Research Journal of Applied Sciences 7 (9-12): 426-434, 2012
ISSN: 1815-932X

Approaches to Solving of Permutation Flow Shop Scheduling Problem: An Exploration Study

Ivan Lazar
Faculty of Manufacturing Technologies, Technical University of Kosice,
Bayerova 1, 080 01 Presov, Slovakia

Abstract: Permutation flow shop scheduling problems present an important class of sequencing problems in the realm of production planning. The study briefly reviews typical categories of the PFSSPs in accordance with a defined classification scheme of the particular problems. Subsequently, a review of frequent approaches and methods for PFSSP solving is treated. A final section of the study summarizes findings of this review that is mapping the field over field over the last 25 years.

Key words: Flow shop, PFSSP, mapping, realm, production

INTRODUCTION

With the advent of just in time manufacturing philosophy which maintains a limited in process inventory, the flow shop scheduling problem with minimum make span and optimization approaches to minimize manufacturing cost started to be intensively studied (Modrak and Moskvich, 2011). Flow shop scheduling problems present an important class of sequencing problems in the field of production planning. Solving this problem means finding a permutation of jobs to be processed sequentially on a number of machines under the restriction that the processing of each job has to be with respect to the objective of minimizing the total processing time, i.e., flow time (Sule, 1982). The Permutation Flow Shop Scheduling Problem (PFSP) is often designed by the symbols $n|m|P|C_{max}$ where n jobs have to be processed on m machines in the same order. The processing of each job on each machine is an operation which requires the exclusive use of the machine for an uninterrupted duration called the processing time. The P indicates that only permutation schedules are considered where the order in which each machine processes the jobs is identical for all machines. Hence, a schedule is uniquely represented by a permutation of jobs. The common objective is to find a schedule that minimizes the makespan C_{max} the time at which the last job is completed on the last machine. In a statistical review of flow shop scheduling research, Reisman concluded that there is lack of relevance to practice for the overall majority of research in this field. They emphasize that flow shop scheduling research is in dire need of paradigm shift to enhance its probability of ever becoming a tool for the practice (Bucki and Chramcov, 2011).

Complexity theory provides a mathematical framework in which computational problems are studied so that they can be classified as easy or hard (Brucker, 1998). For the pure flow shop problem there are generally $(n!)^m$ different sequencing alternatives. However, for the PFSP the search space is reduced to n! because it considers the same order of processing all the jobs in all machines. Consequentially, the n-job m-machine PFSPs belong to the class of NP-hard problems (Lenstra *et al*., 1997) Thus, in a PFSP the computational requirements for obtaining an optimal solution increase exponentially as problem size increases. Nevertheless, it is well known that the case of the PFSP composed of two machines (F2 ||Cmax) could be easily solved using Johnson's rule which generates an optimal schedule in $O(n\times\log(n))$ time (Johnson, 1954a; Carlier and Rebai, 1996). However, for $m\geq 3$, the problem is shown to be strongly NP-hard (Garey *et al*., 1976).

Definition of PFSSP: A permutation flow shop scheduling is a production planning process consisting of a set $J = \{J_1, J_2, ..., J_n\}$ of n jobs to be executed in a set of m machines. In this process every job J_j is composed by m-stages $O_{1,j}, O_{2,j}, ..., O_{m,j}$ named operations. Every operation $O_{i,j}$ has a non-negative processing time $t_{i,j}$ composing the matrix $T = \in R^{+}_{MxJ}$. The job operation $O_{i,j}$ must be only executed on machine i. A machine cannot execute more than operation per time. Operation $O_{i,j}$ can be executed only after operation $O_{i-1,j}$ have finished. Preemption is not allowed, i.e., once an operation is started it must be completed without interruption. All jobs must be executed in the same order on every machine defined by a permutation π: $\{1, ..., n\} \rightarrow J$ with π (i) indicating the i-th job to be executed. The completion time of an operation $O_{i,j}$ denoted by $C_{i,j}$ is defined by recurrence:

$$C_{i,\pi(j)} = \begin{cases} t_{i,\pi(j)}{}' \\ C_{i,\pi(j-1)} + t_{i,\pi(j)}{}'' \\ C_{i-1,\pi(j)} + t_{i,\pi(j)}{}''' \\ \max\left(C_{i\,\pi(j-1)}, C_{i-1,\pi(j)} + t_{i,\pi(j)}{}''''\right) \end{cases}$$

$$\text{if } i = 1 \text{ and } j = 1, '' \text{if } i = 1 \text{ and } j > 1, ''' \text{if } i > 1 \text{ and } j = 1 '''' \text{if } i > 1 \text{ and } j > 1$$

Notationally, the problem is referenced by F/permu/Cmax considering as objective function to be minimized the overall processing time the makespan. An example of a permutation flow shop problem schedule is shown in the Fig. 1.

