Australian Journal of Basic and Applied Sciences, 5(10): 93-98, 2011
ISSN 1991-8178

Application of Hybrid Meta-Heuristic Approach to Solve Flow-Shop Scheduling Problem

[1]Mohammad Mirabi and [2]Mahmood Dehghani Ashkezari

[1]Department of Industrial Engineering Islamic Azad University, Ashkezar Branch, Iran.
[2]Department of Biology Islamic Azad University, Ashkezar Branch, Iran.

Abstract: Flow-shop scheduling problem categorized as NP hard problem and it means development of heuristic and meta-heuristic approaches to solve it is well justified. In this paper we address a permutation flow shop scheduling problem considering the minimization of the make-span. Following that we develop a hybrid meta-heuristic algorithm for the presented problem. The proposed algorithm applied a new method for the initialization and some strong neighborhood searches. The efficiency of the algorithm is tested by numerical experiments on a large number of randomly generated problems. Comparison study demonstrates the superiority of the presented algorithm against one the recent developed method.

Key words: Hybrid meta-heuristic; permutation Flow-shop scheduling; neighborhood search.

INTRODUCTION

Scheduling is an important process widely used in manufacturing, production, management, computer science, and so on. In simple flow-shop problems, each machine centre has just one machine. A manufacturing facility that produces one or two similar products using high-volume specialized equipment like *n* assembly line is an example of a flow shop system. The flow-shop scheduling problems (FSP) have been studied for over five decades. The classical flow-shop problem with the make-span minimization criterion has always attracted the attention of researchers because of its applications in practice (Ekşioğlu *et al*., 2008).

In flow-shop system we have *m* stages in series that in each stage there are one or more machines. *n* jobs must be processed at each of the *m* stages in the same order. That is, each job has to be processed first in stage 1, then in stage 2, and so on. Processing times for each job in different stages may be different.

With a fast glance at flow-shop literature, it can be seen huge number of papers for the regular flow-shop problem with the objective of minimizing the maximum completion time across all jobs (also called make-span and denoted by C_{max}). However, the Sequence Dependent Setup Time Flow-shop Problem (SDST flow-shop problem in short) has less attention between researchers. Specific to the SDST flow-shop are the setup times (Allahverdi and Cheng, 2008).

In the flow-shop scheduling problems we are going to find a sequence for processing the jobs on the machines in accordance to optimization of a given criterion. This yields a total of $n!$ possible sequences on each machine and a total of $(n!)^m$ possible processing sequences. In flow-shop scheduling researchers usually consider permutation sequences, where the processing order of operations is the same for all machines. In this research, the athour considers sequence dependent permutation flow-shop scheduling problem with the make-span minimization criterion.

Based on Gupta (1986), the SDST flow-shop with the C_{max} criteria is NP-hard. This problem is NP-hard even when $m = 1$ and also when $m = 2$ and setups are present only on the first or second machine (Gupta and Darrow, 1986). It means exact solutions are not applicable to solve them and no way for researchers except to develop some heuristics to solve the real world problems in this area. In this research the athour consider a hybrid electromagnetism (EM) To solve the real world situation problem efficiently and effectively. The proposed EM algorithm hybridizes with an effective initialization phase and uses some strong neighborhood searches.

In the pioneering work, Johnson (1954) proposed a simple rule to obtain optimal sequences for the permutation flow-shop problem (PFSP) with two machines. This work raised high interest in the PFSP and was followed by several attempts for solving the PFSP with more than two machines. Based on the NP-completeness of the PFSP (Garey *et al*., 1976), researchers focused on the development of heuristics and meta-heuristics. Salient heuristics are developed by Campbell *et al*., (1970) (also called CDS method) and the well-known NEH heuristic by Nawaz *et al*., (1983). Some noteworthy meta-heuristics have also been proposed, for example the simulated annealing by Osman and Potts (1989), the tabu search by Widmer and Hertz (1989) and the genetic algorithm by Reeves (1995). For a recent review and evaluation of PFSP heuristics and meta-heuristics, the reader can refer to Ruiz and Maroto (2004).

Corresponding Author: Mohammad Mirabi, Department of Industrial Engineering Islamic Azad University, Ashkezar Branch, Iran.
E-mail: M.Mirabi@yahoo.com

Mercado and Bard published two papers for the sequence dependent flow-shop problem with make-span criterion (denoted as $Fm/STsd/C_{max}$). In the first paper (Mercado and Bard, 1999a), they presented a branch and bound algorithm, incorporating lower and upper bounds and dominance elimination criterion, to solve the problem. They provided test results for a wide range of problem instances. In the second paper (Mercado and Bard, 1999b), they proposed a heuristic for the same problem, which transforms an instance of the problem into an instance of the traveling salesman problem by introducing a cost function that penalizes both large setup times and bad fitness of a given schedule. Ruiz *et al*., (2005) proposed two genetic algorithms for the same problem, and showed that their heuristics outperform that of Mercado and Bard (1999b) and others. Ruiz and Stutzle (2008) presented two simple local search based iterated greedy algorithms, and showed that their algorithms perform better than those of Ruiz *et al*., (2005).

