121 | 80 |
| | 30 | 0.484 | 0.106 | 0.106 | 0.055 | 0 | 0.496 | 0.131 | 80 |
| | 36 | 0.882 | 0.146 | 0.279 | 0.138 | 0 | 0.496 | 0.141 | 20 |
| | 42 | 1.397 | 0.257 | 0.390 | 0.242 | 0 | 0.961 | 0.264 | 25 |
| | 48 | 1.673 | 0.203 | 0.432 | 0.157 | 0 | 0.482 | 0.157 | 20 |
| | 54 | 2.187 | 0.173 | 0.419 | 0.173 | 0 | 0.407 | 0.139 | 15 |
| | 60 | 2.405 | 0.212 | 0.511 | 0.187 | 0 | 0.545 | 0.140 | 5 |
| | 72 | 2.737 | 0.265 | 0.396 | 0.234 | 0 | 0.459 | 0.113 | 5 |
| | 84 | 3.368 | 0.241 | 0.456 | 0.228 | 0 | 0.541 | 0.179 | 10 |
| | 96 | 3.849 | 0.248 | 0.334 | 0.198 | 0 | 0.423 | 0.107 | 5 |
| | 108 | 4.063 | 0.239 | 0.293 | 0.181 | 0.014 | 0.574 | 0.145 | 0 |
| | 120 | 3.970 | 0.206 | 0.292 | 0.161 | 0.032 | 0.437 | 0.113 | 0 |
| | 132 | 4.547 | 0.180 | 0.202 | 0.160 | 0 | 0.356 | 0.115 | 10 |
| | 144 | 5.111 | 0.215 | 0.209 | 0.172 | 0.018 | 0.389 | 0.109 | 0 |
| 5 | 12 | 0.027 | 0 | 0 | 0 | 0 | 0 | 0 | 100 |
| | 18 | 0.150 | 0.100 | 0.053 | 0.007 | 0 | 0.066 | 0.020 | 90 |
| | 24 | 0.201 | 0.152 | 0.079 | 0.014 | 0 | 0.244 | 0.055 | 90 |
| | 30 | 0.360 | 0.120 | 0.182 | 0.046 | 0 | 0.382 | 0.103 | 75 |
| | 36 | 0.815 | 0.233 | 0.348 | 0.166 | 0 | 1.035 | 0.266 | 15 |
| | 42 | 0.933 | 0.259 | 0.392 | 0.176 | 0 | 0.500 | 0.153 | 15 |
| | 48 | 1.361 | 0.266 | 0.589 | 0.231 | 0 | 0.578 | 0.169 | 5 |
| | 54 | 1.900 | 0.241 | 0.635 | 0.233 | 0 | 0.607 | 0.169 | 10 |
| | 60 | 2.223 | 0.270 | 0.728 | 0.238 | 0.028 | 0.649 | 0.153 | 0 |
| | 66 | 2.489 | 0.343 | 0.642 | 0.303 | 0 | 0.728 | 0.204 | 5 |
| | 72 | 2.695 | 0.248 | 0.671 | 0.243 | 0 | 0.559 | 0.164 | 5 |
| | 78 | 2.878 | 0.330 | 0.604 | 0.301 | 0 | 0.734 | 0.161 | 5 |
| 6 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 |
| | 18 | 0.019 | 0.192 | 0 | 0 | 0 | 0 | 0 | 100 |
| | 24 | 0.153 | 0.101 | 0.121 | 0.030 | 0 | 0.209 | 0.062 | 75 |
| | 30 | 0.429 | 0.183 | 0.207 | 0.100 | 0 | 0.457 | 0.150 | 45 |
| | 36 | 0.624 | 0.187 | 0.293 | 0.109 | 0 | 0.377 | 0.110 | 20 |
| | 42 | 1.154 | 0.234 | 0.567 | 0.222 | 0 | 0.758 | 0.224 | 10 |
| | 48 | 1.447 | 0.326 | 0.673 | 0.280 | 0 | 0.604 | 0.176 | 5 |
| | 54 | 1.884 | 0.317 | 0.876 | 0.311 | 0 | 0.764 | 0.221 | 5 |
| 7 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 |
| | 18 | 0.040 | 0.121 | 0.024 | 0 | 0 | 0 | 0 | 100 |
| | 24 | 0.441 | 0.172 | 0.050 | 0.013 | 0 | 0.150 | 0.037 | 85 |
| | 30 | 0.439 | 0.168 | 0.168 | 0.057 | 0 | 0.392 | 0.097 | 50 |
| | 36 | 0.702 | 0.185 | 0.415 | 0.126 | 0 | 0.605 | 0.148 | 25 |
| | 42 | 1.086 | 0.252 | 0.646 | 0.238 | 0 | 0.553 | 0.165 | 10 |
| 8 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 |
| | 18 | 0.020 | 0.047 | 0.001 | 0 | 0 | 0 | 0 | 100 |
| | 24 | 0.232 | 0.131 | 0.052 | 0.023 | 0 | 0.281 | 0.068 | 85 |
| | 30 | 0.410 | 0.183 | 0.166 | 0.065 | 0 | 0.254 | 0.084 | 45 |