The completion time of a job J_j is $C_{m,j}$. The makespan of permutation is the maximum completion time of a job (Salmasi *et al*., 2010). The objective of Permutation Flow Shop scheduling problem (PFS) is to find a permutation π that minimizes the makespan (Dannenbring, 1977). For any n-jobs sequence S, the Makespan M(S) can be expressed by:

$$M(S) = \sum_{j}^{n} = 1 p_{mj} + \sum_{j=0}^{n-1} X_{[j][j+1]}^{m},$$

Where:

p_{mj} = Processing time on the last machine m of the job in the j th position of S

$\sum_{j=0}^{n-1} X_{[j][j+1]}^{m}$ = The idle time on the last machine m between the end of the job in position j and the start of the job in position (j+1) of S and j = 0 means a dummy job with zero processing times which is always before the first job in S (Moccellin, 1995)

This problem can be described as follows. Each job j_i (i = 1, 2, ..., n) has to be processed on m-machines M_j (j = 1,..., m) following the same order in all machines.

The processing time of job j_i on machine M_j is p_{ij}. To solve such a problem researchers have to consider many constraints as:

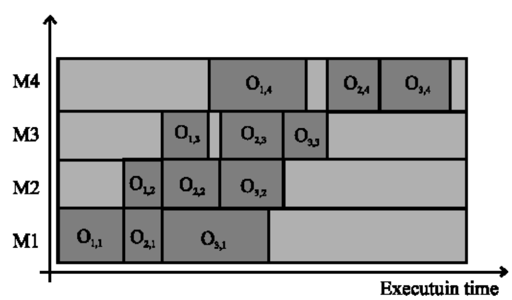

Fig. 1: Example of a permutation flow shop problem schedule

- All jobs are ready for processing at time zero
- The machines are continuously available from time zero onwards (no breakdowns)
- At any time each machine can process at most one job and each job can be processed on at most one machine
- No pre-emption is allowed (that is once the processing of a job on a machine has started, it must be completed without interruption)
- Only permutation schedules are allowed, i.e., all jobs have the same ordering sequence on all machines

MATERIALS AND METHODS

Usually quantitative research is based on a large representative survey and its outcomes are reliable data that can be generalized. As it has been mentioned earlier the survey of the permutation flow shop scheduling problem is aimed to map the field over the last 25 years and is focused:

- To quantify of research effort in developing a wide range of approaches and methods for solving PFSSP
- And to determine which of these methods are most commonly used in solving this problem

In this quantitative research, especially scientific articles registered in scientific databases ACM, science direct and scirus and others were used. Key words applied in search engines were: permutation flow shop, flow shop, scheduling problems, heuristics and metaheuristics methods, genetic algorithm, tabu search, local search, linear, dynamic programming, etc.

Classification of flow shop scheduling problems: Flow shops may be classified into two major categories, non-cyclic and cyclic. These two categories can be broken down into no wait, blocking, limited and unlimited buffer flow shops. Further, classification of flow shop scheduling problems can be done based on the manner in which the job processing logic is affected.

Non-cyclic shop scheduling: For non-cyclic flow shop scheduling is typical that a set of jobs is required to be scheduled in a planning horizon to optimize certain performance measures such as makespan (Cmax) mean tardiness or mean lateness, etc. The job set can be changed for the next planning horizon (Graham *et al*., 1979).

Cyclic shop scheduling: Since, the early 1980s, the diffusion of Just In Time (JIT) among manufacturing companies caused that products of several types have to be produced in proportion of their demand. Consequently, orientation in modern manufacturing has shifted towards cyclic scheduling where the smallest job set that satisfies the respective demand rates of various products is produced cyclically (Wittrock, 1985).

No-wait flow shops: In these shops each job must be processed from start to without any interruption on or between machines (examples are: Fm|no-wait|Cmax, etc.).

Blocking flow shops: In blocking flow shops, there is no buffer between the machines. Partially finished jobs cannot leave the machine on which they are processed unless a free machine is available downstream (examples are: F2|blocking|Ct, Fm|blocking|Cmax, etc.).

Unlimited buffer flow shops: In these shops, unlimited partially finished jobs can temporarily stay in buffers between the machines (examples are: Fm||Ct, F2||Cmax, etc.).

Limited buffer flow shops: The objective is to schedule the given jobs on the given machines such that the mean job flow time is minimized.

Job Scheduling (JS), (simple scheduling): Here, a set of n jobs each having same routing required to be processed in a planning horizon is available at time zero. Job processing times on a machine may differ among jobs. Processing of jobs may involve minor setup changes between jobs. Generally the setup times are included in processing times (Sethi *et al.*, 1992, 1999).

