Jurnal
Teknologi

Full Paper

A COMPARISON OF HEURISTICS FOR SCHEDULING PROBLEMS IN TEXTILE INDUSTRY

Cecilia E. Nugraheni*, Luciana Abednego

Dept. of Informatics, Parahyangan Catholic University, Bandung, Indonesia

Article history
Received
15 July 2015
Received in revised form
18 October 2015
Accepted
19 January 2016

*Corresponding author
cheni@unpar.ac.id

Graphical abstract

	Machine				
Job j	**1**	**2**	**3**	**4**	**5**
t_{j1}	6	4	3	9	5
t_{j2}	8	1	9	5	6
t_{j3}	2	1	5	8	6

Abstract

Scheduling is an important problem in textile industry. The scheduling problem in textile industry generally belongs to the flow shop scheduling problem (FSSP). There are many heuristics for solving this problem. Eight heuristics, namely FCFS, Gupta, Palmer, NEH, CDS, Dannenbring, Pour, and MOD are considered and compared. Experimental results show the best heuristic is NEH and the worst heuristic is FCFS.

Keywords: Textile industry, scheduling, flow shop scheduling problem, heuristics

1.0 INTRODUCTION

In the area of textile industry, there are two basic kinds of industry, which are textile industry and apparel industry [1]. Textile industry is industry that handles the manufacturing of fibers by fiber industry, forming, dressing and colorizing of fabric; whereas apparel industry is industry that produces ready-to-wear garments.

There are two important problems faced by the Indonesia's textile industry. The first problem is the entry of products from foreigner countries. The secod problem is the condition of production machines. The relatively old machines will not only consume a large amount of energy, but also affect the optimality of working speed and the quality of products.

Scheduling is understood as assigning jobs to machines or human (such as operators) for specified time period satisfying some constraints. Scheduling has become an important problem in textile industry. Generally, scheduling problems found in textile industry can be classified into flow shop scheduling problem (FSSP). Given m machines and n jobs that will be processed on each machine, an FSSP is the problem to find a sequence of jobs that meets some particular criteria. One of the important objectives is to find the minimum makespan. Makespan is the time between the beginning of the execution of the first job of the sequence on the first machine and the completion of the execution of the last job of the sequence on the last machine.

FSSP is a popular topic that attracts many researchers. Many methods or heuristics for solving this class of problems have been proposed. In general, those heuristics can be classified into two types: constructive or improvement heuristics [2-3]. Some examples of constructive heuristics are Johnson, Gupta, Palmer, NEH, CDS algorithms, whereas some examples of improvement heuristics are genetic algorithms, simulated annealing, and tabu search [4].

Each heuristic has strengths as well as weaknesses. There is some approach for combining heuristics in order to obtain some new better heuristics. One of the approaches is hyper-heuristic.We are interested in developing a hyper-heuristics framework that can be used to solve FSSP. For a start, we study nine basic heuristics for FSSP, namely FCFS, Johnson, Gupta, Palmer, NEH, CDS, Dannenbring, Pour, and MOD algorithm. We have implemented those heuristics in a computer program and tested on some small case studies related to scheduling problem in textile industry [5].

This work is a continuation of our previous work. The goal of this work is to investigateand to compare the performance of each heuristic in solving more complex problems. Using our program, we conducted an experiment again. Differs from [5], instead of using real problems as case studies, we take the problem instances proposed by Taillard *et al*. [6].

Many similar work to ours, which is comparison of heuristic algorithms for scheduling problems, in particular, FSSP based on makespan criterion, can be found in literature, such as [2,4]. However, the numbers of heuristic algorithms presented are not as many as ours. Therefore, this paper contributes in enriching the results of research related to the comparison of heuristic algorithms for FSSP.

The remainder of this paper is structured as follows. In Section 2, we give a brief description of flow shop scheduling problems including the definition and the heuristics for FSSP. Sections 3 explain and discussthe results of computational experiments. Finally, conclusion and future work are given in Section 4.

.

