Journal of the Operations Research Society of Japan Vol. 34, No. 1, March 1991

## A FLOWSHOP SCHEDULING ALGORITHM TO MINIMIZE TOTAL FLOWTIME

Chandrasekharan Rajendran Dipak Chaudhuri Indian Institute of Technology

(Received February 6, 1989; Revised June 1, 1990)

Abstract In this paper a heuristic algorithm is presented for scheduling in flowshops with the objective of minimizing the sum of completion times or total flowtime of jobs. The algorithm is developed by considering the lower bound on completion times of jobs for various stages of flowshop. Since the proposed lower bound will also hold for the flowshop with no job-waiting constraint at some or all stages of processing, the heuristic can as well be applied to such flowshop problems. The performance of the heuristic algorithm in all types of flowshop problems has been evaluated. It is found to be consistently good and superior to the existing flowshop scheduling algorithms.

## 1. Introduction

The study of flowshop scheduling has attracted considerable interest among researchers over the last three decades. Ignall and Schrage [8] and Lominicki [10] have developed the branch-and-bound algorithms for scheduling to minimize makespan. Heuristic algorithms have been proposed by Campbell, Dudek and Smith [3], Dannenbring [5], Stinson and Smith [15] and Nawaz and Enscore [12]. Ignall and Schrage [8] and Bansal [1] have developed branch-and-bound algorithms for solving the flowshop scheduling problem with the objective of minimizing total flowtime of jobs. Heuristics to achieve this objective have been proposed by Gupta [7] based on weaker form of Dudek-Teuton dominance relations and by Miyazaki, Nishiyama and Hashimoto [11] based on the adjacent pairwise interchange of jobs. Szwarc [16] has analysed some aspects of the flowshop scheduling problems with the total flowtime criterion.

A constraint that is sometimes encountered in flowshops is that a job once started should be processed through all the stages without waiting. If necessary, the start of the job is delayed so that no job awaits processing at any stage except before the first. Wismer [18] and Reddi and Ramamoorthy [14] have solved the constrained flowshop problem with the objective of minimizing the makespan. With this objective, Bonney and Gundry [2] and King and Spachis [9] have developed heuristics. Van Deman and Baker [17] have developed a branch-and-bound algorithm to minimize total flowtime.

The experimental study of a comprehensive performance measure in the flowshop scheduling problem has been conducted by Gupta and Dudek [6]. The study has revealed that scheduling to minimize total flowtime would result in minimizing total scheduling cost more than that obtained by scheduling to minimize makespan. The industrial survey undertaken by Panwalkar, Dudek and Smith [13] has confirmed that minimization of total flowtime is often found to be an important objective in real life situations.

In this paper, a heuristic algorithm for scheduling in flowshops with the objective of minimizing the total flowtime of jobs is presented. The necessity for a heuristic arises due to the NP-completeness of the most of the scheduling problems. The proposed heuristic can be applied to the unconstrained as well as the constrained flowshop. Sometimes the constraint of no job-waiting may hold good for certain stages of the flowshop only. For example, though the metal rolling operation is done without any delay while the metal is hot, other operations like acid pickling, surface finish processes etc. can be undertaken with the metal waiting for such operations. The proposed heuristic can as well be applied to such hybrid flowshops for which the no job-waiting constraint holds good for some stages of processing only. The performance of the heuristic has been evaluated by applying it to a large number of general, constrained and hybrid flowshop problems.

## 2. Terminology

Let

- n be the number of jobs to be scheduled.
- m the number of stages in the flowshop.
- $t_{ij}$  the processing time of the *i*-th job at the *j*-th stage.
- $\sigma$  the available partial schedule.
- $\pi$  the set of unscheduled jobs.
- a, b jobs in the set  $\pi$ .
- $q(\sigma, j)$  the completion time of the partial schedule at stage j.
- $LB(\sigma a, j)$  the lower bound on the completion time of job a at stage j when the job follows  $\sigma$ .  $S_{\sigma ak}$  the sum of lower bound on completion time of job a for the last k stages,  $k = m, m-1, m-2, \dots, 2, 1$ .

## 3. Development of the Algorithm

In order to minimize total flowtime, we need to build up the schedule in a smooth and gradually increasing manner, i.e., the jobs with smaller processing times are to be scheduled ahead of the jobs with larger processing times (Conway et al. [4] and Szwarc [16]). Keeping this in mind, we develop heuristic preference relations considering the lower bound on the sum of completion time of a job for various stages.