Group Scheduling (GS) (two level scheduling): Group scheduling identifies a sequence of groups as well as the sequence of parts in each group that together minimize some measure of the shops effectiveness. In group scheduling, unlike in conventional job scheduling, issues in scheduling are resolved at two levels. First, for each group a sequence for parts in the group that optimizes some measure of effectiveness is determined. Next, a sequence for the groups themselves that optimizes the same measure of effectiveness must be determined. Machine setups are required in changing processing from one group to another. Note that any needed grouping of parts is done using the concept of group technology before solving the group scheduling problem (Hyer, 1987).

Lot Streaming (LS), (Mass produce scheduling): In this production environment, many products are required to be scheduled. Identical units of a product are often grouped into production lots. A lot is transferred from one machine to the next only when all the items of the lot have been processed. Here a lot is split into several sub-lots so that their operations may be overlapped. This is called lot streaming. This procedure reduces idle times on subsequent machines. Machine setups may be required to change from one product to another.

In the taxonomy used in this study, a flow shop has only one machine in each processing stage whereas in a flexible flow shop each processing stage may have one or more identical parallel machines. Figure 2 shows the manner in which different flow shops may be classified.

The dotted lines cutting across the tree in Fig. 2 separate out the three key aspects that make each flow shop problem distinct in view of its machine environment

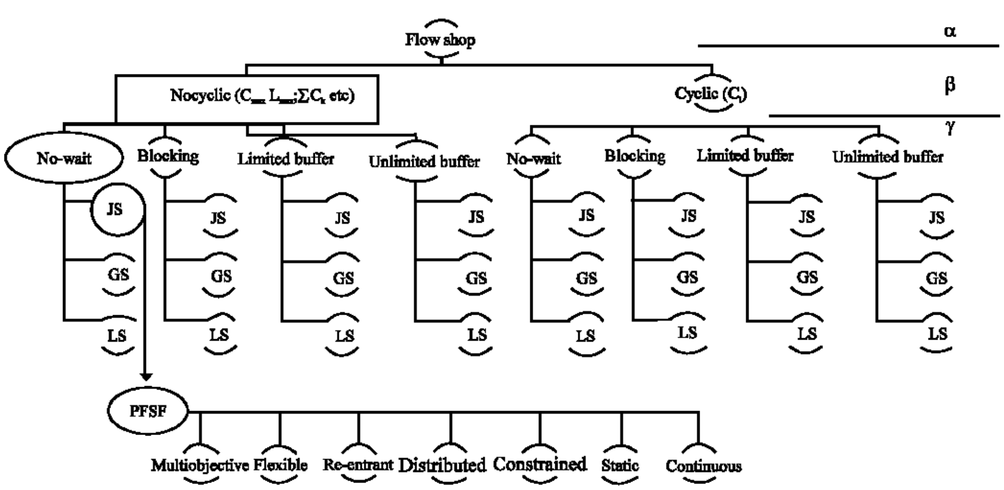

Fig. 2: Flow shop classification (Bagchi *et al.*, 2006)

(α), details of its processing characteristics (β) and the decision objective that is to be optimized (γ). By tracing the branches of this tree a particular flow shop scheduling problem may easily be described as $\alpha/\beta/\gamma$ as noted earlier. For example, an m-machine flow shop (designated by α = Fm) with Lot Streaming (LS) and limited buffer of size β between each machine (designated by β = LS, b) and non cyclic operation with C_{max} minimization as the objective (designated by $\gamma = C_{max}$) will be described by the triplet Fm/LS, b/C_{max}.

RESULTS AND DISCUSSION

Approach and method to solve PFSSP: As it has been stated earlier PFSS problem is NP-hard only some special cases can be solved exactly (Johnson, 1954b). Thus, many approaches have been proposed the find near optimal schedules in reasonable time. Currently available algorithms can be classified as either constructive or improvement methods. Classification of most of methods for PFSSP solving is shown on Fig. 3. Mapping development of these methods presents intensive and extensive studies and therefore in this study is not possible to analyze all of them but vice versa only few of them which accelerated development efforts in given domain.

Among the constructive heuristics are approaches proposed for example by Dannenbring (1997). NEH constructive heuristic seems to be the best performing for a wide variety of problem instances. Improvement approaches are descending local searches (Dannenbring, 1997) and metaheuristics like simulated annealing, Tabu Search (Taillard, 1993; Tailland, 1990; Widmer and Hertz, 1989; Nowicki, 1993). A path algorithm approach based on a particular neighborhood structure was presented in 1993. Stutzle and Hoos (1997) in his study proposed an Ant Colony Optimization (ACO) approach to the flow shop problem. The seminal work on ACO is known under term Ant System (Dorigo and Gambardella, 1997) that was first proposed for solving the Traveling Salesman Problem (TSP).

The performance of ACO approaches with respect to solution quality and convergence speed was enhanced by adding a local search phase (Stutzle and Hoos, 1997; Dorigo and Gambardella, 1997) in which ants are allowed to improve their solutions by a local search procedure.