2.0 FLOW SHOP SCHEDULING PROBLEMS

2.1 Definition

The definition of FSSP is given as follows: Given n jobs to be processed in the same sequence on *m* machines; the processing time of job *i* on machine *j* is fixed and given by t_{ij} (t_{ij}> 0). FSSP consists of minimizing the makespan which is the time between the beginning of the execution of the first job on the first machine and the completion of the execution of the last job on the last machine [7].

For FSSP we assume that the following conditions hold:

- Every job has to be processed at most once on machine 1, 2, …, m.
- Every machine processes only one job at a time
- Every job is processed at most on one machine at a time.
- The operations are not preemptable.
- The set-up times of the operations are included in the processing time and do not depend on the sequence.
- The operating sequences of the jobs are the same on every machine and the common sequence has to be determined.

As illustration, consider a 5-job 3-machine problem shown in Table 1 taken form [7]. A schedule with job ordering 3-5-4-2-1 yields 37 time units, whereas a schedule with job ordering 5-3-4-2-1 yields makespan 36 time units.

Table 1 FSSP Example

	Machine				
Job j	**1**	**2**	**3**	**4**	**5**
t_{j1}	6	4	3	9	5
t_{j2}	8	1	9	5	6
t_{j3}	2	1	5	8	6

2.2 Flow Shop Scheduling Heuristics

In [5], we consider nine basic heuristics for FSSP, namely: FCFS, Johnson, Palmer, Gupta, NEH, CDS, MOD, Dannenbring, and Pour algorithm.More expositions about the heuristics can be found in [6-8].

Among all heuristics, FCFS is the simplest heuristic for FSSP. The job ordering is based on the order of jobs' arrivals.

Johnson algorithm is simple but useful approach for solving n-jobs 2-machines FSSP. This algorithm splits the jobs into two sets such that the first set, S1, contains all the jobs whose processing time in the first machine is smaller than the one in the second machine, and the second set contains all the other jobs that don't satisfy that condition. The solution is constructed by ordering the jobs in the first set increasingly based on the processing time in first machine and continued by ordering the jobs in second set decreasingly based on the processing time in second machine.

Palmer and Gupta algorithms work in a similar manner. For every jobi, they define a slope index, s_i. The schedule is resulted by ordering the jobs based on the descending order of s_i values.

In NEH algorithm the jobs sequence is constructed iteratively. The construction starts with picking two jobs having largest value of total processing times and defining two partial sequences. The partial sequence having small value of makespan is then selected for subsequent iteration. Then, one by one, the other jobs is picked and placed at the best position in the partial sequence that yields lowest makespan. This is done by trying all the possible positions.

The principle of Pour algorithm is similar to NEH algorithm. A partial sequence of jobs is constructed iteratively until all jobs are picked. Differs from NEH, this algorithm is based on the idea of job exchanging instead of inserting a job into the partial sequence.

CDS and Dannenbring shared the same idea: each converts a given n-job m-machine problem (m>2) into p=m-1 number of n-job 2-machine surrogate problems. Every surrogate problem is then solved by Johnson algorithm. The sequence of the surrogate problem yielding minimum value of makespan is selected for scheduling jobs on the machines.

MOD is a constructive heuristic approach proposed in [4]. This algorithm adopts the Johnson's rule in the last step to get the minimum makespan and uses the difference between the sums of processing times for each machine as a pair-splitting strategy to make two groups of the matrix of *n*-job and *m*-machine.

3.0 COMPUTATIONAL EXPERIMENTS

In this work, we use Taillard's benchmark for our experiments. Taillard's benchmarks problem dataset consists of 120 instances, 10 each of one particular size. Taillard's datasets range from 20 to 500 jobs and 5 to 20 machines. Since all problem instances use more than two machines, in consequence, Johnson algorithm can't be used. Hence, wedon't consider Johnson algorithm in our experiments.

We ran every heuristic on each problem size. Totally there are 12 problem sizes. The experimental results are given by Table 2 to Table 13. We use H1, H2, H3, H4, H5, H6, H7, and H8 to denote FCFS, CDS, Dannenbring, Gupta, MOD, NEH, Palmer, and Pour heuristic algorithm, respectively. Also, PI stands for problem instance and *av* for average.

