Intelligent Information Management, 2010, 2, 188-198
doi:10.4236/iim.2012.23022 Published Online March 2010 (http://www. SciRP.org/journal/iim)

Efficient Heuristic to Minimize Makespan in Single Machine Scheduling Problem with Unrelated Parallel Machines

P. Sivasankaran[1], T. Sornakumar[1], R. Panneerselvam[2]
[1]*Department of Mechanical Engineering, Thiagarajar College of Engineering, Madura, India*
[2]*Department of Management Studies, School of Management, Pondicherry University, Pondicherry, India*
E-mail: *sivasankaran.panneerselvam@yahoo.com, sornakumar*2000*@yahoo.com, panneer_dms@yahoo.co.in*
Received November 23, 2009; *revised December* 30, 2009; *accepted February* 2, 2010

Abstract

This paper discusses an efficient heuristic to minimize the makespan of scheduling *n* independent jobs on *m* unrelated parallel machines. The problem of scheduling the jobs on the unrelated parallel machines is combinatorial in nature. Hence, the heuristic approach is inevitable for quicker solution. In this paper, a simple and efficient heuristic is designed to minimize the makespan of scheduling *n* independent jobs on *m* unrelated parallel machines. A mathematical model is also presented for this problem. A factorial experiment is used to compare the results of the proposed heuristic with that of a mathematical model by taking "Method" (Heuristic and Model) as the first factor and "Problem Size" (No. of machines X No. of Jobs: 2X5, 2X6, ……, 2X9, 3X5, 3X6, ……, 3X9, ……., 5X5, 5X6, …5X9) as the second factor. It is found that there is no significant difference between the results of the proposed heuristic and that of the mathematical model. Further, the mean percent error of the results obtained by the heuristic from the optimal results obtained by the model is 2.336 %. The heuristic gives optimal solution for 76.67 % of the problems.

Keywords: Makespan, Heuristic, Unrelated Parallel Machines, Mathematical Model, ANOVA

1. Introduction

The production scheduling problem is classified into single machine scheduling problem, flow shop scheduling problem, job shop scheduling problem, open shop scheduling problem and batch scheduling problem. In this paper, the single machine scheduling problem is considered. The single machine scheduling problem is further classified into single machine scheduling problem with single machine (processor) and single machine scheduling problem with parallel machines (processors). The single machine scheduling problem with parallel machines is again classified into single machine scheduling problem with identical parallel machines, single machine scheduling with uniform parallel machines and single machine scheduling with unrelated parallel machines, which are as explained below.

Let, t_{ij} be the processing time of the job j on the machine i, for i = 1, 2, 3, …., *m* and j = 1 2, 3, …, *n*.

The three types of parallel machines scheduling problem are defined as follows.

1) If $t_{ij} = t_{1j}$ for all i and j, then the problem is called as *identical parallel machines scheduling problem*.

This means that all the parallel machines are identical in terms of their speed. Each and every job will take the same amount of processing time on each of the parallel machines.

2) If $t_{ij} = t_{1j}/s_i$ for all i and j, then the problem is termed as *uniform (proportional) parallel machines scheduling problem*.

Here, s_i is the speed of the machine i and t_{1j} is the processing time of the job j on the machine 1. This means that the parallel machines will have different speeds. The relationship among the speeds of the machines is $s_1 < s_2 < s_3 < \ldots < s_m$. That is the machine 1 is the slowest machine and the machine *m* is the fastest machine. So, the relationship among the processing times of a job on the parallel machines is $1/s_1 > 1/s_2 > 1/s_3 > \ldots > 1/s_m$.

3) If t_{ij} is arbitrary for all i and j, then the problem is known as *unrelated parallel machines scheduling problem*.

In this type of scheduling, there will not be any relationship among the processing times of a job on the parallel machines. This may be due to technological differences among the machines, different features of the jobs, etc.

In this paper, the single machine scheduling problem with unrelated parallel machines is considered. The essential characteristics of the single machine scheduling problem with unrelated parallel machines are listed as follows.

1) It has *n* independent single operation jobs.

2) It has *m* parallel machines.

3) *m* machines are continuously available and they are never kept idle while a work is waiting.

4) t_{ij} is the processing time of the job j on the machine i, for i = 1, 2, 3, .., *m* and j = 1, 2, 3, …, *n*.

5) t_{ij} is arbitrary for different combinations of i and j, for i = 1, 2, 3, .., *m* and j = 1, 2, 3, …, *n*.

Environment of Unrelated Parallel Machines Scheduling Problem

Consider a situation in which we have *m* different parallel machines and *n* independent single operation jobs. Each of the jobs can be processed on each of the unrelated parallel machines. In this process, the processing times of each job on different machines are arbitrary. Like in the uniform parallel machines scheduling ($t_{ij} = t_{1j}/s_i$ for all i and j) or in the identical parallel machines scheduling ($t_{ij} = t_{1j}$ for all i and j), there will not be any relationship among the processing times of a job on the parallel machines of the unrelated parallel machines scheduling.

In a shaft, if a hole is to be made at one of its ends, it can be done either in a lathe or in a drilling machine. The time taken to make the end-hole in these machines will differ, because the type of setting and the processing technology are altogether different in these machines. Hence, in the single machine scheduling problem with unrelated parallel machines, the processing times of each job on different machines are arbitrary. Like this example, in industries, one can encounter many situations. Hence, the machines which have similar processing capabilities will be grouped together and a batch of single operation jobs will be scheduled on these machines to optimize a desired measure of performance.

There are many measures of performance in the single machine scheduling problem [1]. In this paper, the minimization of the makespan of scheduling *n* independent single operation jobs on *m* unrelated parallel machines is considered, because it is considered to be a dominant measure of performance, which represents the earliest completion time of the given batch of jobs.