Obviously, Genetic Algorithms (GA) and Hybrid Genetic Algorithms (HGA) are one among important methods of pushing the development of PFSSP solving. This heuristic is routinely used to generate satisfactory solutions to optimization and search problems. In order to cover wide areas of research in this field and identify decisive development trends, the following Table 1 shows selected literature sources from approximately 100 research articles in this realm.

A general description of the pertinent findings obtained from the presented survey is possible to demonstrate by the following graphs. In accordance with the methodology study and the first objective of this study, one can say that the number of articles dealing with the PFSSP began rapidly to expand starting from 2005 (Fig. 4). This can be perceived as evidence that PFSSP is

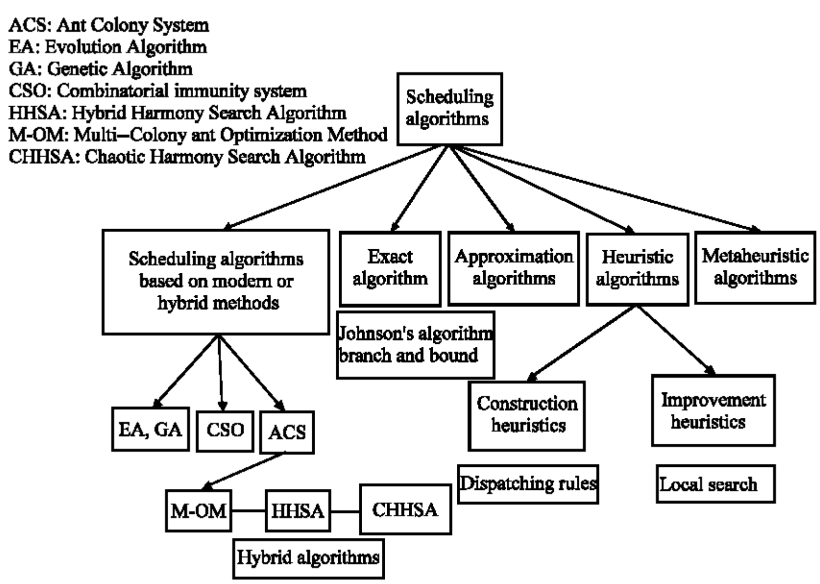

Fig. 3: Algorithms classification

Table 1: Survey of the PFSS problem approaches

Resaerchers	Methods used*	Problem size	Research focus
Moccellin (1995)	H	50×30	N-A
Yamada and Reevesm (1998)	LS, MSXF-GA	20×200	Makespan
Stutzle and Hoos (1997)	ACS	20×100	Makespan
Wang (1997)	LS	N-A	Makespan
Suliman (2000)	H	5×11	Makespan
Fink and Vob (2001)	MH, TS	20×200	TPT
Rajendran and Hans (2004)	ACS	20×100	Makespan
Nowicki (1993)	AA	N-A	CPT
Framinan *et al.* (2004)	H	N-A	Makespan
Duda (2006)	LS, MH	20×200	Makespan
Wang and Shen (2007)	PTA	N-A	TWPT
Engin and Doyen (2007)	AIS	4×4	Makespan
Jarboui *et al.* (2007)	CSO	20×200	Makespan
Seda (2007)	MM	N-A	Makespan
Chen *et al.* (2008)	HTS	3×60	Makespan
Naderi and Ruiz (2008)	H	4×16)	Makespan
Zobolas *et al.* (2009)	MH	20×500	Makespan
Nagarajan and Sviridenko (2009)	TB	N-A	Makespan
Xiaofeng and Zhao (2009)	Greedy Alg.	10×20	Makespan
Chen *et al.* (2008)	HGA	3×50	Makespan
Sun *et al.* (2010)	GA	15×15	Makespan
Chakraborty and Laha (2007)	AbA	N-A	TFT
Modrak and Pandian (2010)	H	7×8	Makespan
Wang *et. al.* (2011)	EA	N-A	Makespan
Rabanimotlagh (2011)	ACS	20×20	Makespan
Tsung-Che *et al.* (2011)	MA	4×4	TFT
Gao and Chen (2011)	H	20×500	Makespan
Khodadadi (2011)	MH	5×9	TrT
Haourai and Ladhari (2003)	BandB	N-A	CT
Yalaoui *et al.* (2010)	M-OM	N-A	Makespan
Minella *et al.* (2011)	MH	N-A	Makespan
Mehta *et al.* (2011)	LS	N-A	Makespan
Pan and Ruiz (2011)	DA	N-A	Makespan
Pan *et al.* (2011)	CHHSA	N-A	Makespan
Tzeng and Chen (2011)	EDA	N-A	Makespan
Bao *et al.* (2011)	GA	N-A	Makespan
Hamdi and Loukil (2011)	GA	N-A	Makespan
Poggi and Sotelo (2012)	AA	N-A	Makespan
Guanlong and Xingsheng (2012)	EA	N-A	Makespan
Shanthikumar and Wu (1985)	MH	N-A	Makespan
Semanco and Modrak (2011)	GA	75×20	Makespan