Since the sequence obtained in Part 2 has a smaller total flow time, it is selected as the sequence generated by the heuristic.

## COMPUTATIONAL RESULTS

To evaluate the performance of the proposed heuristic, computer programs were written for both the DP solution and the heuristic. The computer programs were written in C language and run on a DEC 3400 UNIX computer with 96M bytes of memory. (It is noted that we usually don't have such a large memory space available in practice. The purpose of using the very large memory space is to observe the running time of the DP solution for large problems.) All the data were generated from the discrete uniform distribution but with different parameters. The lower and upper limits for processing times, minor setup times, and major setup times are [10,20], [1,15], and [10,25], respectively. The test problems were generated for three different numbers of families (2,3,4), each of which consists of five different numbers of classes (4,5,6,7,8). For each combination, the maximum size (number of jobs) of the problem that could be solved by the DP solution in no more than one hour (3600 s) was tested. Jobs were assigned to the classes as evenly as possible. For example, the job pattern for problems with 2 families, 5 classes, and 12 jobs was designed as follows:

Table 3. Computional results of the heuristic for three families

| | | % Errors | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Step 1 | Part 1 | Part 2 | Part 1+2 | | | | % Optimum of |
| No. of classes | No. of Jobs | Mean | Mean | Mean | Mean | Min. | Max. | Std. | Part 1+2 |
| 4 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 |
| | 18 | 0.314 | 0 | 0 | 0 | 0 | 0 | 0 | 100 |
| | 24 | 0.010 | 0.105 | 0 | 0 | 0 | 0 | 0 | 100 |
| | 30 | 0.309 | 0.154 | 0.106 | 0.075 | 0 | 0.375 | 0.143 | 70 |
| | 36 | 0.583 | 0.141 | 0.419 | 0.102 | 0 | 0.479 | 0.147 | 45 |
| | 42 | 0.900 | 0.117 | 0.456 | 0.111 | 0 | 0.539 | 0.136 | 35 |
| | 48 | 0.888 | 0.194 | 0.525 | 0.161 | 0 | 0.703 | 0.181 | 25 |
| | 54 | 1.348 | 0.225 | 0.833 | 0.205 | 0 | 0.594 | 0.165 | 15 |
| | 60 | 1.272 | 0.300 | 0.746 | 0.284 | 0 | 0.853 | 0.226 | 5 |
| | 72 | 1.747 | 0.205 | 0.809 | 0.205 | 0 | 0.601 | 0.169 | 5 |
| | 84 | 2.193 | 0.204 | 0.942 | 0.201 | 0 | 0.448 | 0.131 | 5 |
| | 96 | 2.916 | 0.294 | 0.662 | 0.274 | 0 | 0.648 | 0.193 | 5 |
| | 108 | 3.069 | 0.329 | 0.853 | 0.325 | 0.036 | 0.755 | 0.182 | 0 |
| | 120 | 3.676 | 0.221 | 0.624 | 0.209 | 0.007 | 0.521 | 0.130 | 0 |
| | 132 | 3.621 | 0.325 | 0.553 | 0.284 | 0 | 0.618 | 0.186 | 5 |