2. Literature Review

This section presents review of literature of scheduling *n* independent single operation jobs on *m* unrelated parallel machines to minimize the makespan. Since, this problem is NP-hard, any attempt to obtain the optimal solution through exact algorithm may end with failure for most of the instances. This is because of exponential computational time to solve such problem. Hence, researchers are focusing on the development of heuristics, which will give near optimal solution. Van De Velde [2] considered the single machine scheduling problem with unrelated parallel machines. The author aimed to minimize the maximum job completion time which means the minimization of makespan. He presented an optimization algorithm and an approximation algorithm that are based on surrogate relaxation and duality to solve this NP hard problem. The idea behind the surrogate relaxation is to replace a set of nasty (complex) constraints with a single constraint that is a weighted aggregate of these constraints.

Glass, Potts and Shade [3] have applied meta-heuristics to the problem of scheduling jobs on the unrelated parallel machines to minimize the makespan and reported that genetic algorithm gives poor results. Also, they reported that a hybrid method in which a descent is incorporated into the genetic algorithm is comparable in performance with the simulated annealing. Francis Sourd [4] has developed two approximation algorithms for minimizing the makespan of independent tasks assigned on the unrelated parallel machines. The first one is based on a partial and heuristic exploration of a search tree. In the second algorithm, a new large neighbourhood improvement procedure is implemented in an already existing algorithm. He reported that the computational efficiencies of these algorithms are equivalent to the best local search heuristic.

Mokotoff and Chretienne [5] have developed a cutting plane algorithm for the unrelated parallel machine scheduling problem, which minimizes the makespan. Their algorithm deals with the polyhedral structure of this scheduling problem. In their work, strong inequalities are identified for fixed values of the maximum completion time and they are used to build a cutting plane scheme from which an exact algorithm and an approximation algorithm are developed. Mokotoff and Jimeno [6] developed heuristics based on partial enumeration for the unrelated parallel machines scheduling problem. For a given mixed integer linear model with binary decision variables, they presented heuristics based on the partial enumeration. Computational experiments are reported for a collection of test problems, showing that some of the proposed algorithms achieve better solutions than other relevant existing approximation algorithms.

Hariri and Potts [7] developed a heuristic, which consists of two phases to minimize the makespan of scheduling *n* independent jobs on *m* unrelated parallel machines. In the first phase, a linear programming model is used to schedule some of the jobs and then a heuristic is used in the second phase to schedule the remaining jobs.

They have stated that some improvement procedure is necessary to have good solutions. Piersma and Van Dijk [8] have proposed a set of local search algorithms to minimize the makespan of scheduling jobs on the unrelated parallel machines. In these algorithms, the neighbourhood search of a solution uses the efficiency of the machines for each job. They claimed that this approach gives better results when compared to general local search algorithms.

Ghirardi and Potts [9] have considered the problem of scheduling jobs on the unrelated parallel machines to minimize the makespan. They developed a method known as recovering beam search method to minimize the makespan in the unrelated parallel machines. The traditional beam search method is a truncated version of branch and bound method. The recovering beam search allows the possibility of correcting wrong decisions by replacing partial solutions with others. It has a polynomial time complexity function for the *NP* hard problem of this research. Also, they reported the computational results of this method.

Monien and Woclaw [10] have presented an experimental study on the unspittable-Truemper algorithm to minimize the makespan of scheduling jobs on the unrelated parallel machines. This computes 2-approximate solutions in the best worst-case running time known so far. The goal of their simulation was to prove its efficiency in practice. They compared their technique with the algorithms and heuristics in practice, especially with those based on *two-step approach.* Gairing, Monien and Woclaw [11] presented a combinatorial approximation algorithm that matches an integral 2-approximation quality. It is a generic minimum cost flow algorithm, without any complex enhancements, tailored to handle unsplittable flow. In their approach, they replaced the classical technique of solving LP-relaxation and rounding afterwards by a completely integral approach.

From these literatures, it is clear that the development of heuristic to minimize the makepsan of scheduling *n* independent jobs on *m* unrelated parallel machines is a challenging task. Various authors have proposed different methodologies. The range of heuristics varies from simple search procedure to meta-heuristic. In this paper, an attempt has been made to develop an efficient heuristic to minimize the makepsan of scheduling jobs on the unrelated parallel machines and compare its solution with that of a mathematical model, which gives optimal solution. If the solution of the proposed heuristic does not differ significantly from that of the model, for a given set of randomly generated problems, then one can conclude that the proposed heuristic is an efficient one.

For a given solution methodology, say the heuristic development, if a heuristic performs better, it is considered to be an efficient heuristic. Further, among different solution methodologies, viz., simple heuristics, meta heuristics, model, etc., if a simple heuristic proves to be performing better, then that simple heuristic can be termed as an effective heuristic. In this paper, the authors propose an efficient heuristic which in turn proves to be effective, based on a comparison of the results of this heuristic with that of a mathematical model.

3. Mathematical Model to Minimize Makespan of Scheduling Jobs on Unrelated Parallel Machines

The optimal makespan of scheduling *n* independent single operation jobs on *m* unrelated parallel machines can be obtained using a mathematical model. In this section, a mathematical model is presented for this problem.

Let, *n* be the number of independent jobs with single operation.

m be the number of unrelated parallel machines.

t_{ij} be the processing time of the job j on the machine i and it is arbitrary for different combinations of i and j.

A generalized data format of this problem is shown in **Table 1**. The objective is to schedule *n* jobs on *m* unrelated parallel machines such that the makespan is minimized.

The expression for the makespan, *M* is given as follows.

$$M = \text{Max}\left\{\sum_{j=1}^{n} t_{ij}x_{ij}, \quad i=1,2,3,\ldots,m\right\}$$

where, x_{ij} = 1, if the job j is assigned to the machine i
= 0, otherwise, for i = 1, 2, 3, ….., *m* and j = 1, 2, 3, ….., *n*.