*Methods: H: Heuristics; LS: Local Search; ACS: Ant Colony System; MH: Metaheuristics; TS: Tabu Search; AA: Approximation Algorithm; PTA: Polynomial Time Algorithm; EA: Evolution Algorithm; AbA: Annealing Based Algorithm; MA: Memetic Algorithm; HTS: Hybrid Tabu Search; CSO: Combinatorial immunity system; HHSA: Hybrid Harmony Search Algorithm; M-OM-Multi-Colony ant Optimization Method; TFT: Total Flow Time; CPT: Controllable Processing Time; TWPT: Total Weighted Processing Time; TrT: Total reaction Time; CHHSA: Chaotic Harmony Search Algorithm

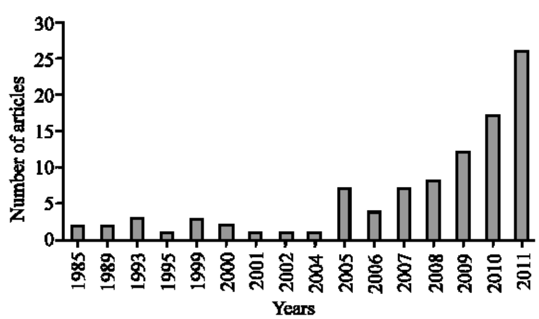

Fig. 4: Frequency of research articles dealing with the PFSSP over the last 25 years

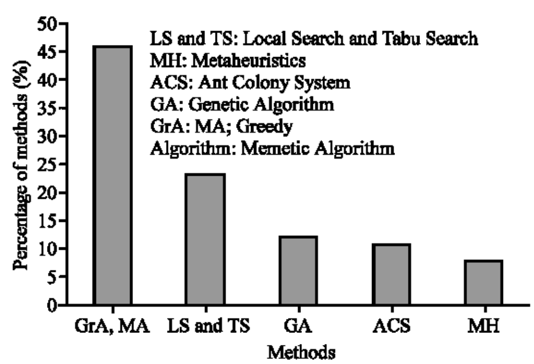

Fig. 5: The most commonly used methods of PFSSP in order of frequency

quite popular research topic which is dealt by increasing number of researchers. In the next graph in Fig. 5 are shown as consistent with the second objective of this study, methods that are most commonly used in solving this problem in order of frequency. The topicality of greedy algorithm seems to be beyond a reasonable doubt due to computational results demonstrating the superiority in terms efficiency and effectiveness (Pan *et al*., 2007). As regards to Memetic Algorithms (MA) they represent one of the recent growing areas of research in evolutionary computation and are widely used as a synergy of evolutionary or any population based approach with separate individual learning or local improvement procedures for problem search. One of the several approaches that may be useful for the PFSSP with the objective to minimize the maximum completion time is Particle Swarm Optimization (PSO) based Memetic Algorithm (MA). In the PSO based MA algorithm both PSO based searching operators and some special local searching operators are employed to balance the exploration and exploitation abilities. In particular, this algorithm applies the evolutionary searching mechanism of PSO which is characterized by individual improvement, population cooperation and competition to effectively perform exploration (Liu *et al*., 2007). The second important group of algorithms includes well-known tabu search and local search. Iterative LA and TS are powerful optimization procedures that have been successfully applied to a number of PFSSPs.

CONCLUSION

Based on the results of presented study can be stated that Permutation Flow Shop Scheduling Problem (PFSSP) is one of the topical problems in operations research which is continuously being updated in accordance with the results of newest approaches.

The intention of this research was to provide an overview of one class of a large group of flow shop scheduling problem. In each its part (especially in the earlier studies) is offered a brief literature of researches that dealt with the particular approaches. A part of the main objectives of this study, considerable attention has been paid to the concept of classification of flow shop problems and algorithms classification that are pertinent to solve specified problems.

The results from the mapping of important trends in developing new methods can be used as a reference for future research needs of improving and developing better approaches to permutation flow shop related scheduling problems.

In order to bridge theoretical approaches to flow shop scheduling problem solving and practical techniques in manufacturing companies it will be necessary to analyze generic features of current shop floor's production planning and scheduling. Production scheduling must consider apart from basic scheduling criterion usually used in theoretical models a lot of further factors such as urgent order, production ability and so on. With increasing complexity of production tasks in present manufacturing environment, the workshop scheduling becomes more challenging and call for deeper collaboration between APS and MES solutions (Modrak and Mandulak, 2009; Zhongyi *et al*., 2011). An other issue how to bridge theoretical approaches and practical needs is to differ between theoretical operations modeling and practical needs by Gant diagrams from a time viewpoint (Modrak and Modrak, 2009). An example of the different approaches to operations modeling is shown in Fig. 6.