| | 144 | 4.246 | 0.246 | 0.410 | 0.220 | 0.039 | 0.454 | 0.127 | 0 |
| 5 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 |
| | 18 | 0.058 | 0.043 | 0 | 0 | 0 | 0 | 0 | 100 |
| | 24 | 0.076 | 0.140 | 0.008 | 0.008 | 0 | 0.158 | 0.035 | 95 |
| | 30 | 0.190 | 0.303 | 0.142 | 0.110 | 0 | 0.735 | 0.205 | 60 |
| | 36 | 0.493 | 0.239 | 0.287 | 0.150 | 0 | 0.592 | 0.176 | 30 |
| | 42 | 0.672 | 0.381 | 0.501 | 0.268 | 0 | 0.725 | 0.238 | 25 |
| | 48 | 0.970 | 0.272 | 0.531 | 0.197 | 0 | 0.698 | 0.190 | 25 |
| | 54 | 1.081 | 0.331 | 0.738 | 0.283 | 0 | 0.719 | 0.202 | 10 |
| | 60 | 1.604 | 0.406 | 0.871 | 0.370 | 0.020 | 1.143 | 0.280 | 0 |
| | 66 | 1.614 | 0.356 | 1.051 | 0.345 | 0.006 | 0.685 | 0.221 | 0 |
| | 72 | 1.958 | 0.388 | 0.948 | 0.334 | 0.003 | 0.712 | 0.189 | 0 |
| | 78 | 2.382 | 0.362 | 0.943 | 0.333 | 0 | 0.791 | 0.233 | 5 |
| 6 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 |
| | 18 | 0 | 0.090 | 0 | 0 | 0 | 0 | 0 | 100 |
| | 24 | 0.087 | 0.167 | 0.021 | 0.015 | 0 | 0 | 0 | 95 |
| | 30 | 0.320 | 0.248 | 0.179 | 0.038 | 0 | 0.284 | 0.084 | 65 |
| | 36 | 0.482 | 0.282 | 0.303 | 0.148 | 0 | 0.686 | 0.212 | 95 |
| | 42 | 0.431 | 0.237 | 0.333 | 0.171 | 0 | 0.399 | 0.132 | 15 |
| | 48 | 0.819 | 0.343 | 0.661 | 0.256 | 0 | 0.710 | 0.217 | 25 |
| | 54 | 1.280 | 0.301 | 0.855 | 0.294 | 0.008 | 0.584 | 0.164 | 0 |
| 7 | 12 | 0.020 | 0 | 0 | 0 | 0 | 0 | 0 | 100 |
| | 18 | 0.016 | 0.084 | 0 | 0 | 0 | 0 | 0 | 100 |
| | 24 | 0.022 | 0.217 | 0.007 | 0 | 0 | 0 | 0 | 100 |
| | 30 | 0.127 | 0.157 | 0.054 | 0.029 | 0 | 0.223 | 0.069 | 80 |
| | 36 | 0.347 | 0.253 | 0.183 | 0.124 | 0 | 0.638 | 0.182 | 40 |
| | 42 | 0.558 | 0.313 | 0.288 | 0.178 | 0 | 0.576 | 0.190 | 35 |
| 8 | 12 | 0.018 | 0.005 | 0.018 | 0 | 0 | 0 | 0 | 100 |
| | 18 | 0.173 | 0.086 | 0.138 | 0 | 0 | 0 | 0 | 100 |
| | 24 | 0.137 | 0.202 | 0.130 | 0.019 | 0 | 0.200 | 0.052 | 85 |
| | 30 | 0.135 | 0.185 | 0.063 | 0.046 | 0 | 0.465 | 0.115 | 75 |

| Family | Class | Jobs |
|---|---|---|
| 1 | 1 | $J_{11}$, $J_{12}$, $J_{13}$ |
| | 2 | $J_{21}$, $J_{22}$, $J_{23}$ |
| 2 | 3 | $J_{31}$, $J_{32}$ |
| | 4 | $J_{41}$, $J_{42}$ |
| | 5 | $J_{51}$, $J_{52}$ |

Based on 20 test problems in each configuration, Tables 2–4 provide information on the percentage error ((heuristic solution – optimal solution) / optimal solution) and percentage optimum (number of times that the heuristic produces optimal solution) of the heuristic for two, three, and four families, respectively. Several observations can be made from the table as follows:

(1) The proposed heuristic performs well. The average error deviation from the optimum for all the test problems is as low as 0.118%. For most combinations of families and classes, the average optimum is higher than 50% when the number of jobs is no more than 30.
(2) For problems with a small number of jobs (e.g., no more than 30 jobs for 2 families with 4 classes),

Table 4. Computional results of the heuristic for four families

| | | % Errors | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Step 1 | Part 1 | Part 2 | Part 1+2 | | | | % Optimum of |
| No. of classes | No. of Jobs | Mean | Mean | Mean | Mean | Min. | Max. | Std. | Part 1+2 |
| 4 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 |
| | 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 |
| | 24 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 |
| | 30 | 0.113 | 0.027 | 0.048 | 0.027 | 0 | 0.336 | 0.086 | 90 |
| | 36 | 0.189 | 0.055 | 0.148 | 0.055 | 0 | 0.365 | 0.108 | 65 |
| | 42 | 0.380 | 0.044 | 0.305 | 0.044 | 0 | 0.290 | 0.085 | 70 |
| | 48 | 0.670 | 0.068 | 0.555 | 0.054 | 0 | 0.253 | 0.075 | 45 |
| | 54 | 1.010 | 0.108 | 0.802 | 0.108 | 0 | 0.535 | 0.134 | 25 |
| | 60 | 1.169 | 0.109 | 0.731 | 0.109 | 0 | 0.441 | 0.139 | 30 |
| | 72 | 1.392 | 0.126 | 0.869 | 0.126 | 0 | 0.417 | 0.122 | 15 |
| | 84 | 1.906 | 0.167 | 1.259 | 0.167 | 0.006 | 0.605 | 0.146 | 0 |
| | 96 | 2.205 | 0.142 | 1.186 | 0.131 | 0 | 0.261 | 0.086 | 10 |
| | 108 | 2.626 | 0.238 | 1.354 | 0.238 | 0 | 0.606 | 0.160 | 5 |
| | 120 | 3.216 | 0.216 | 1.336 | 0.216 | 0.027 | 0.454 | 0.126 | 0 |
| | 132 | 3.186 | 0.158 | 1.166 | 0.158 | 0.009 | 0.424 | 0.128 | 0 |
| | 144 | 3.470 | 0.212 | 1.064 | 0.211 | 0 | 0.452 | 0.126 | 5 |
| 5 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 |
| | 18 | 0 | 0.068 | 0 | 0 | 0 | 0 | 0 | 100 |
| | 24 | 0.098 | 0.136 | 0.013 | 0.013 | 0 | 0.252 | 0.056 | 95 |
| | 30 | 0.100 | 0.149 | 0.082 | 0.017 | 0 | 0.224 | 0.056 | 90 |
| | 36 | 0.144 | 0.094 | 0.068 | 0.027 | 0 | 0.100 | 0.033 | 55 |
| | 42 | 0.506 | 0.233 | 0.339 | 0.185 | 0 | 0.498 | 0.177 | 25 |
| | 48 | 0.562 | 0.323 | 0.428 | 0.221 | 0 | 0.606 | 0.195 | 10 |
| | 54 | 1.010 | 0.357 | 0.799 | 0.344 | 0.074 | 0.797 | 0.227 | 0 |
| | 60 | 1.094 | 0.202 | 0.863 | 0.199 | 0 | 0.653 | 0.165 | 15 |
| | 66 | 1.253 | 0.267 | 0.962 | 0.259 | 0.039 | 1.113 | 0.249 | 0 |
| | 72 | 1.340 | 0.262 | 1.009 | 0.262 | 0 | 0.704 | 0.197 | 5 |
| | 78 | 2.019 | 0.408 | 1.216 | 0.408 | 0.021 | 0.899 | 0.262 | 0 |
| 6 | 12 | 0.032 | 0 | 0 | 0 | 0 | 0 | 0 | 100 |
| | 18 | 0.015 | 0.091 | 0 | 0 | 0 | 0 | 0 | 100 |
| | 24 | 0.056 | 0.161 | 0 | 0 | 0 | 0 | 0 | 100 |
| | 30 | 0.047 | 0.236 | 0.012 | 0.005 | 0 | 0.058 | 0.014 | 85 |
| | 36 | 0.154 | 0.266 | 0.091 | 0.091 | 0 | 0.517 | 0.134 | 45 |
| | 42 | 0.418 | 0.288 | 0.288 | 0.199 | 0 | 0.572 | 0.168 | 15 |
| | 48 | 0.607 | 0.296 | 0.510 | 0.240 | 0.010 | 0.564 | 0.166 | 0 |
| | 54 | 0.885 | 0.294 | 0.611 | 0.262 | 0 | 0.691 | 0.168 | 5 |
| 7 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 |
| | 18 | 0.027 | 0.248 | 0 | 0 | 0 | 0 | 0 | 100 |
| | 24 | 0.149 | 0.116 | 0.092 | 0.014 | 0 | 0 | 0 | 100 |
| | 30 | 0.061 | 0.039 | 0.012 | 0.008 | 0 | 0.162 | 0.036 | 95 |
| | 36 | 0.096 | 0.221 | 0.019 | 0.016 | 0 | 0.190 | 0.047 | 85 |
| | 42 | 0.319 | 0.307 | 0.190 | 0.129 | 0 | 0.546 | 0.150 | 25 |
| 8 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 |
| | 18 | 0.027 | 0.248 | 0 | 0 | 0 | 0 | 0 | 100 |
| | 24 | 0.149 | 0.116 | 0.092 | 0.014 | 0 | 0.220 | 0.051 | 90 |
| | 30 | 0.058 | 0.185 | 0.005 | 0.005 | 0 | 0.107 | 0.024 | 95 |

