International Journal of Industrial Engineering Computations 6 (2015) 173–184

Contents lists available at GrowingScience

International Journal of Industrial Engineering Computations

homepage: www.GrowingScience.com/ijiec

A heuristic algorithm for scheduling in a flow shop environment to minimize makespan

Arun Gupta[a*] and Sant Ram Chauhan[b]

[a]*M. Tech. Student, National Institute of Technology, Hamirpur – 177005, Himachal Pradesh, India*
[b]*Assistant Professor, Department of Mechanical Engineering, National Institute of Technology, Hamirpur – 177005, Himachal Pradesh, India*

C H R O N I C L E

Article history:
Received March 1 2014
Received in Revised Format August 5 2014
Accepted December 9 2014
Available online December 10 2014

Keywords:
Flow-shop
Heuristics
Makespan
Scheduling
Benchmark Problems

A B S T R A C T

Scheduling 'n' jobs on 'm' machines in a flow shop is NP- hard problem and places itself at prominent place in the area of production scheduling. The essence of any scheduling algorithm is to minimize the makespan in a flowshop environment. In this paper an attempt has been made to develop a heuristic algorithm, based on the reduced weightage of machines at each stage to generate different combination of 'm-1' sequences. The proposed heuristic has been tested on several benchmark problems of Taillard (1993) [Taillard, E. (1993). Benchmarks for basic scheduling problems. *European Journal of Operational Research*, 64, 278-285.]. The performance of the proposed heuristic is compared with three well-known heuristics, namely Palmer's heuristic, Campbell's CDS heuristic, and Dannenbring's rapid access heuristic. Results are evaluated with the best-known upper-bound solutions and found better than the above three.

1. Introduction

Scheduling is a decision making practice used on a regular basis in most of the manufacturing industries. Its aim is to optimize the objectives with the allocation of resources to tasks within the given time periods. The resources and tasks in an organization can take a lot of different forms. The resources may be machines in a workshop, processing units in a computing environment and so on. The tasks may be jobs or operations in a production process, executions of computer programs, stages in a construction project, and so on. The objectives can take many different forms and one objective may be the minimization of total completion time of jobs. A typical flow shop scheduling problem involves the determination of the order of processing of jobs with different processing times over different machines. Consider an m-machine flow shop where there are *n*-jobs to be processed on the m machines in the same order. The prime objective is to generate the optimal sequence of processing jobs that minimize the total completion time of all jobs. Scheduling of operations is very difficult issues in the planning and managing of manufacturing processes. Toughness and easiness of scheduling task depends on shop environment, process constraints and the performance measures. Due to the complexity of flow shop scheduling

* Corresponding author.
E-mail: engg.arun12@gmail.com (A. Gupta)

doi: 10.5267/j.ijiec.2014.12.002

problem, exact methods become impractical for instances with medium to large number of jobs and machines. This has introduced the basis for development and adoption of various heuristic algorithms. The flow-shop problem was first studied by Johnson (1954) for two machines. He considered the problem with respect to total completion time as objective function for both m=2 and m>2 flow shops. For $m \geq 2$ it becomes a NP-hard problem (Gonzalez & Sahni, 1978). Many researchers have generalized the Johnson's rule to 'm' machine flow shop scheduling heuristics. While first heuristic for makespan minimization for the flow shop scheduling problem was introduced by Palmer (1965). The heuristic calculates a slope index for each job and then schedules the jobs in descending order of the slope index. Campbell et al. (1970) developed an extension of Johnson's algorithm. The Campbell, Dudek, and Smith (CDS) heuristic generate m-1 sequences by converting m original machines into two auxiliary machines and then solving the two machine problem using Johnson's rule repeatedly. Finally, the best sequence is selected. CDS heuristic performs better as compared to the Palmer heuristic.

Gupta (1971) suggested another heuristic which was similar to Palmer's heuristic. He defined his slope index based on the optimality of Johnson's rule for three machine problem. Dannenbring (1977) developed a method called rapid access (RA). It attempts to combine the advantages of Palmers slope index and the CDS methods. Its purpose is to provide a good solution as quickly as possible. RA heuristic solves only one artificial problem using Johnson's rule in which a waiting scheme is used to determine the processing times for two auxiliary machines. The NEH heuristic algorithm made by Nawaz, Enscore, and Ham (1983) is based on the assumption that the job with larger total processing time should be given higher priority than job with low total processing time. Then, it generates the final sequence by adding a new job at each step and the best partial solution is found.

Hundal and Rajgopal (1988) proposed an improvement in the Palmer's heuristic. Two more slope indexes are calculated and with these two slope indexes and the original Palmer's slope index, three sequences are calculated and the best one is given as a final result. Taillard (1993) proposed 260 scheduling problems that are randomly generated. The problem size corresponds to the practical aspects of industry related problems. They proposed problems for general flow shop, job shop and open shop scheduling problems. The main objective of the problems is the minimization of makespan. Rajendran (1994) introduced a new heuristic for flow shop, in which heuristic preference relation is developed. He considered the problem of scheduling in flow shop and flow-line based manufacturing cell with bi-criteria of minimizing makespan and total flow time of jobs.