A descriptive mathematical model to minimize the makespan is as follows:

$$\text{Minimize} \quad Z = \text{Max}\left\{\sum_{j=1}^{n} t_{ij}x_{ij}, \quad i=1,2,3,\ldots,m\right\}$$

Table 1. Generalized format of processing times.

		Job j						
		1	2	3	.	j	.	*n*
	1	t_{11}	t_{12}	t_{13}	.	t_{1j}	.	t_{1n}
	2	t_{21}	t_{22}	t_{23}	.	t_{2j}	.	t_{2n}
Machine i	3	t_{31}	t_{32}	t_{33}	.	t_{3j}	. .	t_{3n}
	.	.	.	.	.	.	.	.
	i	t_{i1}	t_{i2}	t_{i3}	.	t_{ij}	.	t_{in}
	.	.	.	.	.	.	.	.
	m	t_{m1}	t_{m2}	t_{m3}	.	t_{mj}	.	t_{mn}

subject to

$$\sum_{i=1}^{m} x_{ij} = 1, \quad j=1,2,3,\ldots,n$$

where, $x_{ij} = 1$, if the job j is assigned to the machine i
= 0, otherwise, for i = 1, 2, 3, ….., *m* and j = 1, 2, 3, ….., *n*

The number of variables in this model is *mn* and the total number of constraints is *n*. In this model, the objective function is not in workable form. So, an alternate form of this model as follows contains a linear objective function, which is in workable form.

Minimize $Z_1 = M$

$$M - \sum_{j=1}^{n} t_{ij} x_{ij} \geq 0, \quad i=1,2,3,\ldots,m$$

$$\sum_{i=1}^{m} x_{ij} = 1, \quad j=1,2,3,\ldots,n$$

where, $x_{ij} = 1$, if the job j is assigned to the machine i = 0, otherwise, for i = 1, 2, 3, …., *m* and j = 1, 2, 3, …, *n*

$M \geq 0$ and it is the makespan of the schedule.

In this model, the number of variables is *mn*+1 and the total number of constraints is *m*+*n*. This problem comes under combinatorial category. Hence, the development of an efficient heuristic for this problem is highly essential.

4. Heuristic to Minimize Makespan of Scheduling Jobs on Unrelated Parallel Machines

As stated earlier, the problem of scheduling *n* independent single operation jobs on *m* unrelated parallel machines to minimize the makespan is a combinatorial problem. Hence, the time taken to obtain the optimal solution using either a mathematical model or a complete enumeration technique or a branch and bound technique will grow exponentially with respect to the increase in the problem size. Hence, the usage of a heuristic to overcome this situation is highly inevitable. In this section, the authors present an efficient heuristic to schedule *n* independent jobs on *m* unrelated parallel machines such that the makespan is minimized.

4.1. Rationale of the Heuristic

The proposed heuristic consists of two phases, which are listed as follows.

1) Initial assignment of jobs to the machines.
2) Shifting of jobs from one machine to another.

Phase 1

In this phase, each and every job is assigned to the machine on which it takes the least processing time.

Let the maximum of the completion times of the last jobs on all the machines be *M,* which is known as the initial makespan of the schedule.

M = Max $\{C_i\}$, where C_i is the completion time of the last job on the machine i,
where i = 1, 2, 3, …,*m*.

Phase 2

After having assigned the jobs to the machines as explained in the phase 1, in this phase, an attempt is made to minimize the makespan by the shifting jobs from one machine to other machines.

Consider the generalized data with three machines and 7 jobs as shown in **Table 2**.

Let the initial assignment of the jobs on the three machines be as shown in **Figure 1**.

From the **Figure 1**, it is clear that the makespan (*M*) is C_3, which is the maximum of C_1, C_2 and C_3. This maximum occurs on the machine 3. If the current makespan is to be reduced, then each of the jobs on the machine 3 is to be shifted to some other machine, provided such shifting of the job is economical in terms of reduction in makespan.

If the job 1 on the machine 3 is considered for shifting to machine 1, then the necessary calculations are as shown below.

The completion time of the last job on the machine 1 after the shift = $C_1 + t_{11}$.

The completion time of the last job on the machine 3 after the shift = $C_3 - t_{31}$.

Table 2. Generalized format of processing times for 3 machines and 7 jobs.

		Job j						
		1	2	3	4	5	6	7
Machine i	1	t_{11}	t_{12}	t_{13}	t_{14}	t_{15}	t_{16}	t_{17}
	2	t_{21}	t_{22}	t_{23}	t_{24}	t_{25}	t_{26}	t_{27}
	3	t_{31}	t_{32}	t_{33}	t_{34}	t_{35}	t_{36}	t_{37}

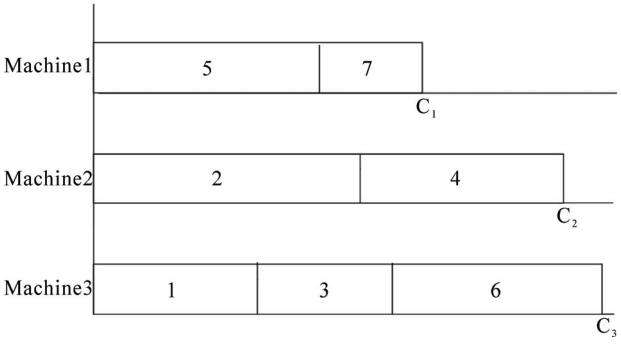

Figure 1. Gantt chart for initial assignment of jobs.

After this shift, let the maximum of the completion times of the last jobs on all the machines be S. If $S < M$, then the shifting of the job 1 to the machine 1 is to be effected; otherwise, the shifting of the job 1 to the machine 1 is to be avoided. Such shifting of jobs is to be carried out until there is no further reduction in the makespan.