Part 2 outperforms Part 1. As the number of jobs increases, Part 1 turns out to be superior. Moreover, for the same number of classes, the break point is increased as the number of families increases. In many cases, the two parts may offset each others' deviations from the optimum.

(3) The performance of the heuristic does not deteriorate significantly as the number of jobs increases, especially for a smaller number of classes. For a fixed number of jobs, the performance of the heuristic does not vary significantly for different numbers of classes. Also, for fixed numbers of classes and jobs, the performance of the heuristic does not vary significantly for different numbers of families.

(4) By comparing the columns of Step 1, Part 1, and Part 2, it is observed that the degree of improvement reached in Step 2 of Part 1 and Part 2 of the heuristic is significant, especially for problems with a large number of jobs.

In addition, we also compare the required computation time of the DP solution with that of the proposed heuristic. The results for three families are summarized in Table 5. It is observed that the computation time of the DP solution increases at a rate much higher than that of the heuristic. This is

Table 5. Required computation times of the DP solution and the proposed heuristic for three-family problems (s)

| No. of Classes | No. of Jobs | DP | | | | Heuristic | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Mean | Min. | Max. | Std. | Mean | Min. | Max. | Std. |
| 4 | 12 | 0.007 | 0.006 | 0.009 | 0.001 | 0.005 | 0.005 | 0.006 | 0 |
| | 18 | 0.030 | 0.028 | 0.033 | 0.001 | 0.016 | 0.015 | 0.022 | 0.002 |
| | 24 | 0.142 | 0.091 | 0.362 | 0.079 | 0.048 | 0.034 | 0.140 | 0.025 |
| | 30 | 0.380 | 0.215 | 0.509 | 0.106 | 0.120 | 0.066 | 0.239 | 0.045 |
| | 36 | 0.910 | 0.468 | 1.148 | 0.167 | 0.197 | 0.114 | 0.288 | 0.054 |
| | 42 | 1.725 | 0.891 | 1.888 | 0.205 | 0.354 | 0.179 | 0.489 | 0.062 |
| | 48 | 3.275 | 2.574 | 3.380 | 0.172 | 0.544 | 0.423 | 0.707 | 0.072 |
| | 54 | 5.562 | 4.538 | 6.653 | 0.356 | 0.778 | 0.644 | 0.944 | 0.076 |
| | 60 | 8.734 | 7.855 | 9.008 | 0.243 | 1.079 | 0.898 | 1.255 | 0.079 |
| | 72 | 19.462 | 18.764 | 19.752 | 0.194 | 1.806 | 1.704 | 1.964 | 0.071 |
| | 84 | 38.936 | 38.546 | 39.570 | 0.287 | 2.897 | 2.785 | 3.067 | 0.074 |
| | 96 | 72.289 | 71.064 | 78.844 | 2.126 | 4.499 | 4.202 | 6.484 | 0.485 |
| | 108 | 123.248 | 121.340 | 137.782 | 3.465 | 6.115 | 5.936 | 6.466 | 0.142 |
| | 120 | 205.112 | 202.035 | 218.234 | 3.465 | 8.671 | 8.314 | 9.606 | 0.357 |
| | 132 | 311.154 | 307.389 | 329.949 | 5.440 | 11.319 | 11.001 | 12.813 | 0.397 |