When the flowshop is empty, the completion time of job a at stage p is given by  $[\sum_{j=1}^{p} t_{aj}]$ . When a partial schedule  $\sigma$  is available, a weak lower bound on the completion time of job a for stage p is given by

$$LB(\sigma a, p) = q(\sigma, 1) + \sum_{j=1}^{p} t_{aj}$$

$$\tag{1}$$

Therefore the sum of lower bound on completion time of job a for m stages is

$$S_{\sigma am} = \sum_{p=1}^{m} LB(\sigma a, p) \tag{2}$$

$$= mq(\sigma, 1) + \sum_{j=1}^{m} [(m - j + 1)t_{aj}]$$
(3)

The sum of lower bound on completion time of job a for the last k stages can be obtained recursively as

$$S_{\sigma ak} = S_{\sigma a,k+1} - q(\sigma,1) - \sum_{j=1}^{m-k} t_{aj}, \quad k = m-1, m-2, \cdots, 2, 1$$
(4)

The following heuristic preference relation is proposed:

## *Copyright* © *by ORSJ. Unauthorized reproduction of this article is prohibited.*

 $\sigma a$  is preferred to  $\sigma b$  if

$$S_{\sigma ak} \le S_{\sigma bk} \tag{5}$$

Let k = m and  $T_i = \sum_{j=1}^{m} [(m - j + 1)t_{ij}]$ . Using relation 5 and Eq. 3, we say that

 $\sigma a$  is preferred to  $\sigma b$ 

 $\mathbf{i}\mathbf{f}$ 

$$mq(\sigma,1) + T_a \le mq(\sigma,1) + T_b \tag{6}$$

or if

$$T_a \le T_b \tag{7}$$

This deduction leads to the generation of a schedule obtained by rank-ordering the jobs in the ascending order of value of  $T_i$ . Likewise by letting  $k = m - 1, m - 2, \dots, 2, 1$  in Eq. 4 and using the relation 5, we obtain at most (m - 1) additional schedules.

The step by step procedure of the heuristic algorithm is as follows:

Step 1: Set p = 1. Step 2: Compute

$$T_i = \sum_{j=1}^{m} [(m-j+1)t_{ij}], \quad i = 1, 2, \cdots, n$$

- Step 3: Form an array of jobs by arranging them in the ascending order of value of  $T_i$ . Ties can be broken arbitrarily.
- Step 4: Take the array of jobs so formed as a schedule and evaluate it with respect to total flowtime of jobs.
- Step 5: If p = m, go to Step 8.

Step 6: Update

$$T_i = T_i - \sum_{j=1}^p t_{ij}, \quad i = 1, 2, \cdots, n$$

- Step 7: Let p = p + 1 and return to step 3.
- Step 8: Choose the schedule, out of the at most m distinct schedules thus generated, that minimizes the total flowtime the most. STOP.

It can be seen that the lower bound on completion times of a job for various stages will also hold good for the constrained as well for the hybrid flowshops. Hence the heuristic algorithm can also be applied to such flowshop scheduling problems thereby having a wide spectrum of application.

# 4. A Numerical Illustration

Consider a flowshop with four stages. There are five jobs to be scheduled without any constraint on waiting (i.e., general flowshop) and their processing times at various stages are as follows:

		Stages					
			1	2	3	4	
	1	:	7	25	 10	12	•
	<b>2</b>	÷	13	18	21	10	
Jobs	3	÷	3	25	3	18	
	4	÷	11	9	1	6	
	<b>5</b>	:	6	4	25	11	

The objective is to minimize total flowtime of jobs.

## 1-st schedule:

$$T_{1} = \sum_{j=1}^{4} [(4 - j + 1)t_{1j}]$$
  
= 135  
$$T_{2} = 158$$
  
$$T_{3} = 111$$
  
$$T_{4} = 79$$

Similarly

Rank ordering the jobs in the ascending order of value of  $T_i$  yields the schedule {45312}. The sum of completion times of jobs in this schedule is 384. Set p = 1.

 $T_5 = 97$ 

## 2-nd schedule:

Update

Similarly

$$T_{1} = T_{1} - \sum_{j=1}^{p} t_{1j}$$
  
= 128  
$$T_{2} = 145$$
  
$$T_{3} = 108$$
  
$$T_{4} = 68$$
  
$$T_{5} = 91$$

Rank ordering the jobs in the ascending order of value of  $T_i$  yields the schedule {45312}. This schedule is same as the first schedule. Set p = 2.