Rajendran and Zeigler (1997) developed a heuristic procedure with an objective of minimizing makespan, where set-up, processing and removal times are separable. Large number of randomly generated problems is used for the evaluation of heuristic. Danneberg et al. (1999) proposed and compared various heuristic algorithms for permutation flow shop scheduling problem including setup times with objective function of weighted sum of makespan and completion times of the jobs. Chakraborty and Laha (2007) modified the original NEH algorithm for makespan minimization problem in permutation flow shop scheduling. Computational study reveals that the quality of the solution is significantly improved while maintaining the same algorithmic complexity. Ruiz and Stutzle (2007) presented a new iterated greedy algorithm that applies two phases iteratively, named destruction, where some jobs are eliminated from the incumbent solution, and construction, where the eliminated jobs are reinserted into the sequence using the well-known NEH construction heuristic.

Chia and Lee (2009) introduced the concept of learning effect in a permutation flow shop for total completion time problems. This concept plays an important role in production environments. In addition, the performances of various well-known heuristics are evaluated with the presence of learning effect. Jabbarizadeh et al. (2009) considered hybrid flexible flow shops with sequence-dependent setup times and machine availability constraints caused by preventive maintenance. Three heuristics, based on SPT, LPT and Johnson rule and two meta-heuristics based on genetic algorithm and simulated annealing is proposed. Zobolas et al. (2009) proposed a hybrid metaheuristic for the minimization of makespan in permutation flow shop scheduling problems in which a greedy randomized constructive heuristic provides an initial solution and then it is improved by genetic algorithm (GA) and variable neighbourhood search (VNS). Ramezanian et al. (2010) presented a new discrete firefly meta-heuristic to minimize the makespan for the permutation flow shop scheduling problem. The results of implementation of the proposed method are compared with other existing ant colony optimization technique which indicate the superiority of new proposed method over the ant colony for some well-known benchmark problems. Wang et al. (2010) proposed a novel hybrid discrete differential evolution (HDDE) algorithm for solving blocking flow shop scheduling problems to minimize the maximum completion time.

Shu-Hui Yang and Ji-Bo Wang (2011) considered the minimization of total weighted completion time in a two-machine flow shop under simple linear deterioration. The objective was to obtain a sequence so that the total weighted completion time is minimized. Chiang et al. (2011) proposed a memetic algorithm by integrating a general multi-objective evolutionary algorithm with a problem-specific heuristic (NEH). Cheng et al (2011) proposed a hybrid algorithm three frequently applied ones: the dispatching rule, the shifting bottleneck procedure, and the evolutionary algorithm. Bhongade and Khodke (2012) proposed two heuristics NEH-BB (Branch & Bound) and Disjunctive to solve assembly flow shop scheduling problem where every part may not be processed on each machine. By computational experiments these methods are found to be applicable to large size problems. Khalili and Reza (2012) presents a new multi-objective electromagnetism algorithm (MOEM) based on the attraction–repulsion mechanism of electromagnetic theories. Choi and Wang (2012) presented a novel decomposition-based approach (DBA), which combines both the shortest processing time (SPT) and the genetic algorithm (GA), to minimizing the makespan of a flexible flow shop (FFS) with stochastic processing times. Computation results show that the DBA outperforms SPT and GA alone for FFS scheduling with stochastic pro-cessing times.

Pour et al. (2013) presented an efficient solution strategy based on a genetic algorithm (GA) to minimize the makespan, total waiting time and total tardiness in a flow shop consisting of n jobs and m machines. Fattahi et al. (2013) presented a two-stage hybrid flow shop scheduling problem with setup and assembly operations. A combinatorial algorithm is proposed using heuristic, genetic algorithm (GA), simulated annealing (SA), NEH and Johnson's algorithm to solve the problem. Jaroslaw et al. (2013) proposed a new idea of the use of simulated annealing method to solve certain multi-criteria problem. Li et al. (2013) proposed a mathematical model for a two-stage flexible flow shop scheduling problem with task tail group constraint, where the two stages are made up of unrelated parallel machines. Behnamian and Ghomi (2014) considered bi-objective hybrid flow shop scheduling problems with bell-shaped fuzzy processing and sequence-dependent setup times. To solve these problem a bi-level algorithm with a combination of genetic algorithm and particle swarm optimization algorithm is used. Wang and Choi (2014) presented a novel decomposition-based holonic approach (DBHA) for minimising the makespan of a flexible flow shop (FFS) with stochastic processing times. Rahmani and Heydari (2014) proposed a new approach to achieve stable and robust schedule despite uncertain processing times and unexpected arrivals of new jobs. Computational results indicate that this method produces better solutions in comparison with four classical heuristic approaches according to effectiveness and performance of solutions.

The above literature review reveals the continuous interest shown by the researchers in solving flow shop scheduling problems. As the problem became NP-hard, most of the researchers developed heuristic methods to obtain optimal schedule of jobs but over the past few years hybrid heuristics / meta-heuristics have been developed to improve the accuracy of results. In these techniques, an initial solution is obtained from existing heuristics and this solution is further improved by using meta-heuristics. In this paper, an attempt has been made to develop a simple heuristic without much sacrificing the accuracy to provide an initial solution for other methods to solve the flow shop scheduling problems for minimizing makespan.

The proposed heuristic is based on the reduced weightage scheme of machines at each stage to generate different combination of sequences for producing optimal results.
The rest of this paper is organized as follows: Section 2 provides basic assumptions and statement of the problem. Section 3 introduces the concept and flowchart of proposed heuristic algorithm. Section 4 describes the evaluation of heuristic methods with experiment design and a detailed presentation of computational results. Towards the end the conclusion are drawn in section 5.