| | 144 | 479.752 | 459.140 | 521.003 | 18.570 | 15.087 | 14.042 | 16.137 | 0.607 |
| | 156 | >3600.000 | | | | 17.802 | 17.388 | 18.933 | 0.503 |
| 5 | 12 | 0.017 | 0.016 | 0.019 | 0.001 | 0.005 | 0.005 | 0.007 | 0.001 |
| | 18 | 0.139 | 0.089 | 0.249 | 0.049 | 0.038 | 0.015 | 0.214 | 0.046 |
| | 24 | 0.680 | 0.354 | 0.907 | 0.130 | 0.060 | 0.035 | 0.125 | 0.026 |
| | 30 | 2.135 | 1.781 | 2.252 | 0.101 | 0.131 | 0.070 | 0.212 | 0.037 |
| | 36 | 5.314 | 5.040 | 5.451 | 0.092 | 0.221 | 0.119 | 0.351 | 0.063 |
| | 42 | 11.392 | 11.163 | 11.525 | 0.099 | 0.396 | 0.255 | 0.579 | 0.078 |
| | 48 | 23.523 | 22.588 | 33.161 | 2.455 | 0.610 | 0.468 | 0.883 | 0.099 |
| | 54 | 42.020 | 41.764 | 42.608 | 0.185 | 0.812 | 0.684 | 0.897 | 0.056 |
| | 60 | 73.448 | 73.125 | 73.900 | 0.241 | 1.135 | 0.981 | 1.276 | 0.068 |
| | 66 | 121.021 | 119.521 | 136.887 | 3.760 | 1.483 | 1.400 | 1.618 | 0.071 |
| | 72 | 190.562 | 188.887 | 205.491 | 3.578 | 1.910 | 1.806 | 2.036 | 0.067 |
| | 78 | 318.076 | 315.167 | 332.758 | 5.033 | 2.436 | 2.251 | 2.584 | 0.095 |
| | 84 | >3600.000 | | | | 3.270 | 3.069 | 4.420 | 0.356 |
| 6 | 12 | 0.056 | 0.037 | 0.271 | 0.055 | 0.005 | 0.005 | 0.007 | 0.001 |
| | 18 | 0.539 | 0.288 | 0.711 | 0.115 | 0.036 | 0.017 | 0.122 | 0.028 |
| | 24 | 2.460 | 2.017 | 2.624 | 0.121 | 0.073 | 0.037 | 0.195 | 0.045 |
| | 30 | 8.628 | 8.352 | 8.816 | 0.102 | 0.143 | 0.073 | 0.239 | 0.050 |
| | 36 | 24.273 | 24.055 | 24.518 | 0.105 | 0.241 | 0.125 | 0.326 | 0.049 |
| | 42 | 62.37 | 62.035 | 62.687 | 0.185 | 0.415 | 0.318 | 0.603 | 0.072 |
| | 48 | 153.373 | 151.346 | 167.928 | 3.490 | 0.615 | 0.534 | 0.704 | 0.050 |
| | 54 | 352.313 | 347.153 | 370.284 | 6.950 | 0.845 | 0.680 | 0.949 | 0.069 |
| | 60 | >3600.000 | | | | 1.152 | 1.140 | 1.268 | 0.028 |
| 7 | 12 | 0.088 | 0.058 | 0.282 | 0.054 | 0.009 | 0.005 | 0.053 | 0.011 |
| | 18 | 1.136 | 0.587 | 1.278 | 0.145 | 0.032 | 0.016 | 0.141 | 0.032 |
| | 24 | 8.202 | 7.755 | 8.354 | 0.127 | 0.073 | 0.037 | 0.215 | 0.041 |
| | 30 | 31.232 | 31.026 | 31.507 | 0.128 | 0.143 | 0.096 | 0.233 | 0.044 |
| | 36 | 123.485 | 122.081 | 138.379 | 3.536 | 0.249 | 0.125 | 0.345 | 0.059 |
| | 42 | 365.056 | 361.979 | 379.281 | 4.779 | 0.417 | 0.295 | 0.555 | 0.067 |
| | 48 | >3600.000 | | | | 0.578 | 0.568 | 0.597 | 0.008 |
| 8 | 12 | 0.157 | 0.100 | 0.351 | 0.073 | 0.007 | 0.005 | 0.033 | 0.006 |
| | 18 | 2.943 | 2.456 | 3.114 | 0.133 | 0.032 | 0.016 | 0.114 | 0.028 |
| | 24 | 30.842 | 30.547 | 31.704 | 0.246 | 0.092 | 0.037 | 0.209 | 0.052 |
| | 30 | 116.004 | 114.482 | 131.704 | 3.650 | 0.145 | 0.074 | 0.294 | 0.056 |
| | 36 | >3600.000 | | | | 0.253 | 0.246 | 0.260 | 0.004 |