Similarly by repeating the procedure of updating  $T_i$ , rank ordering the jobs, generating a schedule and evaluating it, the schedules obtained are as follows:

Schedule	$T_1$	$T_2$	$T_3$	$T_4$	$T_5$	Schedule	Total
generated						generated	flowtime
3	96	114	80	48	81	$\{43512\}$	398
4	54	62	49	27	46	$\{45312\}$	384

The schedule  $\{45312\}$  is selected as it yields the least value of total flowtime of 384. The schedule  $\{45123\}$  yields the optimal total flowtime of 373. The percentage error of the heuristic solution is

$$= (384 - 373) * 100/373$$
$$= 2.95\%$$

*Copyright* © *by ORSJ. Unauthorized reproduction of this article is prohibited.* 

Schedule generated	$T_1$	$T_2$	$T_3$	$T_4$	$T_5$	Schedule generated	Total flowtime	
1	135	158	111	79	97	{45312}	400	
2	128	145	108	68	91	$\{45312\}$	400	
3	96	114	80	48	81	$\{43512\}$	403	

27

46

 $\{45312\}$ 

400

Suppose we introduce the constraint of no job-waiting in the flowshop. The computations involved in solving this problem are as follows:

The schedule  $\{45312\}$  is selected as it yields the least value of total flowtime of 400. The schedule  $\{45123\}$  has been found to yield the optimum total flowtime of 381. The percentage error of heuristic solution is

49

= (400 - 381) \* 100/381= 4.99%

## 5. Computational Experience

4

54

62

The applicability of a heuristic algorithm depends on its performance, i.e., the quality of the solution for a given computational time. Hence a large number of experiments has been conducted to investigate the effectiveness of the proposed algorithm. The experimentation has been carried out in three phases: first the general flowshop problem is considered and the performance of the proposed algorithm is compared with Gupta's MINIT algorithm and Miyazaki's algorithm for 1170 problems; next the constrained flowshop problem is considered and the performance of the proposed heuristic is evaluated for the same 1170 problems. Finally about 3200 hybrid flowshop problems are considered having the constraint of no job-waiting applicable to 25%, 50% and 75% of the flowshop. All the algorithms have been programmed in FORTRAN IV for an IPL 4443 System using WATFIV Compiler. The processing times of the jobs have randomly been generated from a rectangular distribution ranging from 1 to 99.

## (a) General flowshop problem:

In order to carry out the experimental investigations, 1170 problems having 5, 6, 7, 8, 9, 10, 12, 18 and 24 jobs with the number of stages varying from 5 to 25 have been generated and solved by all the three algorithms. In order to evaluate the effectiveness of the algorithms, the following criteria are used:

(a) Absolute percentage error =  $(F - F^*) * 100/F^*$  where

F = Total flowtime of the schedule obtained from a heuristic solution and  $F^*$  = Optimal total flowtime obtained from Bansal's branch-and-bound algorithm [1].

(b) Standard deviation of percentage error.

The heuristic solutions are compared with the optimal values for 720 problems having 5, 6, 7, 8, 9 and 10 jobs. For problems having larger number of jobs, the relative percentage error is determined as follows:

Relative percentage error = (F - F') \* 100/F'

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

where F' denotes the best heuristic solution. The results of evaluation of absolute and relative error for various algorithms are tabulated in Tables 1 and 2.

As it can be seen from the tables, the proposed algorithm gives consistently better results than both the MINIT and Miyazaki's algorithms. Table 3 contains the execution time of algorithms for 750 flowshop problems. The proposed algorithm requires less computational effort than Miyazaki's algorithm. The MINIT algorithm checks for the choice of first two jobs in the schedule and as the value of n increases, the computational effort also increases. Hence the proposed algorithm requires less computational effort than MINIT algorithm for larger size problems which are normally encountered in real-life situations.

## (b) Constrained flowshop problem:

Considering 5, 6, 7, 8, 9 and 10 jobs with the number of stages varying from 5 to 25, 720 problems generated for the general flowshop are then solved with the constraint of no job-waiting. To have a bench-mark of comparison, we have devised the following RANDOM selection rule: (a) Select randomly the job to be appended from the set of unscheduled jobs and hence obtain a randomly generated schedule; (b) Generate as many such schedules as the maximum of the number of jobs and the number of machines (i.e.,  $\max[n;m]$ ) — it can be seen that the number of schedules varies and increases with the problem size and (c) Choose that schedule which yields the least flowtime.

It can be seen that the RANDOM selection rule generates always more number or equal number of schedules as generated by the proposed heuristic algorithm since the heuristic generates at most m schedules. Hence the RANDOM selection rule devised for this study emerges as an effective rule of comparison with the proposed heuristic.

The heuristic solutions and the solutions obtained from the bench mark, RANDOM rule, are compared with the optimal solutions obtained by the branch-and-bound algorithm of Van Deman and Baker [17]. The results are tabulated in Table 4. As it can be seen, the proposed algorithm also yields fairly good solutions for the flowshop problems with the no job-waiting condition.

In addition, we have solved the 450 problems with 12, 18 and 24 jobs that have been generated for the general flowshop and computed the relative error of the solutions for the constrained flowshop given by the proposed algorithm and the RANDOM selection rule. The results are tabulated in Table 5. An inspection of the Table reveals that the proposed heuristic yields good solutions even for the large-sized constrained flowshop problems.