2. Problem Formulation

2.1 Problem Statement

In a flow-shop scheduling problem, a set of n jobs (*1, …, n*) are processed on a set of m machines (*1, …, m*) in the same technological order, i.e. first in machine 1 then on machine 2 and so on until machine m. The objective is to find a sequence for the processing of the jobs in the machines so that the total completion time or makespan of the schedule (C_{max}) is minimized. Let $t_{i,j}$ denote the processing time of the job in position i ($i = 1, 2, \dots, n$) on machine j ($j = 1, 2, \dots, m$). Let $C_{i,j}$ denote the completion time of the job in position i on machine j. Therefore we have:

$$C_{1,1} = t_{1,1} \tag{1}$$
$$C_{i,1} = C_{i-1,1} + t_{i,1} \qquad \text{for } i = 2,\dots, n \tag{2}$$
$$C_{1,j} = C_{1,j-1} + t_{1,j} \qquad \text{for } j = 2,\dots, m \tag{3}$$
$$C_{i,j} = \max (C_{i,j-1} , C_{i-1,j}) + t_{i,j} \qquad \text{for } i = 2,\dots, n \ \ \& \ \ j = 2,\dots, m \tag{4}$$
$$\text{Total Completion Time } (C_{max}) = C_{n,m}$$

2.2 Assumptions

The assumptions regarding this problem are general and common in nature. The same are adapted from Baker (1974), Ruiz and Maroto (2005) and others.

- Each job i can be processed at most on one machine j at the same time.
- Each machine m can process only one job i at a time.
- No preemption is allowed, i.e. the processing of a job i on a machine j cannot be interrupted.
- All jobs are independent and are available for processing at time 0.
- The set-up times of the jobs on machines are negligible and therefore can be ignored.
- The machines are continuously available.
- In-process inventory is allowed. If the next machine on the sequence needed by a job is not available, the job can wait and joins the queue at that machine.

3. Proposed heuristic algorithm

The proposed heuristic algorithm is applied to the processing of n-jobs through m-machines with each job following the same technological order of machines. The algorithm is based on the weightage of machines which is reduced at each stage to generate different combination of sequences of processing jobs to minimize the given performance measure. Similar to CDS heuristic, the algorithm generates *m-1* sequences. The algorithm converts the original m-machines problem into *m-1* artificial 2-machine problems. A weight parameter, $w_{i,j}$ is assigned at each stage which is used in a reverse manner for the two artificial machines. Johnson's rule is then applied to first artificial 2-machine problem to determine the sequence of jobs and the process is repeated by reducing the weight parameter until m-1 sequences are found. Then, makespan value is computed and the sequence with the minimum makespan value is selected as best sequence. The necessary steps for solving a given problem are as follows.