It can be seen that in general makespan grows proportional to the problem size. The bigger the problem size, the more time needed for solving the problem.

From totally 120 problems, NEH gives the smallest makespan for 117 problems. In contrast, FCFS yields the biggest makespan for 88 problems. Based on these results, we conclude that the best heuristic is NEH and the worst is FCFS.

Then, for each problem we calculate the average makespanneeded by every heuristic. Using the calculated makespan, we rank the heuristics from the best to the worst. The ranking is given in Figure 1.

Based on the graphic in Figure 1, we can classify theheuristics into three groups. The first group consists only one heuristic, which is NEH (H6). It gets the best performance, since for every problemit yields the smallest makespan on average.

There are four heuristics in the second group: Dannenbring(H3), MOD (H5), Palmer (H7), and Pour (H8). We may say that the performance of each heuristic is almost similar.

The last group consists of three heuristics that have the worst performance. Among the three heuristics, which are FCFS (H1), CDS (H2), and Gupta (H4), FCFS is the worst algorithm.

4.0 CONCLUSIONS

We have considered eight heuristics used for solving scheduling problems in textile industry, namely FCFS, Pour, MOD, Gupta, Palmer, NEH, CDS, and Dannenbring. Based on the experimental results, the best and the worst heuristic is NEH and FCFS, respectively.

We now are developing the hyper-heuristics framework that can be used to solve FSSP.In developing this frameworkwe use the multi-agent system approach. We also study the formal modelling of scheduling heuristics as multi-agent systems following our previous work [11, 12].

Table 2 Experimental result for 20 jobs 5 machines

PI	H1	H2	H3	H4	H5	H6	H7	H8
1	1448	1436	1381	1400	1322	1286	1384	1377
2	1545	1424	1450	1380	1433	1365	1439	1412
3	1597	1255	1194	1247	1136	1140	1162	1331
4	1754	1485	1406	1554	1475	1325	1453	1459
5	1431	1367	1293	1370	1355	1305	1360	1416
6	1616	1387	1308	1333	1299	1228	1344	1313
7	1528	1403	1445	1390	1366	1279	1400	1321
8	1428	1395	1291	1410	1312	1235	1313	1338
9	1468	1360	1344	1444	1371	1291	1426	1399
10	1404	1196	1187	1215	1235	1151	1229	1255
av	**1522**	**1371**	**1330**	**1374**	**1330**	**1261**	**1351**	**1362**

Table 3 Experimental result for 20 jobs 10 machines

PI	H1	H2	H3	H4	H5	H6	H7	H8
1	2004	1833	1771	2027	1789	1680	1790	1907
2	2104	2021	1869	1960	1820	1729	1948	1896
3	1812	1819	1637	1737	1621	1557	1729	1760
4	1726	1695	1543	1681	1575	1450	1585	1537
5	1944	1781	1672	1878	1714	1502	1648	1618
6	1877	1875	1615	1650	1607	1453	1527	1585
7	1935	1826	1657	1761	1650	1562	1735	1652
8	2044	2056	1892	2097	1799	1609	1763	1737
9	1978	1831	1858	1837	1731	1647	1836	1733
10	2051	2010	1959	2137	1917	1653	1898	1753
av	**1948**	**1875**	**1747**	**1877**	**1722**	**1584**	**1746**	**1718**

Table 4 Experimental result for 20 jobs 20 machines

PI	H1	H2	H3	H4	H5	H6	H7	H8
1	2770	2808	2743	2833	2787	2410	2818	2534
2	2543	2564	2515	2635	2331	2150	2331	2340
3	2625	2977	2742	2900	2598	2411	2678	2535
4	2800	2603	2509	2660	2541	2262	2629	2514
5	2829	2733	2671	2868	2615	2397	2704	2578
6	2597	2707	2520	2709	2439	2349	2572	2421
7	2723	2683	2506	2796	2465	2362	2456	2483
8	2697	2523	2520	2612	2467	2249	2435	2336
9	2713	2617	2700	2701	2550	2320	2754	2571
10	2830	2649	2575	2650	2557	2277	2633	2507
av	**2713**	**2686**	**2600**	**2736**	**2535**	**2319**	**2601**	**2482**