4.2. Steps of the Heuristic

The steps of the proposed heuristic are presented in this section.

***Step* 1:** Input the data:

Number of independent jobs, n

Number of unrelated parallel machines, m

Processing time $T(I,J)$, $I = 1$ to m and $J = 1$ to n.

Set Number of Jobs on Machine I, $NJ(I) = 0$, $I = 1$ to m

Set Machine Completion Time, $MCT(I) = 0$, $I = 1$ to m

***Step* 2:** Assign the jobs to the machines.

Step 2.1: Set the Job Index $J = 1$

Step 2.2: Find the machine which takes the least processing time for the job J.

Let that machine be K.

Step 2.3: Do the following:

1) Increment the number of jobs on the machine K by 1

$NJ(K) = NJ(K) + 1$

2) Add the job J to the job sequence $JS[\ ]$ of the machine K at $NJ(K)^{th}$ position.

$JS[K, NJ(K)] = J$

3) Add the processing of the job J to the machine completion time of the machine K.

$MCT(K) = MCT(K) + T(K,J)$

Step 2.4: $J = J + 1$

Step 2.5: If $J \leq n$ then go to Step 2.2; else go to Step 3.

***Step* 3:** Set *Job Status Index*, *JSI* to $n+1$

***Step* 4**: *Find the machine whose last job completion time is the maximum (makespan)*

Step 4.1: $CMAX = 0$

Step 4.2: $I = 1$

Step 4.3: If $MCT(I\ ) > CMAX$ then update the maximum completion time,

$CMAX = MCT(I\ )$

Set machine with maximum completion time, $MCMAX = I$

Step 4.4: $I = I + 1$

Step 4.5: If $I \leq m$ then go to Step 4.3; else go to Step 4.6.

Step 4.6: Set the best makespan, $BMS = CMAX$.

***Step* 5:** *Shift the jobs on the machine with BMS to some other machines which give maximum reduction in makespan*

Step 5.1: *Initialize the job position on the machine MCMAX to 1* [$X = 1$]

Step 5.2: Store the job at position X on the Machine *MCMAX* in Q.

$Q = JS(MCMAX,X)$

Step 5.3: Find the machine *BMC1*, other than *MCMAX* on which the job Q requires the least time

Step 5.4: Find the completion time on the identified machine to which the job Q is to be transferred,

$BMCCT = MCT(BMC1) + T(BMC1,Q)$

Find completion time on the machine *MCMAX* from which, the job Q is to be transferred,

$MAXCT1 = MCT(MCMAX) - T(MCMAX,Q)$

Step 5.5: Find the maximum of the completion times of the last jobs on all the machines.

Let it be *MAXMCT*.

Step 5.6: If $MAXMCT \geq CMAX$ then set Least Makespan, $LMS1 = 10000$ and then go to Step 5.7;

Else, set Least Makespan, $LMS1 = MAXMCT$ and then go to Step 5.7.

Step 5.7: Find the machine *BMC2*, other than the machine *MCMAX*, with the least completion time for the last job.

Step 5.8: Find the completion time on the identified machine to which the job Q is to be transferred,

$BMCCT = MCT(BMC2) + T(BMC2,Q)$

Find completion time on the machine *MCMAX* from which the job Q is to be transferred,

$MAXCT2 = MCT(MCMAX) - T(MCMAX,Q)$

Step 5.9: Find the maximum of the completion times of the last jobs on all the machines. Let it be *MAXMCT*.

Step 5.10: If $MAXMCT \geq CMAX$ then set Least Makespan, $LMS2 = 10000$ and then go to Step 5.11;

Else, set Least Makespan, $LMS2 = MAXMCT$ and then go to Step 5.11.

Step 5.11: If $LMS1 > LMS2$ then

{Least of Least Makespans, $LMS = LMS2$

$INDEX = 2$

Best Machine, $BBMC2 = BMC2$

}

Else

{Least of Least Makespans, $LMS = LMS1$

$INDEX = 1$

Best Machine, $BBMC1 = BMC1$

}

Step 5.12: If $LMS = 10000$ then go to Step 5.19

Step 5.13: If $LMS \geq BMS$ then go to Step 5.19

Step 5.14: Update the Best Makespan, $BMS = LMS$

Update Job Status Index, $JSI = Q$

Step 5.15: Update the results w.r.t. *BMS* on the machine *MCMAX* as the best solution.

Transfer current machine completion times,

$MCT(I\ )$ to $BMCT(I\ )$ for all machines ($I = 1, 2, 3, \ldots, m$)

Transfer current number of jobs on each machine,

$NJ(I\ )$ to $BNJ(I\ )$ for all machines ($I = 1, 2, 3, \ldots, m$)

Transfer current job sequences on machines,

$JS(I,J)$ to $BJS(I,J)$ for all $I = 1, 2, 3, \ldots, m$ and $J = 1, 2, 3, \ldots, NJ(I\ )$

Step 5.16: If $INDEX = 1$ then go to Step 5.17; else go to Step 5.18.