especially true when the number of classes is relatively large. Also, for the same number of jobs, the computation time of the heuristic is almost unchanged as the number of classes increases, while the required computation time of the DP solution increases exponentially. In fact, these results can be expected by referring to the time complexities of the two methods.

## CONCLUSIONS

The problem of single-facility scheduling with major and minor setups has been considered in this article. A DP formulation has been given to minimize the criterion of the mean flow time. Recognizing the large storage space required by the DP solution, a DP-based heuristic has been developed. The heuristic is not only conceptually easy to understand, but also produces satisfactory results. Computational results show that the heuristic produces solutions with average 0.118% deviations from the optimum. For most combinations of families and classes, the average optimum is higher than 50% when the number of jobs is no more than 30. The framework of the proposed heuristic may be adapted to solve other scheduling problems with a multiple-class environment.

The problem of scheduling with multiple-stage setups occurs frequently in the manufacturing industry. However, because of the difficulty of solving the problem to optimality, little research has been conducted in this area. It is hoped that this article, which deals with a problem with two-stage setups, may stimulate other research in the direction of multiple-stage setups.

## REFERENCES

1. Bedworth, D. D. and Bailey, J. E., *Integrated Production Control System.* Wiley, New York, 1987.
2. Gupta, J. N. D., Optimal schedules for single facility with classes. *Comp. Ops Res.*, 1984, **11**, 409–413.
3. Potts, C. N., Scheduling two job classes on a single machine. *Comp. Ops Res.*, 1991, **18**, 411–415.
4. Gupta, J. N. D., Single facility scheduling with multiple job classes. *Eur. J. Opl Res.*, 1988, **33**, 42–45.
5. Ahn, B. H. and Hyun, J. H., Single facility multi-class job scheduling. *Comp. Ops Res.*, 1990, **17**, 265–272.
6. Psaraftis, H. N., A dynamic programming approach for sequencing groups of identical jobs. *Ops Res.*, 1980, **28**, 1347–1359.
7. Tang, C. S., Scheduling batches on parallel machine with major and minor set-ups. *Eur. J. Op. Res.*, 1990, **46**, 28–37.
8. Leong, G. K. and Oliff, M. D., A sequencing heuristic for dependent setups in a batch process industry. *OMEGA, Int. J. Mgmt Sci.*, 1990, **18**, 283–297.
9. Ozden, M., Egbelu, P. J. and Iyer, A. V., Job scheduling in a group technology environment for a single facility. *Comp. Ind. Engng.*, 1985, **9**, 67–72.
10. Dreyfus, S. E. and Law, A. M., *The Art and Theory of Dynamic Programming.* Academic, New York, 1977.
11. Conway, R. W., Maxwell, W. L. and Miller, L. W., *Theory of Scheduling.* Addison–Wesley, Reading, Mass, 1967.