Table 5 Experimental result for 50 jobs 5 machines

PI	H1	H2	H3	H4	H5	H6	H7	H8
1	3095	2883	2803	2820	2839	2733	2774	2914
2	3515	3032	2996	2975	3152	2843	3041	3122
3	2900	3101	2804	3080	2850	2640	2777	2782
4	3073	3179	2876	3089	2941	2782	2860	2930
5	3071	3188	2998	3114	2882	2868	2963	3040
6	3195	3175	3108	3137	2959	2838	3090	2968
7	3450	3030	2990	3109	3021	2736	2845	2909
8	3140	3189	2884	3091	2827	2694	2826	2879
9	2930	3171	2672	3211	2783	2574	2733	2718
10	3188	3224	2951	3092	2827	2790	2915	2922
av	**3156**	**3117**	**2908**	**3072**	**2908**	**2750**	**2882**	**2918**

Table 6 Experimental result for 50 jobs 10 machines

PI	H1	H2	H3	H4	H5	H6	H7	H8
1	3754	3727	3510	3672	3468	3155	3478	3427
2	3685	3645	3298	3586	3174	3021	3313	3310
3	3612	3677	3380	3664	3191	2986	3321	3327
4	3669	3707	3366	3620	3417	3194	3511	3514
5	3741	3664	3419	3521	3417	3160	3427	3424
6	3736	3584	3349	3547	3340	3158	3323	3459
7	3678	3784	3592	3713	3539	3277	3457	3481
8	3773	3744	3552	3760	3407	3123	3356	3337
9	3792	3584	3330	3561	3422	3002	3414	3265
10	3845	3913	3520	3699	3370	3257	3404	3449
av	**3729**	**3703**	**3432**	**3634**	**3375**	**3133**	**3400**	**3399**

Table 7 Experimental result for 50 jobs 20 machines

PI	H1	H2	H3	H4	H5	H6	H7	H8
1	5094	4759	4736	4697	4347	4038	4272	4345
2	4730	4394	4337	4385	4387	3921	4303	4257
3	4592	4469	4384	4480	4265	3927	4210	4114
4	4797	4793	4535	4778	4360	3927	4233	4366
5	4748	4678	4336	4697	4218	3835	4376	4290
6	4946	4505	4295	4799	4320	3920	4312	4322
7	4742	4776	4404	4713	4138	3952	4306	4256
8	4763	4609	4306	4582	4295	3938	4310	4140
9	4823	4435	4402	4504	4277	3952	4547	4348
10	4901	4537	4383	4506	4222	4079	4197	4317
av	**4814**	**4596**	**4412**	**4614**	**4283**	**3949**	**4307**	**4276**

Table 8 Experimental result for 100 jobs 5 machines

PI	H1	H2	H3	H4	H5	H6	H7	H8
1	5943	5602	5730	5765	5929	5519	5749	5602
2	5878	5669	5464	5697	5436	5349	5316	5370
3	5880	5638	5399	5525	5321	5216	5325	5394
4	5675	5287	5222	5274	5310	5023	5049	5167
5	6095	5584	5421	5535	5424	5261	5317	5509
6	5753	5203	5344	5200	5278	5139	5274	5272
7	5935	5562	5321	5417	5530	5259	5376	5424
8	6068	5521	5270	5551	5230	5105	5263	5309
9	6193	5821	5677	5879	5538	5489	5606	5743
10	6157	5748	5437	5691	5606	5327	5427	5451
av	**5958**	**5564**	**5429**	**5553**	**5460**	**5269**	**5370**	**5424**

Table 9 Experimental result for 100 jobs 10 machines

PI	H1	H2	H3	H4	H5	H6	H7	H8
1	6983	6708	6256	6497	6208	5848	6161	6131
2	6558	6285	5962	6306	5745	5488	5889	5816
3	6667	6648	6090	6369	6043	5789	6126	6141
4	7300	6859	6494	6920	6371	6015	6313	6243
5	6844	6399	6147	6538	6024	5635	6070	6042
6	6591	6136	5995	6203	5852	5412	5870	5628
7	6765	6404	6281	6496	6355	5716	6442	5900
8	6517	6513	6386	6382	6300	5777	6168	6019
9	6859	6356	6405	6322	6304	5990	6081	6230
10	6930	6863	6199	6803	6287	5905	6259	6137
av	**6801**	**6517**	**6222**	**6484**	**6149**	**5758**	**6138**	**6029**