Step 5.17: Update makespan and other results based on the allocation of the job Q to the machine other than the machine *MCMAX,* which requires least processing time.
$BMCT(BBMC1) = BMCT(BBMC1) + T(BBMC1,Q)$
$BMCT(MCMAX) = BMCT(MCMAX) - T(MCMAX,Q)$
$BNJ(BBMC1) = BNJ(BBMC1) + 1$
$BJS(BBMC1, BNJ(BBMC1)) = Q$
Removing the job Q from the machine MCMAX
$ZZ = 0$
FOR $Z = 1$ to $BNJ(MCMAX)$
If $BJS(MCMAX, Z) = Q$ then go to Step 5.17.1
$ZZ = ZZ + 1$
$BJS(MCMAX,ZZ) = BJS(MCMAX,Z)$
Step 5.17.1 NEXT Z
$BNJ(MCMAX) = BNJ(MCMAX) - 1$
Go to Step 5.19

Step 5.18: Update makespan and other results based on the allocation of the job Q to the machine other than the machine *MCMAX*, which has least completion time.
$BMCT(BBMC2) = BMCT(BBMC2) + T(BBMC2,Q)$
$BMCT(MCMAX) = BMCT(MCMAX) - T(MCMAX,Q)$
$BNJ(BBMC2) = BNJ(BBMC2) + 1$
$BJS[BBMC2, BNJ(BBMC2)] = Q$
Removing the job Q from the machine MCMAX
$ZZ = 0$
FOR $Z = 1$ to $NJ(MCMAX)$
If $BJS(MCMAX, Z) = Q$ then go to Step 5.18.1
$ZZ = ZZ + 1$
$BJS(MCMAX,ZZ) = BJS(MCMAX,Z)$
Step 5.18.1 NEXT Z
$BNJ(MCMAX) = BNJ(MCMAX) - 1$
Step 5.19: $X = X + 1$
Step 5.20: If $X \leq NJ(MCMAX)$ then go to Step 5.2; else go to Step 6.
***Step* 6:** If $JSI = n+1$ (*If JSI is unchanged*) then go to Step 8; else go to Step 7.
***Step* 7:** Update the results w.r.t. *BMS* as the best solution.
Transfer current machine completion times,
$BMCT(I)$ to $MCT(I)$ for all machines (I =1, 2, 3, …, m)
Transfer current number of jobs on each machine,
$BNJ((I)$ to $NJ(I)$ for all machines (I = 1, 2, 3, …, m)
Transfer current job sequences on machines,
$BJS(I,J)$ to $JS(I,J)$ for all I = 1, 2, 3, …, m and J = 1, 2, 3, …, $BNJ(I)$
Go to Step 3
***Step* 8:** Print the results.
Machine Completion Time, $MCT(I)$, I = 1, 2, 3,…., m.
Job assignments on the machines, $JS(I,J)$, I = 1, 2,, m & J = 1, 2, …., $NJ(I)$
Minimized makespan, *CMAX*.
***Step* 9:** Stop.

5. Comparison of Heuristic and Model

In this section, the solutions obtained through the proposed heuristic are compared with that obtained through the mathematical model using a complete factorial experiment. In the factorial experiment, two factors are assumed, viz., "*Method (M)*" and "*Problem Size (P)*". The number of levels for the method is 2, viz. *Heuristic* and *Model*. The number of levels for the Problem Size is 20 which are *2X5, 2X6, 2X7, 2X8, 2X9, 3X5, 3X6, 3X7, 3X8, 3X9, 4X5, 4X6, 4X7, 4X8, 4X9, 5X5, 5X6, 5X7, 5X8 and 5X9*. For each of the 40 experimental combinations, the data for three replications have been randomly generated. The process times (minutes) of the problems for the Replication 1, Replication 2 and Replication 3 are given in **Table 3**, **Table 4** and **Table 5**, respectively. STORM software is used to solve the model. The values of the makespan of the problems using the heuristic and the model are presented in **Table 6**. The formula to compute the percent deviation of the makespan of a problem given by the heuristic from that given by the model is as follows.

Percentage deviation of makespan

$$= \left\{ \frac{\text{Makespan using Heuristic} - \text{Makespan using Model}}{\text{Makespan using Model}} \right\} 100$$

The respective ANOVA model is as follows.

$y_{ijk} = \mu + M_i + P_j + MP_{ij} + e_{ijk}$

where, μ is the overall mean of the makespan

y_{ijk} is the response in terms of the makespan for the k^{th} replication of the i^{th} level of the factor M and the j^{th} level of the factor P.

M_i is the effect of the i^{th} level of the factor M on the response y_{ijk}

P_j is the effect of the j^{th} level of the factor P on the response y_{ijk}

MP_{ij} is the effect of the i^{th} level of the factor M and the j^{th} level of the factor P on the response y_{ijk}

e_{ijk} is the error in the k^{th} replication for the i^{th} level of the factor M and the j^{th} level of factor P.

The results of the corresponding ANOVA model are shown in **Table 7.** The hypotheses of this ANOVA model are listed as follows.

Factor "Method (*M*)*"*:

H_0: There is no significant difference between the methods (*Model and Heuristic*) in terms of the make span.

H_1: There is significant difference between the methods (*Model and Heuristic*) in terms of the makespan.

In the **Table 7**, the calculated F ratio for the factor "Me- thod (*M*)" is 0.09183275, which is less than the corresponding table F value (3.97) for (1,80) degrees of freedom at a significance level of 0.05. Hence, the null hypothesis is to be accepted. This means that *there is no significant difference between the methods (Heuristic and Model)* in terms of the makespan. So, the heuristic performs better for the assumed problems in terms of

Table 3. Process times of replication 1.