Recently, electromagnetism (EM) type algorithm has been attracted many researchers for optimization problems like scheduling. EM type algorithm has been used for optimization problems, which starts with a randomly selected points from the feasible region for a given optimization problem. EM employs an attraction-repulsion mechanism to move points (particles) towards the optimal solution. Each point (particle) is treated as a solution and has a charge. A better solution contains a stronger charge. The charge of each point relates to the objective function value we like to optimize. EM method has been tested on available test problems in Birbil and Fang (2003).

There are some researcher extended EM algorithm or applied EM to solve different problems. Debels *et al*., (2006) integrated a scatter search with EM for the solution of resource constraint project scheduling problems. It is the first paper that includes an EM type methodology for the solution of a combinatorial optimization problem. Birbil and Feyzioglu (2003) used EM type algorithms solving fuzzy relation equations, and Wu *et al*., (2005) obtained fuzzy if-then rules. Though EM algorithm is designed for solving continuous optimization problems with bounded variables, the algorithm can be extended to solve combinatorial optimization problem (COP). When we extend the EM algorithm to COPs, the first important step is the representation of a solution. Bean (1994) introduced a randomkey (RK) approach for real-coded GA for solving sequencing problem. Subsequently, numerous researchers show that this concept is robust and can be applied for the solution of different kinds of COPs (Norman and Bean, 1999; Snyder and Daskin, 2006). The random key approach is used to solve single machine scheduling problems and permutation flowshop problems using particle swarm optimization (PSO) algorithm by (Tasgetiren *et al*., 2007).

Though EM algorithm is designed for solving continuous problems, the algorithm can be extended to solve scheduling problems. Chang *et al*., (2009) applied the random-key approach to represent a schedule incorporated with the EM methodology to solve a single machine scheduling problem and the objective is to minimize the total sum of earliness and tardiness penalties. In this paper we apply priority assigning idea to incorporate EM algorithm for solving flow-shop scheduling with sequence dependent setups and make-span criterion.

2. Hybrid Meta-Heuristic:

As disscused before flow-shop scheduling problems are categorized as a hard optimization problem. Obviously a simple EM may not perform well for the real world problems. Therefore, in this research the EM hybridized with a new approach for the initializaion, acceptance criteria and some effective neighbourhood search. EM hybridizes with the modified NEH heuristic proposed by Ruiz at al., (2005) to initialization of pheromone values. Furthermore, it hybridizes with some neighbourhood search. We use three different search neighbourhoods as pair-wise interchange neighbourhood, forward insertion neighbourhood and backward insertion neighbourhood (Gupta and Smith, in press). One step in the local search is to decide whether the new sequence is accepted or not as the incumbent solution for the next iteration. We consider an acceptance criterion that is frequently used in simulated annealing (SA) algorithms. The hybrid system starts from determining whether a new solution obtained from one of initial solution using local search is accepted by SA or moved by EM.

The following algorithm is the main procedure of the presented hybrid meta-heuristic.

1. Initialization ()
2. Priority assignment
3. While number of iteration is less than the maximum number does
4. Initialize Max-iterations, Temp-start
5. Set Count = 1, T = Temp-start
6. A ← the average make-span of all solutions ()
7. x^c ← the worst make-span ()
8. x^{New} ← the best make-span ()
9. Neighborhood search ()
10. x^{Nei} ← the neighbouring make-span ()
11. Priority assignment of x^{Nei}

12. Calculate $C_{\max}(x^{Nei})$
13. If $C_{\max}(x^{Nei}) \leq$ A
14. $x^c \leftarrow x^{Nei}$, go to 27
15. Else If $C_{\max}(x^{Nei}) >$ A
16. Set T= Temp-start/log (1+Count);
17. With probability $e^{-\Delta/T}$ set $x^c \leftarrow x^{Nei}$, go to 27
18. With probability $1-e^{-\Delta/T}$
19. Move x^{Nei} by EM () and let the new solution be called x^{New}
20. Priority assignment of x^{New}
21. Calculate $C_{\max}(x^{New})$
22. If $C_{\max}(x^{New}) \leq$ A
23. $x^c \leftarrow x^{New}$, go to 26
24. Else if $C_{\max}(x^{New}) >$ A, go to 9
25. End if
26. End if
27. Increment Count by 1
28. If Count<Max-iterations, go to step 6
29. End while
30. Output the best sequence or x^{best}