## (c) Hybrid flowshop problem:

The problems generated for the general flowshop having 5, 6, 7 and 8 jobs with the number of stages varying from 5 to 25 are solved as hybrid flowshop problems having the constraint of no job-waiting applicable to 25%, 50% and 75% of the stages of the flowshop. The heuristic solutions and the solutions obtained from the bench mark, RANDOM rule, are then compared with the optimal solutions which have been obtained by complete enumeration. The results are given in Tables 6, 7 and 8. It is found that the performance of the heuristic in solving the hybrid flowshop problems is good.

In addition, we have solved all the 450 large-sized problems that have been generated for the general flowshop in each class of hybrid flowshop (viz., constraint for 25%, 50% and 75% of the flowshop) using the proposed heuristic and the RANDOM selection rule. The results of relative evaluation for the 1350 large-sized hybrid flowshop problems are tabulated in Tables 9, 10 and 11. The inspection of these tables reaffirms our earlier findings that the heuristic algorithm yields good solution even for the hybrid flowshop problem.

The proposed algorithm fares better in the unconstrained flowshop problems than in the

constrained problems. In the constrained flowshop problem, the waiting of a job occurs at the first stage only. While developing more efficient heuristics for the constrained flowshop problems, one has to give due consideration to such job-waiting.

## 6. Conclusion

A heuristic algorithm has been proposed for scheduling in flowshop with the objective of minimizing the sum of completion time of jobs. The algorithm is so developed that it can also be applied to the constrained and the hybrid flowshops. The algorithm has been found to be superior to the existing algorithms for the general flowshop and found to yield good solutions to the constrained and hybrid flowshops. Thus the heuristic finds a wide spectrum of application in the real-life flowshops.

Number	Number	Number	Proposed	Miyazaki's	Gupta's
of	of	of	$\operatorname{Algorithm}$	Algorithm	Algorithm
jobs	stages	problems			
			ABSOLUTE	PERCENTAGE	ERROR
			Mean Std. dev.	Mean Std. dev.	Mean Std. dev.
	5	30	$1.8326 \ 1.9836$	$6.8120 \ 7.0165$	$4.8154 \ 5.2521$
	10	30	$1.1210 \ 1.7379$	$3.1210 \ 2.7978$	$2.8020 \ 2.4201$
5	15	30	$0.6003 \ 0.9666$	$3.7693 \ 3.3834$	$1.9930 \ 2.4125$
	20	30	$0.7270 \ 0.9123$	3.3393 3.9006	$1.7956 \ 1.7216$
	25	30	$0.6683 \ 0.9623$	3.3957 3.0206	$1.3407 \ 1.6671$
	5	30	$1.9390 \ 2.2432$	6.2253 $6.9049$	5.0867 $5.1835$
	10	30	$1.1253 \ 1.1218$	4.4050 3.2156	$3.4340 \ 2.7441$
6	15	30	$1.1703 \ 1.3237$	5.0730 $3.6809$	$3.3467 \ 2.2596$
	20	30	$1.1487 \ 1.1595$	$4.4147 \ 2.8324$	$1.8497 \ 1.8284$
	25	30	$1.0358 \ 1.4412$	$4.2785 \ 2.8611$	$1.7895 \ 1.4088$
	5	30	$2.7863 \ 2.4821$	$6.5840 \ 3.7942$	5.4130 $3.5536$
	10	30	$1.7813 \ 1.7419$	5.1403 $3.9751$	$4.8353 \ 2.7598$
7	15	30	$1.8137 \ 1.1216$	$5.6490 \ 3.1475$	$3.3643 \ 2.5553$
	20	30	$1.3193 \ 1.3741$	$4.3963 \ 2.7226$	$3.0460 \ 2.0103$
	25	30	$1.1437 \ 1.0718$	$3.7533 \ 2.5648$	$2.5367 \ 2.4101$
	5	30	$3.7590 \ 2.4985$	6.3249 4.5179	$6.1061 \ 3.9030$
	10	30	2.5876 $2.1274$	6.5377 $4.8215$	5.7137 $3.6475$
8	15	30	$2.0116 \ 1.8771$	5.5186 $3.0188$	$4.5370 \ 3.2741$
	20	30	$2.6259 \ 1.6132$	$4.9685 \ 3.1154$	$4.9998 \ 2.3862$
	25	30	$1.9502 \ 1.4988$	$4.1596 \ 2.6880$	$3.8685 \ 2.7863$
	5	30	$4.8108\ 2.8337$	$6.9497 \ 4.6512$	7.4989 $3.5015$
9	10	30	$3.3856 \ 2.2050$	7.8859 $3.6683$	5.5834 $3.8456$
	15	30	$3.1585 \ 1.9915$	$6.5872 \ 2.6253$	$5.5289\ 2.7398$
10	5	30	$5.0839\ 2.4551$	$9.4517 \ 4.0636$	7.8032 $3.9836$