Problem Size	Machine	Job								
		1	2	3	4	5	6	7	8	9
2X5	1	24	4	5	6	12				
	2	18	2	19	1	3				
2X6	1	23	10	22	9	1	22			
	2	9	9	1	9	16	15			
2X7	1	5	4	14	4	22	8	21		
	2	9	10	21	5	19	6	11		
2X8	1	9	13	9	5	23	16	23	19	
	2	19	23	3	14	8	14	9	13	
2X9	1	19	18	16	18	12	20	2	17	22
	2	2	1	12	10	21	3	14	19	10
3X5	1	21	19	9	16	3				
	2	6	24	19	2	19				
	3	18	4	9	21	4				
3X6	1	3	23	1	24	17	11			
	2	2	7	14	15	9	7			
	3	4	2	14	6	22	8			
3X7	1	17	12	20	2	4	21	9		
	2	18	13	9	14	2	18	2		
	3	7	11	14	8	18	20	11		
3X8	1	23	12	11	4	23	7	18	10	
	2	13	19	21	20	4	14	12	2	
	3	24	12	1	15	6	12	12	22	
3X9	1	15	5	11	19	22	12	8	13	17
	2	6	24	6	22	14	7	17	18	1
	3	9	2	16	2	20	10	2	20	5
4X5	1	24	16	10	22	14				
	2	11	12	10	21	4				
	3	22	17	17	2	10				
	4	18	5	1	23	3				
4X6	1	5	9	4	2	10	17			
	2	17	7	1	10	23	12			
	3	16	1	23	4	1	2			
	4	18	7	7	23	15	12			
4X7	1	11	8	1	14	7	17	8		
	2	24	13	23	5	20	21	13		
	3	20	24	20	19	23	9	12		
	4	5	10	21	16	12	21	20		
4X8	1	22	9	16	15	12	22	5	16	
	2	2	2	16	7	19	14	7	15	
	3	8	13	18	14	24	5	24	9	
	4	4	4	19	3	13	5	6	21	
4X9	1	8	11	22	7	1	8	18	19	1
	2	8	19	10	22	20	20	7	11	18
	3	7	11	17	5	11	7	7	23	22
	4	9	22	21	7	2	8	6	6	22
5X5	1	10	4	23	7	10				
	2	20	1	24	9	5				
	3	8	2	15	2	2				
	4	19	16	18	1	11				
	5	11	2	5	20	1				
5X6	1	22	8	4	7	5	4			
	2	7	2	9	7	5	14			
	3	19	2	5	17	5	14			
	4	12	3	13	5	12	9			
	5	3	13	3	23	5	4			
5X7	1	17	1	23	5	6	10	5		
	2	15	10	8	7	19	2	5		
	3	1	10	20	3	24	22	23		
	4	18	22	7	7	3	12	8		
	5	5	13	3	3	14	6	12		
5X8	1	13	6	23	18	11	17	5	12	
	2	18	19	10	16	23	2	23	24	
	3	16	5	2	7	22	9	4	8	
	4	20	10	23	24	18	18	23	2	
	5	4	24	13	12	13	10	23	10	
5X9	1	15	24	11	16	14	6	5	19	7
	2	2	21	16	17	2	15	7	10	13
	3	19	16	22	12	3	6	10	5	21
	4	7	20	23	15	17	19	12	19	2
	5	5	15	3	22	6	9	17	2	15

Table 4. Process times of replication 2.

Problem Size	Machine	Job								
		1	2	3	4	5	6	7	8	9
2X5	1	3	9	5	7	14				
	2	24	23	15	11	5				
2X6	1	5	11	20	11	10	5			
	2	1	4	5	6	4	23			
2X7	1	20	16	5	1	23	13	6		
	2	5	1	6	4	18	24	15		
2X8	1	24	13	21	1	12	19	3	10	
	2	21	9	7	11	20	22	23	5	
2X9	1	17	18	6	4	5	10	19	5	24
	2	15	24	2	16	20	18	1	7	15
3X5	1	8	18	**8**	2	16				
	2	3	5	11	24	13				
	3	7	19	11	23	17				
3X6	1	21	14	11	14	7	12			
	2	8	12	6	10	8	6			
	3	16	12	19	13	10	17			
3X7	1	1	5	1	5	6	14	14		
	2	24	5	4	11	7	13	15		
	3	2	23	10	16	3	11	21		
3X8	1	15	17	15	1	2	2	5	7	
	2	19	8	1	15	15	15	5	8	
	3	4	6	15	5	7	17	20	8	
3X9	1	3	21	20	1	23	17	15	2	22
	2	5	12	2	8	16	22	6	24	16
	3	11	8	21	23	3	20	19	3	23
4X5	1	21	24	21	10	9				
	2	24	7	22	20	6				
	3	5	5	12	14	8				
	4	22	22	7	15	1				
4X6	1	17	4	1	7	3	11			
	2	19	23	12	24	17	4			
	3	1	18	7	23	3	18			
	4	5	18	6	22	3	18			
4X7	1	5	11	3	5	18	7	23		
	2	2	12	21	19	5	15	4		
	3	8	8	1	23	21	12	11		
	4	6	17	21	5	6	5	7		
4X8	1	22	4	15	15	22	15	6	19	
	2	2	1	3	16	5	15	12	20	
	3	2	5	16	17	5	24	1	5	
	4	16	13	24	9	5	2	1	6	
4X9	1	10	3	6	1	5	16	3	9	14
	2	16	5	5	11	24	18	11	7	1
	3	15	7	13	15	17	10	8	17	1
	4	21	22	20	19	12	11	19	1	8
5X5	1	14	13	4	11	9				
	2	22	6	7	20	20				
	3	23	4	19	13	2				
	4	18	5	23	21	21				
	5	17	8	16	18	15				
5X6	1	5	6	6	11	15	8			
	2	15	14	21	5	8	13			
	3	17	17	7	22	10	8			
	4	1	20	8	22	17	5			
	5	10	1	13	4	7	6			
5X7	1	9	10	17	3	20	2	19		
	2	18	23	2	22	9	5	1		
	3	20	11	15	4	19	20	24		
	4	24	2	2	17	7	22	12		
	5	14	20	2	15	24	11	22		
5X8	1	20	7	4	21	12	11	6	4	
	2	5	21	12	7	10	15	19	10	
	3	19	18	21	20	15	5	15	21	
	4	16	18	6	19	16	22	22	1	
	5	16	13	8	24	12	11	7	6	
5X9	1	13	1	14	4	11	12	21	20	14
	2	2	4	14	8	24	9	22	23	21
	3	4	8	5	8	22	20	1	6	5
	4	2	16	4	6	15	9	13	24	11
	5	13	15	14	13	17	22	13	9	5

Table 5. Process times of replication 3.