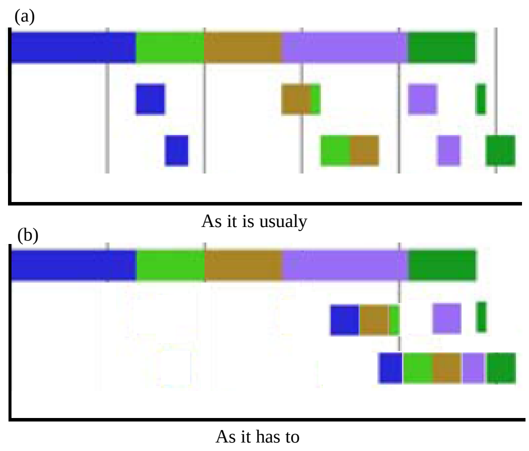

Fig. 6: Different approaches to operations modeling

An interaction between so called group scheduling and the production flow analysis also seems to be a topical subject of debate. Several related issues on this topic are discussed, e.g. by Modrak and Pandian (2010) and Semanco *et al*. (2011).

A final remark is addressed to the adaptability of scheduling approaches. Dynamic adaptation in PFSS practical problems become increasingly important as planning must take into account changing data (e.g., new jobs). In this respect, there are many vital approaches as for example local search methods (Fink and Vob, 2001) that provide the means of adaptively changing schedules on the basis of existing plans.

REFERENCES

Bagchi, T.P., J.N.D. Gupta and C. Sriskandarajah, 2006. A review of TSP based approaches for flow shop scheduling. Eur. J. Operat. Res., 169: 816-854.

Bao, Y., H. Jing and L. Zheng, 2011. A hybrid based GA and HS algorithms in blocking flow shop scheduling problems. Proceedings of the 2nd International Conference on Mechanic Automation and Control Engineering, July 15-17, 2011, Hohhot, pp: 953-956.

Brucker, P., 1998. Scheduling Algorithms. 2nd Edn., Springer, Heidelberg, USA.

Bucki, R. and B. Chramcov, 2011. Modelling and simulation of the order realization in the serial production system. Int. J. Math. Models Methods Applied Sci., 5: 1233-1240.

Carlier, J. and I. Rebai, 1996. Two branch-and-bound algorithms for the permutation flow shop problem. Eur. J. Operat. Res., 90: 238-251.

Chakraborty, U.K. and D. Laha, 2007. An improved heuristic for permutation flow shop scheduling. Int. J. Inform. Commun. Technol., 1: 89-97.

Chen, J.S., J.C.H. Pan and C.K. Wu, 2008. Hybrid tabu search for re-entrant permutation flow shop scheduling problem. Expert Syst. Appl., 34: 1924-1930.

Dannenbring, D.G., 1997. An evaluation of flow shop sequencing heuristics. Manage. Sci., 23: 1174-1182.

Dorigo, M. and L.M. Gambardella, 1997. Ant colonies for the traveling salesman problem. Bio Systems, 43: 73-81.

Duda, J., 2006. Local search and nature based metaheuristics: A case of flow shop scheduling problem. Proceedings of the International Multiconference on Computer Science and Information Technology, November 6-10, 2006, Wisla, Poland, pp: 17-24.

Engin, O. and A. Doyen, 2007. A new approach to solve flow shop scheduling problems by artificial immune systems. Dogus Universitesi Dergisi, 8: 12-27.

Fink, A. and S. Vob, 2001. Solving the continuous flow shop scheduling problem by meta-heuristics. Eur. J. Operat. Res., 151: 400-414.

Framinan, J.M., J.N.D. Gupta and R. Leisten, 2004. A review and classification of heuristics for permutation flow shop scheduling with makespan objective. J. Operat. Res. Soc., 55: 1243-1255.

Gao, J. and R. Chen, 2011. An NEH-based heuristic algorithm for distributed permutation flow shop scheduling problems. Scient. Res. Essays, 6: 3094-3100.

Garey, M.R., D.S. Johnson and R. Sethi, 1976. The complexity of flow shop and job shop scheduleing. Math. Operat. Res., 1: 117-129.

Graham, R.L., E.L. Lawler, J.K. Lenstra and A.H.G. Rinnooy Kan, 1979. Optimization and approximation in deterministic sequencing and scheduling theory: A survey. Ann. Discrete Math., 5: 287-326.

Guanlong, D. and G. Xingsheng, 2012. A hybrid discrete differential evolution algorithm for the no-idle permutation flow shop scheduling problem with makespan criterion. Comput. Operat. Res., 38: 2152-2160.

Hamdi, I. and T. Loukil, 2011. Minimizing the makespan in the permutation flow shop problem with minimal and maximal time lags. Proceedings of the International Conference on Communications, Computing and Control Applications, March 3-5, 2011, Tunisia, pp: 1-6.