# Table 1. Computational results for general flowshop

Number	Number	Number	Proposed	Miyazaki's	Gupta's
of	of	of	Algorithm	Algorithm	$\operatorname{Algorithm}$
jobs	stages	problems			
			RELATIVE	PERCENTAGE	ERROR
			Mean Std. dev.	Mean Std. dev.	Mean Std. dev.
	5	30	0.4960 1.0083	3.5676 $3.5446$	3.5943 $3.8034$
	10	30	$0.3587 \ 0.8374$	$3.0613 \ 2.6698$	$3.6907 \ 3.5106$
12	15	30	$0.2210 \ 0.7361$	$3.1607 \ 2.0245$	$2.4147 \ 2.2832$
	20	30	$0.3323 \ 0.6566$	$3.2913 \ 2.4772$	3.2420 $3.0234$
	25	30	0.1946 0.6133	$2.2803 \ 2.2881$	$2.6860 \ 1.8494$
	5	30	$0.6553 \ 1.2959$	3.3327 3.5681	$4.0490 \ 2.9262$
	10	30	0.3583 1.3539	$3.4367 \ 2.4423$	$4.7597 \ 3.1708$
18	15	30	0.1390 0.3256	$2.8357 \ 1.7914$	$4.8930 \ 3.2728$
	20	30	$0.1917 \ 0.5361$	$3.1697 \ 2.5037$	$4.1173 \ 2.9055$
	25	30	$0.1053 \ 0.3597$	3.2810 1.8626	3.7360 2.1315
	5	30	0 7543 1 6032	2 3893 2 9534	4 8830 4 0094
	10	30	0.2517 0.6149	2.0033 2.3034	4 7117 4 0004
94	15	20	0.2017 0.0149	2.4410 2.4029	4.(11) 4.0094 E E700 9 8000
24	15	30	0.2007 0.7107	2.8203 2.0830	5.5790 3.8209
	20	30	0.2037 0.7763	2.5827 $2.1653$	4.2197 2.7553
	25	30	$0.1533 \ 0.5412$	$2.4397 \ 1.6259$	4.7490 2.1767

# Table 2.Computational results for the general flowshopwith larger number of jobs.

٥

Number	Number	Number	Proposed	Miyazaki's	Gupta's
of	of	of	Algorithm	Algorithm	Algorithm
jobs	stages	problems			
	5	30	3.02	6.65	3.54
	10	30	7.76	27.18	6.67
6	15	30	13.42	66.78	9.84
	20	30	22.18	<b>130</b> .91	13.15
	25	30	33.37	224.15	16.59
	5	30	3.62	8.28	4.70
	10	30	9.06	33.21	8.93
7	15	30	15.95	81.32	13.29
	20	30	26.36	156.91	17.81
	25	30	37.68	265.65	23.83
	5	30	6.37	16.81	12.31
	10	30	16.36	58.55	23.79
12	15	30	31.18	165.61	34.99
	20	30	45.21	317.70	47.81
	25	30	68.80	526.74	60.93
	5	30	10.29	29.97	26.39
	10	30	26.03	120.06	49.59
18	15	30	48.91	292.56	74.01
	20	30	71.52	528.56	99.84
	25	30	104.82	918.07	125.24
	5	30	14.71	51.82	45.64
	10	30	37.15	191.67	89.93
24	15	30	66.91	<b>433.61</b>	132.11
	20	30	97.80	811.86	173.86
	25	30	147.76	1350.38	233.46

Table 3. Total execution time of algorithms (in seconds)

of	c		1 0	
••	ot	of	ABSOLUTE	PERCENTAGE ERROR
jobs	stages	problems	Mean Std.	Mean Std.
			dev.	dev.
	5	30	$2.2617 \ 2.1034$	$9.1953 \ 7.0254$
	10	30	$1.2247 \ 1.4565$	$4.3957 \ 3.1543$
5	15	30	$0.5917 \ 0.9262$	$3.5710\ 2.1897$
	20	30	0.8980 1.1386	$3.1734 \ 2.3337$
	25	30	0.7660 1.0656	2.8990 1.6919
	5	30	2.6576 2.8653	$12.6973 \ 8.1547$
	10	30	$1.3603 \ 1.4542$	$6.0340 \ 3.1969$
6	15	30	$1.2573 \ 1.1022$	5.6187 2.9603
	20	30	$0.8873 \ 1.1772$	$5.1333\ 3.0871$
	25	30	$1.2018 \ 1.2457$	$3.2643 \ 2.2275$
	5	30	3.3729 2.9986	$14.8017 \ 8.2656$
	10	30	2.3893 2.4012	10.1563 4.3111
7	15	30	1.8933 1.5232	8.2437 3.7126
	20	30	$1.5329 \ 1.3371$	7.2747 4.3727
	25	30	1.4163 1.2775	5.4417 3.0927
	5	30	4.4564 2.8359	14 7488 5 4378
	10	30	3.5732 3.0665	12 5961 5 8645
8	15	30	2.6442.2.2731	9 5230 4 2361
0	20	30	2 2081 1 5101	7 5481 2 3218
	25 25	30	2.2550 1.8856	6.5113 3.4439
	F	20	5 9147 9 4999	15 1990 6 1965
0	10	30	2 0566 2 0081	
9	15	30	3.9300 J.0001	
	19	30	2.1955 1.9026	11.8538 4.5009
10	5	30	6.5213 3.0116	21.8835 9.7479