Problem Size	Machine	Job								
		1	2	3	4	5	6	7	8	9
2X5	1	12	11	9	12	23				
	2	3	24	6	11	9				
2X6	1	23	16	13	8	18	5			
	2	2	8	23	9	14	6			
2X7	1	8	20	21	24	19	20	18		
	2	15	6	20	24	13	17	18		
2X8	1	10	18	18	1	16	1	1	15	
	2	17	5	10	6	24	17	18	3	
2X9	1	19	2	15	19	12	8	19	22	16
	2	14	2	23	18	22	17	15	13	12
3X5	1	12	14	20	24	9				
	2	3	24	4	19	24				
	3	9	8	14	3	7				
3X6	1	14	5	15	2	24	20			
	2	13	14	2	23	17	15			
	3	9	4	7	4	1	6			
3X7	1	17	5	5	1	21	9	5		
	2	3	16	6	5	12	14	17		
	3	9	15	18	15	12	19	2		
3X8	1	14	20	8	20	8	4	21	5	
	2	18	23	19	20	23	20	22	9	
	3	18	13	21	10	3	24	20	24	
3X9	1	10	12	13	7	2	11	16	2	12
	2	4	4	23	4	20	5	23	5	24
	3	1	20	6	23	11	5	21	19	5
4X5	1	3	7	9	2	19				
	2	5	14	18	5	20				
	3	5	24	5	1	19				
	4	10	5	11	12	19				
4X6	1	11	20	19	4	15	10			
	2	18	5	6	14	16	15			
	3	19	24	8	21	8	22			
	4	7	6	9	5	8	8			
4X7	1	3	2	7	3	23	10	8		
	2	2	24	20	2	3	3	18		
	3	8	19	10	14	19	17	6		
	4	13	2	15	3	5	8	4		
4X8	1	18	24	13	4	24	8	10	8	
	2	16	9	12	17	21	22	13	24	
	3	6	11	23	14	5	18	19	7	
	4	10	4	17	23	12	24	8	11	
4X9	1	2	5	20	2	22	13	19	2	5
	2	13	3	3	14	19	18	18	22	22
	3	17	5	9	19	9	11	4	6	10
	4	8	22	23	23	4	11	10	8	15
5X5	1	7	14	5	9	23				
	2	7	10	22	13	7				
	3	9	10	23	7	18				
	4	5	22	11	5	3				
	5	12	24	7	9	18				
5X6	1	22	19	9	17	10	5			
	2	6	15	4	20	6	8			
	3	20	11	13	1	4	2			
	4	2	8	24	6	2	1			
	5	15	22	12	5	16	14			
5X7	1	13	9	16	13	5	17	8		
	2	15	13	5	2	20	19	2		
	3	12	12	15	16	5	14	19		
	4	23	4	9	10	5	5	3		
	5	11	7	15	9	8	5	23		
5X8	1	22	5	10	11	18	24	12	4	
	2	17	23	4	13	14	14	16	5	
	3	11	12	17	6	6	16	19	11	
	4	14	1	14	1	20	9	18	14	
	5	12	24	17	5	17	3	23	3	
5X9	1	2	20	11	5	24	1	13	23	10
	2	6	12	10	1	9	3	16	21	10
	3	1	7	3	3	7	8	24	7	12
	4	8	16	14	21	19	17	9	20	21
	5	19	12	15	4	2	3	6	23	24

Table 6. Makespan results of the problems.

Problem size		Approach	
		Heuristic	Model
	2X5	21	21
		17	17
		23	23
	2X6	25	25
		16	16
		26	26
	2X7	35	35
		25	25
		56	54
	2X8	43	43
		42	42
		28	28
	2X9	38	38
		40	40
		56	56
	3X5	9	9
		17	17
		11	11
	3X6	12	12
		21	21
		15	14
	3X7	22	21
		15	15
		17	14
	3X8	23	22
		10	10
		31	31
	3X9	24	23
		24	24
		18	18
	4X5	11	11
		10	10
		19	19
	4X6	7	7
		11	11
		12	12
	4X7	16	16
		10	10
		8	7
	4X8	17	16
		11	11
		18	18
	4X9	18	17
		14	14
		14	13
	5X5	8	8
		17	17
		10	10
	5X6	6	6
		8	7
		9	9
	5X7	6	6
		9	9
		11	11
	5X8	12	12
		13	13
		15	12
	5X9	18	16
		12	12
		13	12

providing near optimal solution.

Factor "Problem Size" (P):

H_0: There is no significant difference between the problems in terms of the makespan [*Problem size* (*P*)].

H_1: There is significant difference between the problems in terms of the makespan [*Problem size* (*P*)].

In the **Table 7**, the calculated F ratio for the factor, "*Problem Size* (*P*)" is 17.60259, which is more than the table F value (1.72) for (19, 80) degrees of freedom at a significance level of 0.05. Hence, the corresponding null hypothesis is to be rejected. This means that there is significant difference between the problems in terms of the makespan.

Interaction "MethodXProblem Size" (MXP):

H_0: There is no significant difference between the interaction terms: M_1P_1, M_1P_2, ……, M_1P_{20}, M_2P_1, M_2P_2, ……., M_2P_{20}, in terms of the makespan.

H_1: There is significant difference between at least one pair of the interaction terms: M_1P_1, M_1P_2, ……,M_1P_{20}, M_2P_1, M_2P_2,…….,M_2P_{20}, in terms of the makespan.