Haouari, M. and T. Ladhari, 2003. A branch-and-bound based local search method for the flow shop problem. J. Operat. Res. Soc., 54: 1070-1084.

Hyer, N.L., 1987. Capabilities of Group Technology. 1st Edn., Computer and Automated Systems Association of SME, Dearborn, MI., Pages: 428.

Jarboui, B., S. Ibrahim, P. Siarry and A. Rebai, 2007. A combinatorial particle swarm optimization for solving permutation flow shop problems. Comput. Indus. Eng., 54: 526-538.

Johnson, S.M., 1954b. Optimal two and three stage production schedule with set up times included. Naval Res. Logistics Q., 8: 61-68.

Johnson, S.M., 1954a. Optimal two-and three-stage production schedules with setup times included. Naval Res. Logistics Q., 8: 1-61.

Khodadadi, A., 2011. Solving constrained flow shop scheduling problem with three machines. Int. J. Acad. Res., 3: 38-40.

Lenstra, J.K., A.H.G. Rinnooy Kan and P. Brucker, 1977. Complexity of machine scheduling problems. Ann. Discrete Math., 1: 343-362.

Liu, B., L. Wang and Y.H. Jiu, 2007. An effective PSO-based Memetic Algorithm for flow shop scheduling. IEEE Trans. Syst. Man Cybernetics Part B: Cybernetics, 37: 18-27.

Mehta, P., P. Pandit, D. Philip and P. Sharma, 2011. An efficient local search for minimizing completion time variance in permutation flow shops. Comput. Operat. Res., 39: 1000-1009.

Minella, G., R. Ruiz and M. Ciavotta, 2011. Restarted iterated pareto greedy algorithm for multi-objective flow shop scheduling problems. Comput. Operat. Res., 38: 1521-1533.

Moccellin, J.V., 1995. A new heuristic methods for the permutation flow shop scheduling problem. J. Operat. Res. Soc., 46: 883-886.

Modrak, V. and J. Mandulak, 2009. Mapping development of MES functionalities. Proceedings of the 6th International Conference on Informatics in Control, Automation and Robotics Signal Processing, Systems Modeling and Control, July 2-5, 2009, Milan, Italy, pp: 244-247.

Modrak, V. and J. Modrak, 2009. Relationships between batch sizes, sequencing and lead-times. Proceedings of the 6th International Conference on Informatics in Control, Automation and Robotics Signal Processing, Systems Modeling and Control, July 2-5, 2009, Milan, Italy, pp: 380-383.

Modrak, V. and R.S. Pandian, 2010. Flow shop scheduling algorithm to minimize completion time for n-jobs m-machines problem. Tehnicki Vjesnik, 17: 273-278.

Modrak, V. and V. Moskvich, 2011. Impacts of RFID implementation on cost structure in networked manufacturing. Int. J. Prod. Res., 50: 3847-3859.

Naderi, B. and R. Ruiz, 2008. The distributed permutation flow shop scheduling problem. http://www.upv.es/deioac/Investigacion/Naderi_Ruiz_DPFSP.pdf.

Nagarajan, V. and M. Sviridenko, 2009. Tight bounds for permutation flow shop scheduling. Math. Operat. Res., 34: 417-427.

Nowicki, E., 1993. An approximation algorithm for the m-machine permutation flow shop scheduling problem with controllable processing times. Eur. J. Operat. Res., 70: 342-349.

Pan, Q.K. and R. Ruiz, 2011. An estimation of distribution algorithm for lot-streaming flow shop problems with setup times. Omega, 40: 166-180.

Pan, Q.K., L. Wang and B.H. Zhao, 2007. An improved iterated greedy algorithm for the no-wait ow shop scheduling problem with makespan criterion. The Int. J. Adv. Manufactur. Technol., 38: 778-786.

Pan, Q.K., L. Wang and L. Gao, 2011. A chaotic harmony search algorithm for the flow shop scheduling problem with limited buffers. Applied Soft Comput. J., 11: 5270-5280.

Poggi, M. and D. Sotelo, 2012. A linear time approximation algorithm for permutation flow shop scheduling. Theor. Comput. Sci., 416: 87-94.

Rabanimotlagh, A., 2011. An efficient ant colony optimization algorithm for multiobjective flow shop scheduling problem. World Acad. Sci. Eng. Technol., 51: 127-133.

Rajendran, R.C. and Z. Hans, 2004. Ant-colony algorithms for permutation flow shop scheduling to minimize makespan/total flowtime of jobs. Eur. J. Operat. Res., 155: 426-438.

Salmasi, N., R. Logendran and M.R. Skandari, 2010. Total flow time minimization in a flow shop sequence-dependent group scheduling problem. Comput. Operat. Res., 37: 199-212.

Seda, M., 2007. Mathematical models of flow shop and job shop scheduling problems. World Acad. Sci. Eng. Technol., Vol. 31.