Table 4. Computational results for the constrained flowshop

## Flowshop Scheduling Algorithm

Numbe	rNumber	r Number	Proposed	algorithm	n Best	RANDOM schedule
of	of	of	RELATIV	Έ	PERCENTAG	E ERROR
jobs	stages	problems	Mean	Std.	Mean	Std.
				dev.		dev.
	5	30	0	0	13.3707	6.4058
	10	30	0	0	10.3763	6.0254
12	15	30	0.1533	0.8255	10.3583	4.3925
	20	30	0	0	9.1493	4.4348
	25	30	0	0	8.1660	5.1915
	5	30	0	0	14.5173	6.3280
	10	30	0	0	13.3937	6.6634
18	15	30	0.1227	0.6606	13.5157	4.1882
	20	30	0	0	11.7600	4.1854
	25	30	0	0	11.1087	4.6670
	5	30	0	0	17.5829	5.4748
	10	30	0	0	14.5297	4.4956
24	15	30	0	0	14.1411	6.0684
	20	30	0	0	15.3150	3.9438
	25	30	0	0	12.9103	4.4179

# Table 5. Computational results for the constrained flowshop with larger number of jobs

Number	Number	Number	Proposed algorithm	Best RANDOM schedule
$\mathbf{of}$	of	of	ABSOLUTE	PERCENTAGE ERROR
jobs	stages	problems	Mean Std.	Mean Std.
			dev.	dev.
	5	30	1.8160 1.9149	7.8897 6.2520
	10	30	1.2390 1.7777	4.4404 3.9185
<b>5</b>	15	30	0.8703 1.1547	2.4852 2.3539
	20	30	$1.1337 \ 1.2377$	$2.6889\ 1.8515$
	25	30	$1.1027 \ 1.2321$	$1.9344 \ 1.3897$
	5	30	1.9220 2.3256	11.0547 7.3250
	10	30	1.0297 1.1118	4.9751 3.4121
6	15	30	$1.2877 \ 1.3267$	$3.7546 \ 2.7589$
	20	30	$1.1957 \ 1.2503$	3.5069 1.8843
	25	30	$0.8520 \ 1.3742$	2.3401 1.8781
	5	30	2.6557 2.3549	$11.9968 \ 6.7525$
	10	30	1.6010 1.7079	$6.4805 \ 2.6452$
7	15	30	$1.9097 \ 1.4869$	5.8792 3.1049
	20	30	1.2445 1.5029	5.0276 3.1831
	25	30	$1.2175 \ 1.1067$	4.3004 2.2878
	5	30	3.3634 2.3983	15.5634 6.5248
	10	30	2.1645 1.2269	8.8544 3.5719
8	15	30	2.3454 1.2964	6.4828 $4.5323$
	20	30	2.2873 1.8238	$4.4558\ 2.1955$
	25	30	$1.7182 \ 1.2821$	$4.5613 \ 2.0315$

Table 6. Computational results for the hybrid flowshopwith the constraint for 25% of the flowshop.

## Flowshop Scheduling Algorithm

,

Number	Number	Number	Proposed algorithm	Best RANDOM schedule
of	of	of	ABSOLUTE	PERCENTAGE ERROR
jobs	stages	problems	Mean Std.	Mean Std.
			dev.	dev.
	5	30	$1.8703 \ 1.9182$	8.1069 6.6479
	10	30	2.1756 2.2096	5.4502 $4.3118$
5	15	30	1.0932 1.1268	2.8676 2.2971
	20	30	1.7830 1.6047	$3.5080 \ 2.0747$
	25	30	$1.6567 \ 1.5065$	$2.4054 \ 1.4923$
	5	30	2.0070 $2.5048$	$11.0438\ 7.2173$
	10	30	$1.3587 \ 1.4482$	$5.1707 \ 2.9789$
6	15	30	1.0749 1.4335	4.2969 2.6184
	20	30	$1.0552 \ 1.3539$	$3.9187 \ 2.3005$
	25	30	$0.9727 \ 1.4638$	2.6379 3.0670
	<b>5</b>	30	2.6120 2.5616	$13.0798\ 7.7124$
	10	30	2.1930 2.1311	8.0743 3.0885
7	15	30	1.8050 1.7909	$6.5160 \ 3.4216$
	20	30	$1.3773 \ 1.7431$	5.5359 3.7053
	25	30	$1.4295 \ 1.1759$	$4.5406 \ 2.5156$
	5	30	$3.4873 \ 2.2843$	$16.7926 \ 6.9447$
	10	30	$2.9917 \ 1.8760$	$10.4321 \ 4.7910$
8	15	30	2.5958 $2.0576$	7.3002 4.9057
	20	30	2.8978 1.9719	5.9076 2.6965
	25	30	$1.7086 \ 1.5560$	5.8104 2.6191