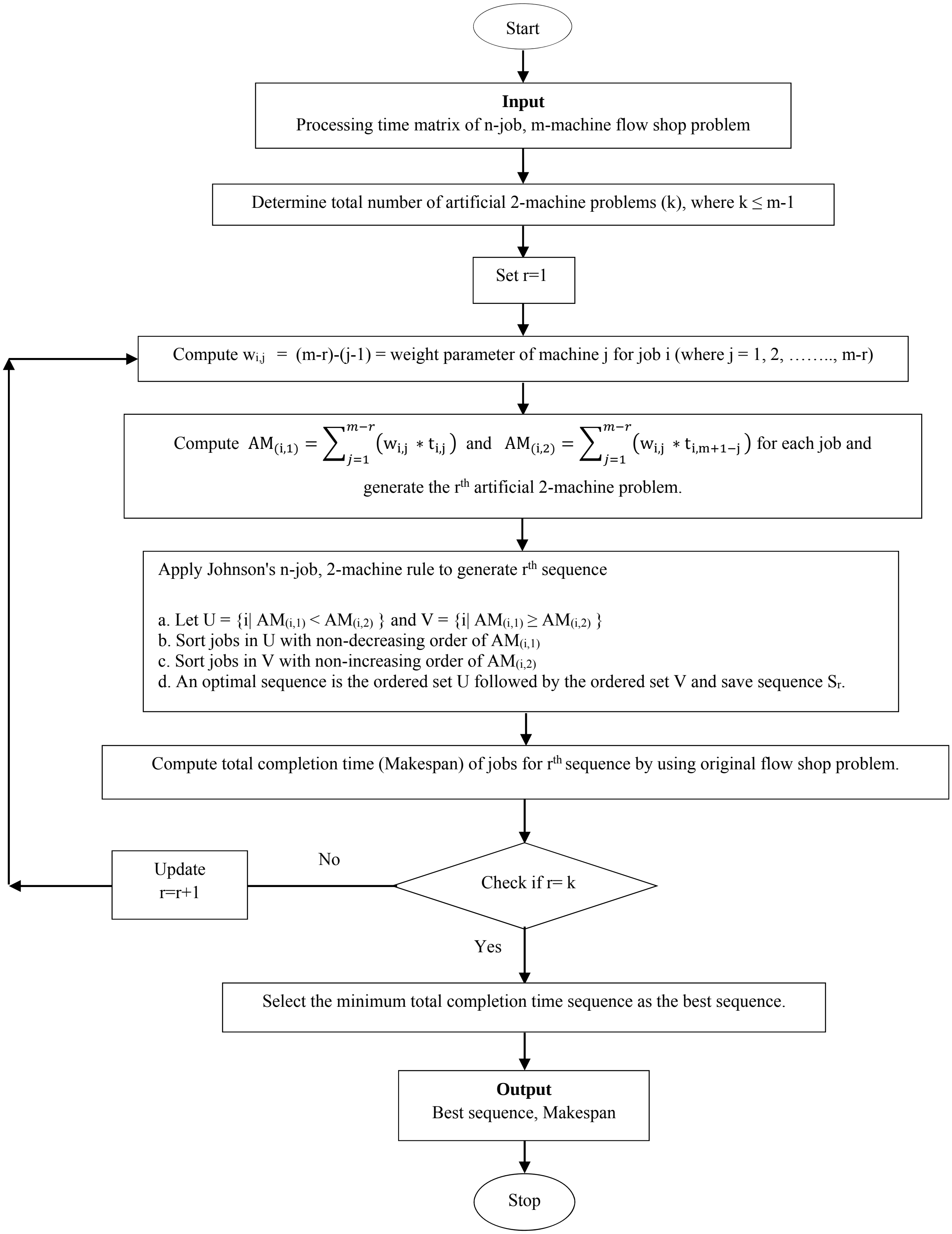

Fig. 1. Flow chart of proposed heuristic algorithm

4. Heuristics evaluation

4.1 Experiment design

In this section, we compare the performance of the heuristic algorithms using the MATLAB software on a HP 430 workstation with INTEL(R) Core(TM)-i3 CPU, M370 @ 2.40 GHz, 2GB RAM processor.

The evaluation of the heuristics is done by varying the number of jobs and the number of machines. The benchmark problems for evaluating proposed heuristic and making a comparative study are taken from Taillard (1993). These problems with their best-known upper bound solutions are taken from the OR Library (http://mscmga.ms.ic.ac.uk/info.html). These test problems have varying sizes with number of jobs varying from 20 to 500 and the number of machines varying from 5 to 20. There are 120 instances from Taillard's benchmark problems, 10 each of sizes 20×5, 20×0, 20×20, 50×5, 50×10, 50×20, 100×5, 100×10, 100×20, 200×10, 200×20 and 500×20.

Each instance is solved by the proposed heuristic, Palmer, CDS and RA heuristic algorithms. Best-known upper bounds for these problems are used for comparison purposes. We compare the performance of the heuristics using one measure: average percentage gap. The gap, in percent, which refers to as the difference between the Makespan and Upper Bound, is calculated by:

$$\%\,Gap = \frac{Makespan - Upper\ Bound}{Upper\ Bound} \times 100 \tag{5}$$

4.2 Computational results

Tables 1-4 show the results for Taillard's 20-job, 50-job, 100-job & 200-job and 500-job benchmark problems. In each of these tables, we display the results for Proposed Heuristic, Palmer, CDS and RA. We also show the best-known upper bounds and percentage gap from the best-known upper bound for each problem. The bold figures represent the minimum percentage gap for the particular problem. A summary of the average percentage gap (across all jobs and machines) is given in Table 5.

Table 1

Makespans and percentage gaps for Taillard's 20-Job benchmark problems

Problem Description		Makespan				Gap (%)			
Problem Instance	Upper Bound	Proposed Heuristic	Palmer	CDS	RA	Proposed Heuristic	Palmer	CDS	RA
20x5									
1	1278	1367	1384	1390	1381	**6.96**	8.29	8.76	8.06
2	1359	1432	1439	1424	1450	5.37	5.89	**4.78**	6.7
3	1081	1162	1162	1255	1194	**7.49**	**7.49**	16.1	10.45
4	1293	1402	1490	1418	1406	**8.43**	15.24	9.67	8.74
5	1236	1300	1360	1323	1293	5.18	10.03	7.04	**4.61**
6	1195	1276	1344	1312	1308	**6.78**	12.47	9.79	9.46
7	1239	1393	1400	1393	1445	**12.43**	12.99	**12.43**	16.63
8	1206	1291	1313	1345	1291	**7.05**	8.87	11.53	**7.05**
9	1230	1352	1426	1360	1344	9.92	15.94	10.57	**9.27**
10	1108	1190	1229	1164	1187	7.4	10.92	**5.05**	7.13
20x10									
1	1582	1658	1790	1757	1771	**4.8**	13.15	11.06	11.95
2	1659	1802	1948	1854	1869	**8.62**	17.42	11.75	12.66
3	1496	1621	1729	1651	1637	**8.36**	15.57	10.36	9.43
4	1378	1548	1585	1547	1543	12.34	15.02	12.26	**11.97**
5	1419	1638	1648	1558	1672	15.43	16.14	**9.8**	17.83
6	1397	1557	1527	1591	1615	11.45	**9.31**	13.89	15.6
7	1484	1576	1735	1630	1657	**6.2**	16.91	9.84	11.66
8	1538	1733	1763	1766	1892	**12.68**	14.63	14.82	23.02
9	1593	1755	1836	1720	1858	10.17	15.25	**7.97**	16.64
10	1591	1846	1898	1884	1959	**16.03**	19.3	18.42	23.13
20x20									
1	2297	2559	2818	2559	2743	**11.41**	22.68	**11.41**	19.42
2	2100	2303	2331	2285	2515	9.67	11	**8.81**	19.76
3	2326	2567	2678	2565	2742	10.36	15.13	**10.27**	17.88
4	2223	2458	2629	2415	2509	10.57	18.26	**8.64**	12.87
5	2291	2454	2704	2506	2671	**7.11**	18.03	9.38	16.59
6	2226	2424	2572	2422	2520	8.89	15.54	**8.81**	13.21
7	2273	2421	2456	2489	2506	**6.51**	8.05	9.5	10.25
8	2200	2343	2435	2362	2520	**6.5**	10.68	7.36	14.55
9	2237	2450	2754	2409	2700	9.52	23.11	**7.69**	20.7
10	2178	2331	2633	2439	2575	**7.02**	20.89	11.98	18.23
Averages						**9.02**	14.14	10.59	13.51

For Taillard's 20-job problems, i.e., 20×5, 20×10 and 20×20 size problems, proposed heuristic provides the minimum average gap of for all three problem sets as 7.7%, 10.61% and 8.76% respectively. RA heuristic gives closer results with average gap of 8.81% for instance of size 20×5 and CDS with average gap of 12.81 % and 9.39% for 20×10 and 20×20 respectively (see Table 1).