Semanco, P. and V. Modrak, 2011. Hybrid GA-based improvement heuristic with makespan criterion for flow shop scheduling problems. Commun. Comput. Inf. Sci., 220: 11-18.

Semanco, P., V. Modrak, R.S. Pandian and P. Knuth, 2011. Using genetic algorithm for elimination of exceptional elements in the stage of PFA. J. Eng. Applied Sci., 6: 122-126.

Sethi, S.P., C. Sriskandarajah, G. Sorger, J. Blazewicz and W. Kubiak, 1992. Sequencing of parts and robot moves in a robotic cell. Int. J. Flexible Manuf. Syst., 4: 331-358.

Sethi, S.P., C. Sriskandarajah, S. van de Velde, Y. Wang and H. Hoogeveen, 1999. Minimizing makespan in a pallet-constrained flow shop. J. Scheduling, 2: 115-133.

Shanthikumar, J.G. and Y.B. Wu, 1985. Decomposition approaches in permutation scheduling problems with application to the M-machine flow shop scheduling problems. Eur. J. Operat. Res., 19: 125-141.

Stutzle, T. and H. Hoos, 1997. The MAX-MIN Ant System and local search for the traveling salesman problem. Proceedings of the IEEE International Conference on Evolutionary Computation, April 13-16, 1997, Indianapolis, IN., pp: 309-314.

Sule, D.R., 1982. Sequencing n jobs on two machines with setup, processing and removal times separated. Naval Res. Logist. Q., 29: 517-519.

Suliman S.M.A., 2000. A two-phase heuristic approach to the permutation flow shop scheduling problem. In. J. Prod. Econ., 64: 143-152.

Sun, L., X. Cheng and Y. Liang, 2010. Solving job shop scheduling problem using genetic algorithm with penalty function. Int. J. Intell. Inf. Process., 1: 65-77.

Tailland, E., 1990. Some efficient heuristic methods for the flow shop sequencing problems. Eur. J. Operat. Res., 47: 65-74.

Taillard, E., 1993. Benchmarks for basic scheduling problems. Eur. J. Operat. Res., 47: 278-285.

Tsung-Che, C., C. Hsueh-Chien and F. Li-Chen, 2011. NNMA: An effective memetic algorithm for solving multiobjective permutation flow shop scheduling problems. Exp. Syst. Appl., 38: 5986-5999.

Tzeng, Y.R. and C.L. Chen, 2011. A hybrid EDA with ACS for solving permutation flow shop scheduling. Int. J. Adv. Manuf. Technol., 19: 1-9.

Wagner, H.M., 1959. An integer linear-programming model for machine scheduling. Naval Res. Logist. Quart., 6: 131-140.

Wang, B., 1997. Integrated Product, Process and Enterprise Design. 1st Edn., Springer, London, ISBN-10: 0412620200, Pages: 504.

Wang, L. and W. Shen, 2007. Process Planning and Scheduling for Distributed Manufacturing. Springer, London, ISBN: 1846287510, Pages: 429.

Wang, L.P., Q.K. Tasgetiren and M. Fatih, 2011. A hybrid harmony search algorithm for the locking permutation flow shop scheduling problem. Comput. Ind. Eng., 61: 76-83.

Widmer, M. and A. Hertz, 1989. A new heuristic method for the flow shop sequencing problem. Eur. J. Operat. Res., 1: 186-193.

Wittrock, R., 1985. Scheduling algorithms for flexible flow lines. IBM J. Res. Dev., 29: 401-412.

Xiaofeng, L. and H. Zhao, 2009. Greedy algorithm solution of flexible flow shop scheduling problem. IJCSNS, 9: 177-180.

Yalaoui, N., L. Amodeo, F. Yalaoui and H. Mahdi, 2010. Particle swarm optimization under fuzzy logic controller for solving a hybrid reentrant flow shop problem. Proceedings of the IEEE International Symposium on Parallel and Distributed Processing, Workshops and Phd Forum, April 19-23, 2010, Atlanta, GA., pp: 1-6.

Yamada, T. and C.R. Reevesm, 1998. Solving the Csum permutation flow shop scheduling problem by genetic local search. Proceedings of the IEEE International Evolutionary Computation Proceedings, May 4-9, 1998, Anchorage, AK., pp: 230-234.

Ying, K.C. and C.J. Liao, 2004. Ant colony system for permutation flow shop sequencing. Comput. Operat. Res., 31: 791-801.

Zhongyi, M., M. Younus and L. Yongjin, 2011. Automated planning and scheduling system for the composite component manufacturing workshop. Eng. Lett., 19: 75-83.

Zobolas, G.I., C.D. Tarantilis and G. Ioannou, 2009. Minimizing makespan in permutation flow shop scheduling problems using a hybrid metaheuristic algorithm. Comput. Operat. Res., 36: 1249-1267.