Table 7. Computational results for the hybrid flowshop with the constraint for 50% of the flowshop

Number	Number	Number	Proposed algorithm	Best RANDOM schedule
of	of	of	ABSOLUTE	PERCENTAGE ERROR
jobs	stages	problems	Mean Std.	Mean Std.
			dev.	dev.
	5	30	1.7630 1.8906	8.1855 6.4447
	10	30	2.7160 $2.6393$	6.0501 $4.3592$
5	15	30	$1.6593 \ 1.4701$	$3.3830 \ 2.4376$
	20	30	2.2417 2.0498	4.0871 $2.4991$
	25	30	2.1230 1.5142	3.2186 1.8219
	5	30	$1.7597 \ 2.2361$	$12.1013 \ 7.5675$
	10	30	1.4263 1.5074	5.3966 2.9975
6	15	30	1.3737 1.2424	4 6629 2 6137
	20	30	1.1803 1.5256	4.4631 2.6791
	25	30	1.0450 1.4601	3.1473 2.1365
	5	30	2 7637 2 6017	14 3130 7 3018
	10	30	2.1031 2.0011	0 1282 4 0651
7	10	20	2.2913 2.1293	7 6846 4 1449
1	10	30 20	1.9097 1.7900	6 2662 2 0046
	20	30 20		0.3003 3.9940
	25	30	1.4/15 1.411/	4.9030 2.9741
	5	30	3.8282 2.8106	18.7745 7.3830
	10	30	$2.8246 \ 2.1157$	11.5015 $4.9378$
8	15	30	$2.6487 \ 1.9494$	8.1695 5.9561
	20	30	2.8455 $2.1874$	7.1854 $3.4546$
	25	30	1.9195 1.7451	6.0631 2.0871

Table 8. Computational results for the hybrid flowshop with the constraint for 75% of the flowshop.

## Flowshop Scheduling Algorithm

Number	Numbe	r Number	Proposed	algorithm	n Best	RANDOM schedule
of	of	of	RELATIVE	E	PERCENTAGE	ERROR
jobs	stages	problems	Mean	Std.	Mean	Std.
				dev.		dev.
	5	30	0	0	10.4203	5.1485
	10	30	0.0333	0.1795	7.5640	4.5517
12	15	30	0	0	5.9919	3.0365
	20	30	0.0865	0.4659	5.2734	3.0164
	25	30	0.0764	0.3162	4.6473	3.0978
	5	30	0.0270	0.1454	10.8687	5.6370
	10	30	0.0443	0.2387	8.3840	4.6497
18	15	30	0.0263	0.1294	7.3190	3.6671
	20	30	0	0	7.0773	3.4221
	25	30	0.0048	0.0260	5.2020	2.5022
	5	30	0	0	12.2100	4.6398
	10	30	0	0	9.3153	3.8104
24	15	30	0	0	7.7627	3.8410
	20	30	0	0	6.9120	2.5957
	25	30	0	0	5.3377	2.9998

# Table 9. Computational results for the hybrid flowshop having larger number of jobs with the constraint for 25% of the flowshop.

Number	Numbe	r Number	Proposed	algorithm	Best	RANDOM schedule
of	of	of	RELATIVE		PERCENTAGE	ERROR
jobs	stages	problems	Mean	Std.	Mean	Std.
				dev.		dev.
	5	30	0	0	11.8297	5.9372
	10	30	0	0	9.1013	6.0408
12	15	30	0	0	8.2854	4.6640
	20	30	0	0	6.8007	3.7009
	25	30	0.0562	0.2394	5.8437	3.6689
	5	30	0	0	12.6234	6.0827
	10	30	0	0	10.3270	5.1587
18	15	30	0.1312	0.5297	9.7940	4.6077
	20	30	0.0232	0.1251	8.8663	4.5150
	25	30	0	0	7.4347	3.4137
	5	30	0	0	14.7157	5.5639
	10	<b>3</b> 0	0	0	12.1103	4.4913
24	15	<b>3</b> 0	0	0	11.2540	4.6098
	20	30	0	0	10.2357	3.5035
	25	30	0	0	8.5533	3.9139

Table 10. Computational results for the hybrid flowshop having larger number of jobs with the constraint for 50% of the flowshop.