For Taillard's 50-job problems, the results are quite similar to that of 20-job problems. At instances of size 50×5, the proposed heuristic results are better than others with an average gap of 4.09%, at size 50×10 with 10.96% and at size 50×20 with 12%. The results, which are closer to the proposed heuristic, are of Palmer with an average gap of 5.34% for size 50×5 problems and of CDS with an average gap of 12.43% and 13.31% for size 50×10 and 50×20 problems respectively (see Table 2).

Table 2
Makespans and percentage gaps for Taillard's 50-Job benchmark problems

Problem Description		Makespan				Gap (%)			
Problem Instance	Upper Bound	Proposed Heuristic	Palmer	CDS	RA	Proposed Heuristic	Palmer	CDS	RA
50x5									
1	2724	2800	2774	2883	2803	2.79	**1.84**	5.84	2.9
2	2834	3015	3041	3032	2996	6.39	7.3	6.99	**5.72**
3	2621	2702	2777	2703	2804	**3.09**	5.95	3.13	6.98
4	2751	2845	2860	2884	2876	**3.42**	3.96	4.83	4.54
5	2863	2960	2963	3038	2998	**3.39**	3.49	6.11	4.72
6	2829	2995	3090	3031	3108	**5.87**	9.23	7.14	9.86
7	2725	2893	2845	2944	2958	6.16	**4.4**	8.04	8.55
8	2683	2747	2826	2867	2884	**2.38**	5.33	6.86	7.49
9	2552	2625	2733	2784	2679	**2.86**	7.09	9.09	4.98
10	2782	2909	2915	2942	2951	**4.56**	4.78	5.75	6.07
50x10									
1	3025	3468	3478	3382	3510	14.64	14.97	**11.8**	16.03
2	2892	3174	3313	3263	3298	**9.75**	14.56	12.83	14.04
3	2864	3180	3321	3287	3380	**11.03**	15.96	14.77	18.02
4	3064	3353	3511	3393	3366	**9.43**	14.59	10.74	9.86
5	2986	3356	3427	3375	3419	**12.39**	14.77	13.03	14.5
6	3006	3309	3323	3400	3349	**10.08**	10.55	13.11	11.41
7	3107	3441	3457	3530	3592	**10.75**	11.26	13.61	15.61
8	3039	3392	3356	3371	3552	11.62	**10.43**	10.92	16.88
9	2902	3219	3414	3265	3330	**10.92**	17.64	12.51	14.75
10	3091	3368	3404	3429	3520	**8.96**	10.13	10.93	13.88
50x20									
1	3875	4256	4272	4324	4736	**9.83**	10.24	11.59	22.22
2	3715	4255	4303	4216	4374	14.54	15.83	**13.49**	17.74
3	3668	4104	4210	4203	4384	**11.89**	14.78	14.59	19.52
4	3752	4203	4233	4267	4535	**12.02**	12.82	13.73	20.87
5	3635	4091	4376	4122	4336	**12.54**	20.38	13.4	19.28
6	3698	4140	4312	4238	4295	**11.95**	16.6	14.6	16.14
7	3716	4138	4306	4134	4404	11.36	15.88	**11.25**	18.51
8	3709	4173	4310	4283	4306	**12.51**	16.2	15.48	16.1
9	3765	4254	4547	4219	4402	12.99	20.77	**12.06**	16.92
10	3777	4167	4197	4264	4383	**10.33**	11.12	12.89	16.04
Averages						**9**	11.43	10.71	13

For Taillard's 100-job problems, i.e. for the instance size of 100×5, the minimum average gap from the upper bound is 2.33% at Palmer compared with 2.88% at proposed heuristic. Proposed heuristic offer good results with an average gap of 7.64% for the size instance of 100×10 and 10.53% for 100×20 (see Table 3).