2.1. Initialization:

The initial solution for EM should be generated by constructive heuristic. For the sequence dependent flow-shop scheduling with make-span criterion, we use the NEHT_RMB heuristic and a modified NEHT_RMB heuristic proposed by Ruiz *et al*., (2005). Recall that NEH is an insertion heuristic, where at each step the next unscheduled job is tentatively inserted in each possible position of some partial solution. The job is then finally inserted into the position where the objective function takes the lowest value. For executing such an insertion heuristic, the jobs need to be ordered in some way. We obtain *m* initial solutions based on this method.

2.2. Priority Assigning:

In this step we assign one random variable x_k^i between 0 and 1 to each job *k* in each solution i. For example consider one problem with 4 jobs numbered 1 to 4. Assume the second initial solution is represented by (1, 4, 3, 2). It means job 1 is the first job in the sequence, job 4 is second, job 3 is third and job 2 is the last. We assign one random variable between 0.75 and 1 to job 1, one between 0.5 and 0.75 to job 4, one between 0.25 and 0.5 to job 3 and finally one between 0 and 0.25 to job 2. One of the results can be shown as follows:

$x_1^2 = 0.92 \qquad x_4^2 = 0.67 \qquad x_3^2 = 0.37 \qquad x_2^2 = 0.09$

Therefore $x^2 = (0.92, 0.67, 0.37, 0.09)$. Also if there are *n* jobs in each sequence, one random variable between (*n*-1)/*n* and *n* is assigned to the first job, one between (*n*-2)/*n* and (*n*-1)/*n* to the second and so on. Finally random variable of the last job is between 0 and 1/n. Hence if there are m initial solutions, there are m random variables for each job *i* (*i*=1,…,*n*).

2.3. Solution Charges:

Let the force exerted on neighborhood solution by current solution *i* use the fixed charge of q_i. We have:

$$q^i = \frac{B - C_{\max}(x^i)}{\sum_{k=1}^{m}\left(B - C_{\max}(x^i)\right)}, \ \forall i = 1,...,m \tag{1}$$

where B is the average make-span of all solutions *i* (*i*=1,…,*m*). It is clear that $\sum_{i=1}^{m} q^i = 0$. After the q_i is obtained, we calculate the force on x^{Nei} by other solutions *i*. To calculate the electrostatic forces imposed by all solution for x^{Nei}, we obtain electrostatic forces imposed to each particle of x^{Nei} (particle means x_1^{Nei}, x_2^{Nei},…, x_n^{Nei}) as follows (related to the force of particle i indicated in 8):

$$F_k^{Nei} = \sum_{i=1}^{n} F_k^i = \sum_{i=1}^{n} \left(x_k^i \times q^i\right), \ \forall k = 1,...,n \tag{2}$$

Therefore

$$x_k^{New} = x_k^{Nei} + F_k^{Nei}, \ \forall k = 1,...,n \tag{3}$$

We sort all jobs in x^{New} based on its x_k^{New} in decreasing order and obtain a new sequence of jobs corresponding x^{New}. Thus solution x^{Nei} moves to $x^{Nei} + F_k^{Nei}$. For example if the solution related to x^{Nei} is represented by (2, 1, 4, 3) and new particles of x^{New} are (0.22, 0.52, 0.43, 0.85), the new solution will be (3, 1, 4, 2). Therefore to obtain x^{New}, following algorithm is used.

1. For i = 1 to m
2. $q^i = \dfrac{B - C_{\max}\left(x^i\right)}{\sum_{k=1}^{m}\left(B - C_{\max}\left(x^i\right)\right)}$
3. End for
4. For k = 1 to n
5. $F_k^{Nei} = \sum_{i=1}^{n} F_k^i = \sum_{i=1}^{n} \left(x_k^i \times q^i\right)$
6. $x_k^{New} = x_k^{Nei} + F_k^{Nei}$
7. End for
8. Output x^{New}
9. Output new sequence based on x^{New}

2.4. Stop Rule:

The stop rule of the EM could be a maximum number of iterative cycles, specified CPU time limit, or maximum number of cycles between two improvements of the global best solution. In this paper, we use a given number of iterative cycles as the stopping criterion. Therefore, in our experiment setting, the algorithm will terminate when a given number of cycles has been executed. This loop is executed for Itemax = 2000 iterations.