Table 10 Experimental result for 100 jobs 20 machines

PI	H1	H2	H3	H4	H5	H6	H7	H8
1	7840	7586	7171	7663	7092	6618	7075	7074
2	7591	7617	7109	7672	7194	6462	7058	7212
3	7755	7491	7284	7695	7350	6578	7181	7034
4	7885	7909	7178	7671	7226	6549	7039	7144
5	7729	7574	7548	7626	7057	6695	7259	7202
6	8072	7583	7306	7793	7168	6708	7109	7152
7	8033	8125	7351	7984	7156	6672	7279	7236
8	8138	7902	7717	7955	7425	6823	7567	7306
9	7907	7674	7593	7683	7017	6618	7271	6975
10	8099	7955	7476	7571	7267	6710	7305	7237
av	**7905**	**7742**	**7373**	**7731**	**7195**	**6643**	**7214**	**7157**

Table 11 Experimental result for 200 jobs 10 machines

PI	H1	H2	H3	H4	H5	H6	H7	H8
1	12193	12151	11382	12113	11629	10942	11443	11319
2	12796	12040	11189	12104	11236	10712	10986	11228
3	12556	12441	11401	11964	11539	11034	11336	11668
4	12198	11707	11309	11705	11397	11057	11221	11330
5	12110	11634	11146	11666	11194	10576	11125	11077
6	12116	11812	11060	11580	11438	10430	10865	10968
7	12848	12423	11451	12011	11564	10998	11333	11343
8	12294	11728	11536	12074	11361	10829	11275	11214
9	12010	12186	11279	12108	11250	10609	11184	11119
10	12274	11769	11516	11862	11436	10835	11355	11177
av	**12340**	**11989**	**11327**	**11919**	**11404**	**10802**	**11212**	**11244**

Table 12 Experimental result for 200 jobs 20 machines

PI	H1	H2	H3	H4	H5	H6	H7	H8
1	13576	13433	12673	13627	12750	11586	13042	12432
2	13628	13146	12849	13144	12494	11709	12813	12547
3	14152	13609	12784	13594	12805	11829	12846	12545
4	13479	13303	12671	13624	12734	11716	13053	12525
5	13686	13072	12499	13116	12528	11665	12827	12528
6	13917	13561	12502	13329	12491	11628	12404	12344
7	13836	13030	12793	13367	12511	11786	12584	12655
8	13855	13752	12699	13881	12561	11843	12824	12737
9	13409	13338	12470	13545	12917	11709	12523	12564
10	14101	13730	13057	13915	12873	11850	12615	12682
av	**13764**	**13397**	**12700**	**13514**	**12666**	**11732**	**12753**	**12556**

Table 13 Experimental result for 500 jobs 20 machines

PI	H1	H2	H3	H4	H5	H6	H7	H8
1	24253	23131	21813	23697	22154	20850	21460	21826
2	24488	24753	22757	24123	22587	20845	22647	22014
3	25271	24425	22219	24246	22191	21113	22300	22261
4	24893	23677	22267	23376	22046	20838	22198	22103
5	24828	23646	22406	23690	22377	20822	21605	21762
6	25123	23345	21726	23209	22146	20704	21728	22097
7	24653	24778	23128	24041	22445	21211	22433	22165
8	24533	23261	23098	23762	22227	20766	22581	21960
9	24950	24104	22230	23584	22082	20715	22010	21820
10	24229	23983	22280	23387	22419	21195	21747	22050
av	**24722**	**23910**	**22392**	**23712**	**22267**	**20906**	**22071**	**22006**