Table 3
Makespans and percentage gaps for Taillard's 100-Job benchmark problems

Problem Description		Makespan				Gap (%)			
Problem Instance	Upper Bound	Proposed Heuristic	Palmer	CDS	RA	Proposed Heuristic	Palmer	CDS	RA
100x5									
1	5493	5673	5749	5592	5730	3.28	4.66	**1.8**	4.31
2	5268	5380	5316	5548	5464	2.13	**0.91**	5.31	3.72
3	5175	5452	5325	5493	5399	5.35	**2.9**	6.14	4.33
4	5014	5148	5049	5273	5222	2.67	**0.7**	5.17	4.15
5	5250	5286	5317	5484	5421	**0.69**	1.28	4.46	3.26
6	5135	5316	5274	5259	5344	3.52	2.71	**2.41**	4.07
7	5246	5346	5376	5561	5322	1.91	2.48	6	**1.45**
8	5106	5273	5263	5387	5318	3.27	**3.07**	5.5	4.15
9	5454	5694	5606	5758	5677	4.4	**2.79**	5.57	4.09
10	5328	5413	5427	5708	5437	**1.59**	1.86	7.13	2.05
100x10									
1	5770	6153	6161	6239	6256	**6.64**	6.78	8.13	8.42
2	5349	5745	5889	5851	5962	**7.4**	10.1	9.38	11.46
3	5677	5945	6127	6023	6090	**4.72**	7.93	6.09	7.27
4	5791	6262	6313	6408	6494	**8.13**	9.01	10.65	12.14
5	5468	5915	6070	6018	6147	**8.17**	11.01	10.06	12.42
6	5303	5745	5870	5751	5995	**8.33**	10.69	8.45	13.05
7	5599	6229	6442	6202	6281	11.25	15.06	**10.77**	12.18
8	5623	6194	6168	6196	6330	10.15	**9.69**	10.19	12.57
9	5875	6281	6081	6349	6405	6.91	**3.51**	8.07	9.02
10	5845	6117	6259	6387	6199	**4.65**	7.08	9.27	6.06
100x20									
1	6286	6957	7075	6962	7171	**10.67**	12.55	10.75	14.08
2	6241	6853	7058	6970	7109	**9.81**	13.09	11.68	13.91
3	6329	7102	7221	7233	7274	**12.21**	14.09	14.28	14.93
4	6306	7027	7039	7148	7178	**11.43**	11.62	13.35	13.83
5	6377	7057	7259	7118	7548	**10.66**	13.83	11.62	18.36
6	6437	7143	7109	7279	7306	10.97	**10.44**	13.08	13.5
7	6346	6972	7279	7124	7351	**9.86**	14.7	12.26	15.84
8	6481	7184	7567	7181	7717	10.85	16.76	**10.8**	19.07
9	6358	7017	7271	7181	7621	**10.36**	14.36	12.94	19.86
10	6465	7013	7305	7144	7476	**8.48**	12.99	10.5	15.64
Averages						**7.01**	8.29	8.73	9.97

For Taillard's 200-job and 500-job problems (200×10, 200×20, 500×20) the solutions found by proposed heuristic are quite similar to those of 100-job problems. The minimum average gap from the upper bound is 5.02% at Palmer compared with 5.32% at proposed heuristic for the instance of size 200×10 and proposed heuristic provides good results with an average gap of 9.4% for the size instance of 200×20 and 6.29% for 500×20 (see Table 4).

Table 4
Makespans and percentage gaps for Taillard's 200-Job and 500-job benchmark problems

Problem Description	Makespan					Gap (%)			
Problem Instance	Upper Bound	Proposed Heuristic	Palmer	CDS	RA	Proposed Heuristic	Palmer	CDS	RA
200x10									
1	10868	11258	11443	11610	11382	**3.59**	5.29	6.83	4.73
2	10494	11093	10986	11358	11189	5.71	**4.69**	8.23	6.62
3	10922	11412	11336	11732	11401	4.49	**3.79**	7.42	4.39
4	10889	11210	11221	11381	11309	**2.95**	3.05	4.52	3.86
5	10524	11107	11125	11324	11146	**5.54**	5.71	7.6	5.91
6	10331	11128	10865	11337	11060	7.71	**5.17**	9.74	7.06
7	10857	11380	11303	11649	11451	4.82	**4.11**	7.29	5.47
8	10731	11310	11275	11470	11536	5.4	**5.07**	6.89	7.5
9	10438	11171	11184	11259	11277	**7.02**	7.15	7.87	8.04
10	10676	11315	11333	11515	11516	**5.98**	6.15	7.86	7.87
200x20									
1	11294	12587	13042	12536	12673	11.45	15.48	**11**	12.21
2	11420	12400	12813	12558	12849	**8.58**	12.2	9.96	12.51
3	11446	12513	12846	12804	12784	**9.32**	12.23	11.86	11.69
4	11347	12477	13053	12623	12671	**9.96**	15.03	11.25	11.67
5	11311	12292	12827	12536	12505	**8.67**	13.4	10.83	10.56
6	11282	12316	12404	12440	12502	**9.16**	9.94	10.26	10.81
7	11456	12293	12584	12711	12793	**7.31**	9.85	10.95	11.67
8	11415	12409	12824	12621	12699	**8.71**	12.34	10.56	11.25
9	11343	12350	12523	12666	12470	**8.88**	10.4	11.66	9.94
10	11422	12789	12642	12913	13057	11.97	**10.68**	13.05	14.31
500x20									
1	26189	27881	28227	28385	28131	**6.46**	7.78	8.38	7.42
2	26629	28542	29441	29091	29549	**7.18**	10.56	9.25	10.97
3	26458	28141	28087	28639	28585	6.36	**6.16**	8.24	8.04
4	26549	28346	28109	29058	29014	6.77	**5.88**	9.45	9.28
5	26404	27715	27768	28260	28126	**4.96**	5.17	7.03	6.52
6	26581	28127	28516	28706	28304	**5.82**	7.23	7.99	6.48
7	26461	27956	27878	28410	28525	5.65	**5.35**	7.37	7.8
8	26615	28271	28296	28904	28670	**6.22**	6.32	8.6	7.72
9	26083	27816	27734	28503	28091	6.64	**6.33**	9.28	7.7
10	26527	28348	28313	28653	28615	6.86	**6.73**	8.01	7.87
Averages						**7**	7.98	8.97	8.59

Overall the proposed heuristic algorithm performed better than Palmer, CDS and RA heuristics. Out of 120 benchmark problems considered, our heuristic algorithm performs better for 74 problems, and for the remaining problems also the results are very close to other heuristic algorithms. The average gap from the best-known upper bound was only 8% for all Taillard's problems (see Table 5).