Number	Numbe	r Number	Proposed	algorithm	Best	RANDOM schedule
of	of	of	RELATIVE	C	PERCENTAGE	ERROR
jobs	stages	problems	Mean	Std.	Mean	Std.
				dev.		dev.
	5	30	0	0	13.2240	6.0898
	10	30	0	0	9.6957	6.2559
12	15	30	0.0840	0.4524	9.9926	5.9423
	20	30	0.0683	0.3678	8.0247	4.4683
	25	30	0.0637	0.3429	7.0017	4.5288
	5	30	0	0	14.4113	6.7075
	10	30	0	0	12.6983	6.1458
18	15	30	0.0680	0.3662	12.9403	4.6868
	20	30	0	0	11.0817	4.3073
	25	30	0	0	9.2674	4.0482
	5	30	0	0	17.7593	6.5639
	10	30	0	0	14.3215	4.9060
24	15	30	0	0	13.3860	5.0084
	20	30	0	0	13.4077	4.0197
	25	30	0.0122	0.0659	11.4260	4.7190

Table 11.	Computational results for the hybrid flowshop
having	larger number of jobs with the constraint
	for 75% of the flowshop.

## Acknowledgement

The authors are indebted to referees for their constructive comments and valuable suggestions for improving the earlier version of the paper.

## References

- [1] Bansal, S.P.: Minimizing the sum of completion times of *n*-jobs over *M*-machines in a flowshop A branch and bound approach, AIIE Transactions, Vol. 9 (1977), 306-311.
- [2] Bonney, M.C. and Gundry, S.W.: Solutions to the constrained flowshop sequencing problem, *Operational Research Quarterly*, Vol. 24 (1976), 869-883.
- [3] Campbell, H.G., Dudek, R.A. and Smith, M.L.: A Heuristic Algorithm for the n-job, M-machine sequencing problem. Management Science, Vol. 16 (1970), B630-B637.
- [4] Conway, R.W., Maxwell W.L. and Miller, L.W.: Theory of Scheduling, Addison-Wesley Publishing Co., Massachusetts, 1967.
- [5] Dannenbring, D.G.: An evaluation of flowshop sequencing heuristics, Management Science, Vol. 23 (1977), 1174-1182.
- [6] Gupta, J.N.D. and Dudek, R.A.: Optimality criteria for flowshop schedule, AIIE Transactions, Vol. 3 (1971), 199-205.

- [7] Gupta, J.N.D.: Heuristic algorithms for multistage flowshop scheduling problem, AIIE Transactions, Vol. 4 (1972), 11-18.
- [8] Ignall, E. and Schrage, L.: Application of the branch and bound technique to some flowshop scheduling problems, *Operations Research*, Vol. 13 (1965), 400-412.
- [9] King, J.R. and Spachis, A.S.: Heuristics for flowshop scheduling, International Journal of Production Research, Vol. 18 (1980), 343-357.
- [10] Lominicki, Z.A.: A branch-and-bound algorithm for the exact solution of the three machine scheduling problem, *Operational Research Quarterly*, Vol. 16 (1965), 89-100.
- [11] Miyazaki, S., Nishiyama, N. and Hashimoto, F.: An adjacent pairwise approach to the mean flow-time scheduling problem, Journal of the Operations Research Society of Japan, Vol. 21 (1978), 287-299.
- [12] Nawaz, M. and Emory Enscore, E.: A heuristic algorithm for the *m*-machine, *n*-job sequencing problem, *Omega*, Vol. 11 (1983), 91-95.
- [13] Panwalkar, S.S., Dudek, R.A. and Smith, M.L.: Sequencing research and the industrial scheduling problem, Proceedings of the Symposium on the Theory of Scheduling and its applications (May 1972), Springer-Verlag, Berlin, 1973.
- [14] Reddi, S. and Ramamoorthy, C.: On the flowshop sequencing problem with no wait in process, Operational Research Quarterly, Vol. 23 (1972), 323-331.
- [15] Stinson, J.P. and Smith, A.E.: A heuristic programming procedure for sequencing in static flowshop, International Journal of Production Research, Vol. 20 (1982), 753-764.
- [16] Szwarc, W.: The flowshop problem with mean completion time criterion, *IIE Transac*tions, Vol. 15 (1983), 172–176.
- [17] Van Deman, D.M. and Baker, K.R.: Minimizing mean flowtime in the flowshop with no intermediate queues, AIIE Transactions, Vol. 6 (1974), 28-34.
- [18] Wismer, D.A.: Solution of the flowshop sequencing problem with no intermediate queues, Operations Research, Vol. 20 (1972), 689-697.

Chandrasekharan Rajendran Division of Industrial Engineering and Management Department of Humanities and Social Sciences Indian Institute of Technology Madras - 600 036 India