3. Result Analysis:

In this section we are going to compare the proposed hybrid electromagnetism (referred to as EM) with one of the recent heuristic. For comparison study we consider the tabu search algorithm by Eksioglu *et al*., (2008), which will be referred to as TS. The platform of our experiments is a personal computer with a Pentium-III 1.2 Hz CPU and 256 MB RAM. The programs are coded in MATLAB. All algorithms are compared using different problem sizes (n=10, 20, 30, 40, 50, 100, 200 and m=5, 10, 15, 20). For each class of the problem defined by given (n, m), 10 instances of problem are randomly generated. Thus we obtain a total of 280 problem instances. Processing time and setup time are given from Uniform random U(1, 99) and U(1, 9) discrete distributions respectively. The numerical results are averaged through each ten instances.
The results of two algorithms are shown in Table 1.

4. Conclusions:

In this research we presented one effective meta-heuristic approach to solve sequence dependent scheduling problems. The main concept of the meta-heuristic approach was electromagnetism that hybridized with one effective initialization phase, a well-known acceptance rule based on simulated annealing and also some neighbourhood search. The purpose of this hybrid method is to take advantage of the EM algorithm, SA algorithm and local search.

Computational results demonstrate the superiority of proposed method compared to one of the strong algorithm recently developed. It is noticeable when the differences between EM and TS are considered, most of them are also significant in the level $\alpha = 0.05$. It demonstrates the significant strength of EM to solve scheduling problems.

Table 1:

Class of problem	n	m	Ave. MS or ($\overline{X}$)		Ave. SD or (S)		T	υ	t	Sig.
			EM	TS	EM	TS				
1	10	5	756.18	764.07	2.40	3.35	2.79	18	1.73	Yes
2	10	10	1103.52	1115.84	1.14	2.61	3.13	15	1.75	Yes
3	10	15	1294.44	1291.02	3.96	2.86	-1.45	15	1.75	No
4	10	20	1605.11	1610.06	3.73	4.59	1.28	18	1.73	No
5	20	5	1342.25	1336.09	2.57	3.31	1.14	18	1.73	No
6	20	10	1576.67	1573.20	5.46	4.61	-1.30	17	1.74	No
7	20	15	1860.67	1880.68	2.23	2.71	1.81	18	1.73	Yes
8	20	20	2137.42	2163.87	4.12	4.49	2.23	18	1.73	Yes
9	30	5	1854.57	1875.67	2.43	5.46	1.96	13	1.77	Yes
10	30	10	2157.03	2180.58	4.27	5.41	1.70	18	1.73	No
11	30	15	2441.80	2462.76	1.57	4.40	3.59	12	1.78	Yes
12	30	20	2706.63	2727.53	4.76	5.55	1.96	18	1.73	Yes
13	40	5	2413.06	2455.28	4.01	5.16	3.18	17	1.74	Yes
14	40	10	2701.31	2696.39	6.82	3.89	-1.89	14	1.76	Yes
15	40	15	2963.04	2991.95	1.90	6.50	2.90	11	1.80	Yes
16	40	20	3225.97	3265.87	2.40	6.72	3.51	12	1.78	Yes
17	50	5	2985.21	2983.77	5.35	5.98	-1.42	18	1.73	No
18	50	10	3259.87	3271.28	5.38	7.81	2.04	17	1.74	Yes
19	50	15	3481.64	3521.68	1.93	9.78	3.40	10	1.81	Yes
20	50	20	3799.89	3822.99	6.52	7.51	2.46	18	1.73	Yes
21	100	5	5735.71	5762.44	1.08	13.13	3.19	9	1.83	Yes
22	100	10	5981.54	6061.89	5.94	11.00	2.45	14	1.76	Yes
23	100	15	6228.77	6290.56	12.07	15.36	0.88	17	1.74	No
24	100	20	6586.27	6652.42	9.28	11.96	1.91	17	1.74	Yes
25	200	5	11201.38	11283.13	14.42	19.17	2.49	17	1.74	Yes
26	200	10	11607.62	11676.69	22.80	23.87	0.37	18	1.73	No
27	200	15	11832.52	11832.11	23.30	20.05	-0.20	18	1.73	No
28	200	20	12285.45	12340.45	21.34	24.99	2.02	18	1.73	Yes
Ave:Average, MS:Makespan, SD:Standard deviation, Sig:Significant										
Each class contains 10 independent instances										

Table 1 demonstrates that EM is better than TS and also many differences in the level $\alpha = 0.05$ are significant.

REFERENCES

Allahverdi, A., CT. Ng, TCE. Cheng, MY. Kovalyov, 2008. A survey of scheduling problems with setup times or costs. European Journal of Operational Research, 187: 985-1032.