The average percentage gap decreases for all heuristics as the number of job increases and increases as the number of machine increases and proposed heuristic provides the minimum average percent gaps (Fig.2 and Fig.3). Therfore, it can be seen that for increasing number of jobs and machines, proposed heuristic performs better than the existing ones in terms of makespan as performance measure.

Table 5
Average percentage gaps for Taillard benchmark problems.

Instance Size	No. of Instances	Average Gap (%)			
		Proposed Heuristic	Palmer	CDS	RA
20x5	10	**7.7**	10.81	9.57	8.81
20x10	10	**10.61**	15.27	12.81	15.39
20x20	10	**8.76**	16.34	9.39	16.34
50x5	10	**4.09**	5.34	6.38	6.18
50x10	10	**10.96**	13.49	12.43	14.5
50x20	10	**12**	15.46	13.31	18.34
100x5	10	2.88	**2.33**	4.95	3.56
100x10	10	**7.64**	9.09	9.11	10.46
100x20	10	**10.53**	13.44	12.13	15.9
200x10	10	5.32	**5.02**	7.42	6.14
200x20	10	**9.4**	12.16	11.14	11.66
500x20	10	**6.29**	6.76	8.36	7.98
Overall	120	**8**	10.46	9.75	11.27

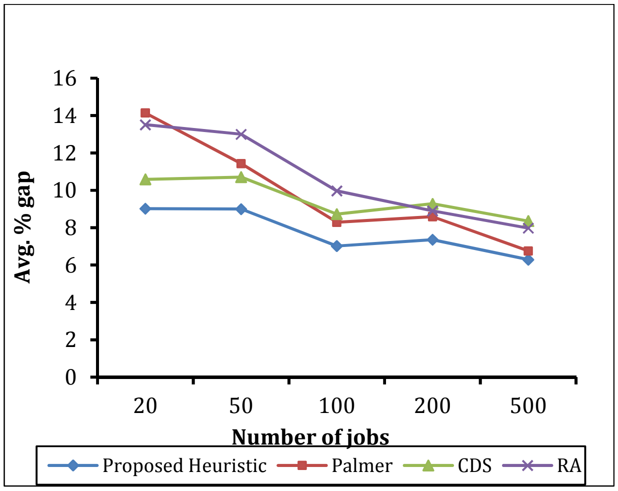

Fig. 2 Heuristics avg. % gap versus number of jobs

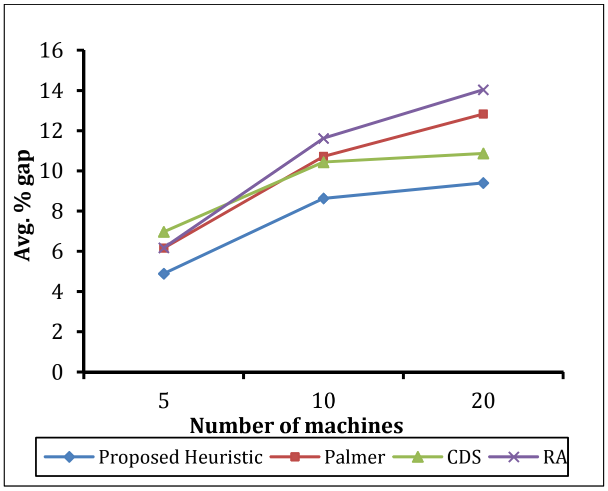

Fig. 3 Heuristics avg. % gap versus number of machines

5. Conclusion

In this paper, we have presented a heuristic for the general flow shop scheduling to minimize the makespan. The proposed method was based on the principle that weightage of the machines at each stage was reduced to obtain different combination of sequences. The sequence with minimum makespan is selected as the best sequence. The heuristic was tested using various benchmark problems taken from Taillard. The percentage gaps with best-known upper bound value were also tabulated. The computational results indicate that the proposed heuristic significantly performed better than the heuristics of CDS, Palmer and RA. Also, it can been seen that as the number of jobs increases, proposed heuristic provides good quality results. Therefore, it is the main reason to recommend this heuristic mainly for large size problems.

Future scope of this research provides the extensive use of proposed heuristics for researchers to develop hybrid heuristics / metaheuristics for solving flow shop scheduling problems and use of this algorithm for the generation of initial solutions because of the superiority over existing heuristic algorithms.

References

Baker, K.R. (1974). *Introduction to Sequencing and Scheduling*. John Wiley & Sons, New York.

Behnamiana, J. & Ghomi, S.M.T. (2014). Multi-objective fuzzy multiprocessor flowshop scheduling. *Applied Soft Computing*, 21, 139–148.

Bhongade, A.S. & Khodke, P.M. (2012). Heuristics for production scheduling problem with machining and assembly operations. *International Journal of Industrial Engineering Computations*, 3, 185–198.

Campbell, H. G., Dudek, R. A., & Smith, M. L. (1970). A heuristic algorithm for the `n' job `m' machine sequencing problem. *Management Science*, 16, B630-B637.

Chakraborty, U.K. & Laha, D. (2007). An improved heuristic for permutation flowshop scheduling. *International Journal of Information and Communnication Technology*, 1, 89–97.

Cheng, H.C., Chiang, T.C. & Fu, L.C. (2011). A two-stage hybrid memetic algorithm for multiobjective job shop scheduling. *Expert Systems with Applications,* 38, 9, 10983-10998.