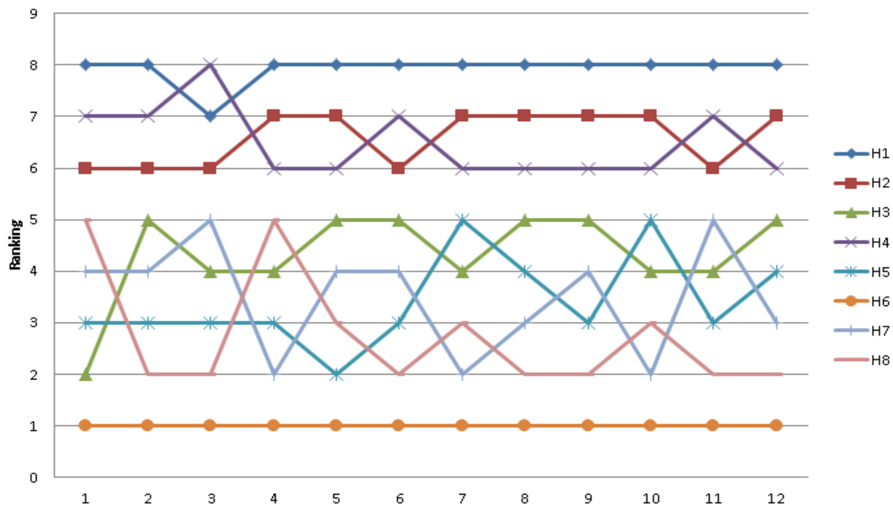

Figure 1 Heuristics ranking based the average makespan

Acknowledgement

This work was supported by Indonesian Directorate General of Higher Education (DIKTI) under research scheme HibahBersaing year 2015.

References

[1] Scholz-Retter Bernd *et al.* 2015. Applying Autonomous Control in Apparel Manufacturing. Proc. Of 9th WSEAS Int. *Conference on Robotics, Control and Manufacturing Technology*. 73-78.

[2] Vladimir Modrák and R. Sudhakara Pandian. 2010. Flow Shop Scheduling Algorithm to Minimize Completion Time for n-Jobs m-Machines Problem. *Technical Gazette*. 17(3): 273-278.

[3] Rubén Ruiz and Concepción Maroto. 2005. A Comprehensive Review and Evalution of Permutation Flowshops Heuristics. *European Journal of Operational Research*. 165: 479-494.

[4] PavolSemanco and Vladimir Modrak. 2012. A Comparison of Constructive Heuristics with the Objective of Minimizing Makespan in the Flow-Shop Scheduling Problem. *Acta Polytechnica Hungarica*. 9(5).

[5] Cecilia, E. Nugraheni and Luciana Abednego. 2015. A Survey on Heuristics for Scheduling Problem in Textile Industry. *Proc. of ICEAI 2015*. May. Kyoto, Japan.

[6] E. Taillard. 1990. Some Efficient Heuristic Methods for the Flow Shop Sequencing Problem. *European Journal of Operational Research*. 47: 65-74.

[7] Thomas Morton. 1993. *Heuristic Scheduling Systems: With Application to Production Systems and Project Management*. John Wiley & Sons.

[8] F. Wang, Y. Rao, and Y. Hou. 2011. Design and Application of A New Hybrid Heuristic Algorithm for Flow Shop Scheduling. *I.J. Computer Network and Information Security*. 2: 41-49.

[9] Sulaksmi, A., Annisa Kesy Garside, dan Fithriany Hadziqah. Penjadwalan Produksi dengan Algoritma Heuristik Pour (StudiKasus: Konveksi One Way Malang). *Jurnal Teknik Industri*. 15(1): 35-44.

[10] Robert, N. Tomastik, Peter, B. Luh, and Guandong, Liu. Scheduling Flexible Manufacturing System for Apparel Production. *IEEE Transaction on Robotics and Automation*. 12(5): 789-799.

[11] Cecilia, E. Nugraheni. Universal Properties Verification of Parameterized Parallel Systems Lecture Notes in Computer Science. 3482: 453-462.

[12] Cecilia E. Nugraheni and Luciana Abednego. A Study On Agent Scheduling Scenarios In Multi-Agent Systems. *Journal of Theoretical and Applied Information Technology*. 76(2): 238-245.