In the **Table 7**, the calculated F ratio for the interaction "Method x Problem Size" is 0.006703162, which is less than the table F value (1.72) for (19,80) degrees of freedom at a significance level of 0.05. Hence, the corresponding null hypothesis is to be accepted. This means that there is no significant difference between the interaction terms, M_1P_1, M_1P_2, ……, M_1P_{20}, M_2P_1, M_2P_2,……,M_2P_{20}, in terms of the makespan.

The mean percentage deviation of the makespan of the problems given by the heuristic from that given by the mathematical model is 2.336 %. As per the ANOVA experiment, there is significant difference between the problems in terms of the makespan and there is no significant difference between the proposed heuristic and the mathematical model as well as between the interaction terms (*Method* and *Problem Size*) in terms of the makespan. The proposed heuristic gives optimal makespan for 76.67 percent of the problems. These clearly indicate that the heuristic presented in this paper is an efficient heuristic in terms of providing very near optimal makespan.

6. Conclusions

Production scheduling is the key for the early completion of the components/ products. This paper considered the problem of scheduling *n* independent single operation jobs on *m* unrelated parallel machines such that the makespan is minimized. Since, the objective of minimizing the makespan of this problem comes under combinatorial category, obtaining the optimal solution using any of the exact procedures, viz., mathematical model, complete enumeration method, dynamic programming method, branch and bound method, etc., would lead to infeasible alternative for large size problems. Hence, in

Table 7. ANOVA results.

Source of variation	Degrees of freedom	Sum of squares	Mean Sum of Squares	F_{Ratio} (calculated)	F_{Table} (α=0.05)	Remark
Method(M)	1	3.332031	3.332031	0.09183275	3.97	Not Significant
Problem Size(P)	19	12135.05	638.6869	17.60259	1.72	Significant
Method X Problem Size(MXP)	19	4.621094	0.243215	0.006703162	1.72	Not significant
Error	80	2902.695	36.28369			
Total	119	15045.7				

this paper, an efficient heuristic for this problem is presented to obtain near optimal solution. The acceptability of any heuristic depends on its ability to provide the solution with better accuracy when compared to the optimal solution provided by a suitable mathematical model for a given problem. Hence, a model is also presented for this case. The accuracy can be tested only through a detailed experimentation. Hence, in this paper, a complete factorial experiment has been conducted with two factors, viz. *Method* (Proposed Heuristic and Model) and *Problem Size* (No. of Machines X No. of Jobs: 2X5, 2X6, ….,2X9, 3X5, 3X6, ….., 3X9, ..……., 5X5, 5X6, …5X9) with three replications for each experimental combination. From the ANOVA experiment, it is found that there is no significant difference between the methods, viz., "Proposed Heuristic" and "Model" in terms of providing the makespan for scheduling *n* independent jobs on *m* unrelated parallel machines. There is significant difference between different problems (2X5, 2X6, …., 2X9, ……., 5X5, 5X6, …5X9) in terms of providing the makepsan. There is no significant difference between interaction terms of "*Method"* and "*Problem Size"* in terms of providing the makespan. The mean percent deviation of the makespan obtained by the heuristic from that obtained by the mathematical model is 2.336 %. The percentage of the number of problems for which the optimality has been achieved is 76.67 %. From these results, is clear that the heuristic presented in paper is an efficient one in terms of providing optimal/near optimal makespan for scheduling *n* independent single operation jobs on *m* unrelated parallel machines, which is a significant contribution. Since, the proposed heuristic gives results on par with that of the mathematical model, with no statistical significance for the difference between the results, the proposed heuristic is an effective heuristic. The problems of the data set presented can serve as benchmark problems for future researches.

7. Acknowledgment

We thank the anonymous referees for their constructive suggestions, which helped us to improve the paper.

8. References

[1] R. Panneerselvam, "Production and operations management (Second Edition)," PHI Learning Private Limited, New Delhi, 2005.

[2] S. L. Van De Velde, "Duality-based algorithms for scheduling unrelated parallel machines," ORSA Journal of Computing, Vol. 5, pp. 182–205, 1993.

[3] C. A. Glass, C. N. Potts, and P. Shade, "Unrelated parallel machine scheduling using local search," Mathematical and Computer Modelling, Vol. 20, pp. 41–52, 1994.

[4] Francis Sourd "Scheduling tasks on unrelated machines: Large neighbourhood improvement procedures," Journal of Heuristics, Vol. 7, pp. 519–531, 2001.

[5] E. Mokotoff and P. Chretienne, "A cutting plane algorithm for the unrelated parallel machine scheduling problem," European Journal of Operational Research, Vol. 141, pp. 515–525, 2002.

[6] E. Mokotoff and J. L. Jimeno, "Heuristics based on partial enumeration for the unrelated parallel processor scheduling problem," Annals of Operations Research, Vol. 117, pp. 133–150, 2002.

[7] A. M. A. Hariri and C. N. Potts, "Heuristics for scheduling unrelated parallel machines," Computers and Operations Research, Vol. 18, pp. 323–331, 1991.

[8] N. Piersman and W. Van Dijk, "A local search heuristic for unrelated parallel machine scheduling with efficient neighbourhood search," Mathematical and Computer Modelling, Vol. 24, pp. 11–19, 1996.

[9] M. Ghirardi and C. N. Potts, "Makespan minimization for scheduling unrelated parallel machines: A recovering beam search approach," European Journal of Operational Research, Vol. 165, pp. 457–467, 2005.

[10] B.Monien and A. Woclaw "Scheduling unrelated parallel machines computational results," Experimental Algorithms, Springer, Berlin/Heidelberg, Vol. 4007, pp. 195–206, 2006.

[11] M. Gairing, B. Monien, and A. Woclaw, "A faster combinatorial approximation algorithm for scheduling unrelated machines," Theoretical Computer Science, Vol. 380, pp. 87–99, 2007.