Chia & Lee (2009). Minimizing the total completion time in permutation flow shop. *Computers & Operations Research,* 6, 2111-2121.

Chiang, T.C., Cheng, H.C. & Fu, L.C. (2011). NNMA: An effective memetic algorithm for solving multiobjective permutation flow shop scheduling problems. *Expert Systems with Applications,* 38, 5, 5986–5999.

Choi, S.H. & Wang, K. (2012). Flexible flow shop scheduling with stochastic processing times: A decomposition-based approach. *Computers & Industrial Engineering,* 63, 2, 362–373.

Danneberg, D., Tautenhahn, T. & Werner, F. (1999). A comparison of heuristic algorithms for flow shop scheduling problems with setup times and limited batch size. *Mathematical and Computer Modeling,* 29(9), 101–126.

Dannenbring, D. G. (1977). An evaluation of flow shop sequencing heuristics. *Management Science*, 23, 1174-1182.

Fattahi, P., Hosseini, S.M.H. & Jolai, F. (2013). Some heuristics for the hybrid flow shop scheduling problem with setup and assembly operations. *International Journal of Industrial Engineering Computations,* 4, 393–416.

Gonzalez, T. & Sahni, S. (1978). Flow shop and job shop schedules. *Operations Research*, 26, 36–52.

Gupta, J. (1971). A functional heuristic algorithm for the flow shop scheduling problem. *Operations Research Quarterly*, 22, 39-47.

Hundal, T.S. & Rajgopal, J. (1988). An extension of Palmer's heuristic for the flow-shop scheduling problem. *International Journal of Production Research*, 26(6), 1119-1124.

Jabbarizadeh, F., Zandieh, M. & Talebi D. (2009). Hybrid flexible flowshops with sequence-dependent setup times and machine availability constraints. *Computers & Industrial Engineering*, 57, 949–957.

Jarosław, P., Czesław, S. & Dominik, Z. (2013), Optimizing Bicriteria Flow Shop Scheduling Problem by Simulated Annealing Algorithm. *Procedia Computer Science,* 18, 936–945.

Johnson, S.M. (1954). Optimal two and three-stage production schedules with set-up times included. *Naval Research Logistics Quarterly*, 1, 61-68.

Khalili, M. & Reza, M.T. (2012). A multi-objective electromagnetism algorithm for a bi-objective flowshop scheduling problem. *Journal of Manufacturing Systems*, 31(2), 232–239.

Li, Z.T., Chen, Q., Mao, N.,Wang, X. & Liu, J. (2013). Scheduling rules for two-stage flexible flow shop scheduling problem subject to tail group constraint. *International Journal of Production Economics,* 146, 2, 667–678.

Nawaz, M., Enscore Jr. E. E., & Ham, I. (1983). A heuristic algorithm for the `m' machine, `n' job flow shop sequencing problem. *OMEGA*, 11, 91-95.

Palmer, D. S. (1965). Sequencing jobs through a multi stage process in the minimum total time - a quick method of obtaining a near optimum. *Operational Research Quarterly*, 16, 101-107.

Pour, N.S., Moghaddam, R.T. & Asadi, H. (2013). Optimizing a multi-objectives flow shop scheduling problem by a novel genetic algorithm. *International Journal of Industrial Engineering Computations,* 4, 345–354.

Rahmani, D. & Heydari, M. (2014). Robust and stable flow shop scheduling with unexpected arrivals of new jobs and uncertain processing times. *Journal of Manufacturing Systems,* 33, 1, 84–92.

Rajendran, C. & Ziegler, H. (1997). An efficient heuristic for scheduling in a flowshop to minimize total weighted flowtime of jobs. *European Jounal Operation Research*, 103, 129 –138.

Rajendran, C. (1994). A heuristic for scheduling on flow shop and flow line based manufacturing cell with multi criteria. *International Journal of Production Research*, 32, 2541-2558.

Ruiz, R. & Maroto, C. (2005). A comprehensive review and evaluation of permutation flowshop heuristics. *European Journal of Operational Research*, 165, 479–494.

Ruiz, R. & Stutzle, T. (2007). A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research*, 177, 2033–2049.

Sayadi, M.K., Ramezanian, R. & Nasab, N.G. (2010). A discrete firefly meta-heuristic with local search for makespan minimization in permutation flow shop scheduling problems. *International Journal of Industrial Engineering Computations,* 1, 1–10.

Taillard, E. (1993). Benchmarks for basic scheduling problems. *European Journal of Operational Research*, 64, 278-285.

Wang, K. & Choi, S.H. (2014). A holonic approach to flexible flow shop scheduling under stochastic processing times. *Computers & Operations Research,* 43, 157–168.

Wang, L., Pan, Q.K, Suganthan, P.N., Wang, W.H. & Wang, Y.M. (2010). A novel hybrid discrete differential evolution algorithm for blocking flow shop scheduling problems. *Computers & Operations Research,* 37, 3, 509–520.

Yang, S.H. & Wang, J.B. (2011). Minimizing total weighted completion time in a two-machine flow shop scheduling under simple linear deterioration. *Applied Mathematics and Computation*, 217, 4819-4826.

Zobolas, G.I., Tarantilis, C.D. & Ioannou, G. (2009). Minimizing makespan in permutation flow shop scheduling problems using a hybrid metaheuristic algorithm. *Computers & Operations Research*, 36, 1249 – 1267.