Bean, JC., 1994. Genetic Algorithms and Random Keys for Sequencing and Optimization, ORSA Journal on Computing, 6(2): 154-160.

Birbil, SI., SC. Fang, 2003. An Electromagnetism-like Mechanism for Global Optimization. Journal of Global Optimization, 25: 263-282.

Birbil, SI., O. Feyzioglu, 2003. A global optimization method for solving fuzzy relation equations. Lecture Notes in Artificial Intelligence, 2715: 718-724.

Campbell, HG., RA. Dudek, ML. Smith, 1970. A heuristic algorithm for the n job, m machine sequencing problem. Management Science, 16(10): B630-B637.

Chang, PC., SS. Chen, CY. Fan, 2009. A Hybrid Electromagnetism-Like Algorithm for Single Machine Scheduling Problem, Expert Systems with Applications, 36(2): 1259-1267.

Debels, D., De B. Reyck, R. Leus, M. Vanhoucke, 2006. A hybrid scatter search/electromagnetism meta-heuristic for project scheduling. European Journal of Operational Research, 169: 638-653.

Ekşioğlu, B., SD. Ekşioğlu, P. Jain, 2008. A tabu search algorithm for the flowshop scheduling problem with changing neighborhoods. Computers & Industrial Engineering, 54: 1-11.

Garey, MR., DS. Johnson, R. Sethi, 1976. The complexity of flowshop and jobshop scheduling, Mathematics of Operations Research, 1(2): 117-129.

Gupta, JND., 1986. Flowshop schedules with sequence dependent setup times. Journal of the Operations Research Society of Japan, 29(3): 206-219.

Gupta, JND., WP. Darrow, 1986. The two-machine sequence dependent flowshop scheduling problem. European Journal of Operational Research, 24(3): 439-446.

Gupta, SR., JS. Smith, 2005. (in press) Algorithms for single machine total tardiness scheduling with sequence dependent setups. European Journal of Operational Research, doi:10.1016/j.ejor.

SM. Johnson, 1954. Optimal two- and three-stage production schedules with setup times included. Naval Research Logistics Quarterly, 1: 61-68.

Nawaz, M., Jr E.E. Enscore, I. Ham, 1983. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. OMEGA, The International Journal of Management Science, 11(1): 91-95.

Norman, B.A., J.C. Bean, 1999. A Genetic Algorithm Methodology for Complex Scheduling Problems, Naval Research Logistics, 46(2): 199-211.

Osman, I.H., C.N. Potts, 1989. Simulated annealing for permutation flow-shop scheduling. OMEGA, The International Journal of Management Science, 17(6): 551-557.

Reeves, C.R., 1995. A genetic algorithm for flowshop sequencing. Computers & Operations Research, 22(1): 5-13.

Ríos-Mercado, R.Z., J.F. Bard, 1999a. A branch-and-bound algorithm for permutation flow shops with sequence-dependent setup times. IIE Transactions, 31: 721-731.

Ríos-Mercado, R.Z., J.F. Bard, 1999b. An enhanced TSP-based heuristic for makespan minimization in a flow shop with setup times. Journal of Heuristics, 5: 53-70.

Ruiz, R., C. Maroto, 2004. A comprehensive review and evaluation of permutation flowshop heuristics. European Journal of Operational Research (to appear).

Ruiz, R., C. Maroto, J. Alcaraz, 2005. Solving the flowshop scheduling problem with sequence dependent setup times using advanced metaheuristics. European Journal of Operational Research, 165: 34-54.

Ruiz, R., Stutzle. T. 2008. An iterated greedy heuristic for the sequence dependent setup times flowshop with makespan and weighted tardiness1 objectives. European Journal of Operational Research, 87(3): 1143-1159.

Snyder, L.V., M.S. Daskin, 2006. A Random-Key Genetic Algorithm for the Generalized Traveling Salesman Problem, European Journal of Operational Research, 174(1): 38-53

Tasgetiren, M.F., M. Sevkli, Y.C. Liang, 2008. Gencyilmaz, G. Forthcoming, Particle Swarm Optimization Algorithm for Makespan and Total Flowtime Minimization in Permutation Flowshop Sequencing Problem, Accepted to the EJOR Special Issue on Evolutionary and Meta-Heuristic Scheduling by European Journal of Operational Research.

Widmer, M., A. Hertz, 1989. A new heuristic method for the flow shop sequencing problem. European Journal of Operational Research, 41: 186-193.

Wu, P., K.J. Yang, Y.Y. Hung, 2005. The study of electromagnetism-like mechanism based fuzzy neural network for learning fuzzy if-then rules. Lecture Notes in Computer Science, 3684: 382-388.