A REVIEW OF HEURISTICS FOR PERMUTATION FLOW SHOPS AND FLOW SHOPS WITH SEQUENCE DEPENDENT SETUP TIMES

Anoop A. K.[1], Dr. Rajesh Vanchipura[2]
[1]M. Tech, Dept. of Mechanical Engineering, Govt. Engineering College, Thrissur
[2]Assistant Professor, Dept. of Mechanical Engineering, Govt. Engineering College, Thrissur
Email: anoopak007@yahoo.in[1], rajeshvanchipura@gmail.com[2]

Abstract

This paper presents a review of both the permutation flow shop and the flow shop with Sequence Dependent Setup Times (known as SDST flow shop). The paper focuses on works which involve heuristic approach as a solution methodology. Generally, heuristic approaches are classified into constructive heuristics and improvement heuristics. Improvement heuristics are further classified into neighborhood search heuristics and population based heuristics. The review analyzes the various methodologies used by the researchers and classifies them. Several researches have been conducted on the permutation version of the flow shop scheduling problem and wide variety of heuristics have been developed. However, literature shows that there is limited application of such heuristics on a more realistic variation of the basic flow shop namely, SDST flow shop.

Index Terms: constructive heuristics, improvement heuristics, permutation flow shop, Sequence Dependent Setup Times.

I. INTRODUCTION

Scheduling is the allocation of resources (e.g. machines) to tasks (e.g. jobs) in order to ensure the completion of these tasks in a reasonable amount of time. A flow shop comprises of a number of jobs which need to be processed on a number of machines where each job has the same processing route. The general flow shop scheduling problem involves a set of *n* number of jobs or tasks (1, 2, …, *n*) to be processed on a set of *m* number of machines or processors (1, 2, …, *m*) in the same order, i.e. first on machine 1, then on machine 2 and so on until the machine *m*. Therefore, we assume that the machines are ordered in the order they are visited by each job. Although for the general flow shop, the job sequence may not be the same for every machine, here, the assumption is the permutation flow shop, i.e. the job sequence is the same on every machine.

An important and more realistic variation of the basic flow shop is the flow shop operating in a sequence dependent setup time environment. The objective is to find a sequence for processing the jobs on the machines so that the total completion time or makespan of the schedule is minimized.

II. PROBLEM FORMULATION

A. Assumptions

- Each job is available at time zero.
- Each job can be processed at most on only one machine at a time.
- Each machine is able to process only one job at a time.
- Setup of a machine can be done without the job being available at the machine.
- No pre-emption of job is allowed, i.e., a job cannot be passed to the next machine while it is being processed on a machine.
- The setup times of the jobs on the machines are separable and the job is not necessary to

do the setup.

- The job sequence is the same on every machine and the machines are continuously available.
- All the processing times and setup times are known in advance.
- Setup time is dependent on the sequence in which the jobs are processed.
- The sequence dependent setup time is assumed to be asymmetric; i.e., $S_{ijk} \neq S_{ikj}$.

B. SDST flow shop Formulation

Notations used

m	Total number of machines in the flow shop
n	Total number of jobs to be scheduled
i	Index for the machine; *(i-1)* indicates the previous machine in the sequence, where $i=1,2,\ldots,m$
j	Index for the job; *(j-1)* indicates the previous job processed on the machine, where $j=0,1,\ldots,n$
σ	Ordered set of jobs already scheduled, out of *n* jobs, called the partial sequence
n_σ	Number of jobs in the partial sequence, *σ*
p_{ij}	Processing time of the job *j* on machine *i*
S_{ijk}	Setup time on the machine *i*, when job *k* is preceded by job *j*
q(σ,i)	Completion time of the partial sequence *σ* on machine *i*
q(σj,i)	Completion time of job *j* on machine *i*, when job *j* is appended to the partial sequence *σ*

The completion time of the partial sequence *σj* on machine *i* is determined using the following recursive equation:

$$q(\sigma j, i) = \max\{q(\sigma, i) + S_{ijk}, q(\sigma j, i-1)\} + p_{ij} \qquad (1)$$

where $q(\varphi, i) = 0$ and $q(\sigma, 0) = 0$, for all *σ* and *i*, with *φ* denoting a null schedule. The flow time of job j is given by:

$$C_j = q(\sigma j, m) \qquad (2)$$

The makespan is the total length of the schedule when all the jobs have finished processing. When all the jobs are scheduled, the makespan is given by:

$$M = \max\{C_j, j = 1, 2, \ldots, n\} \qquad (3)$$

Makespan is the maximum of the C_j values of all the jobs, which will be, in effect, the C_j value of the n^{th} job. This makespan value should be the minimum possible.

III. LITERATURE REVIEW

The permutation flow shop scheduling problem (PFSP) was proved to be NP hard by Garey *et al.* [1]. Moreover, when the sequence dependent setup times are also considered, the problem becomes NP complete as shown by Gupta [2], which means that it cannot be solved in polynomial time. Hence, researchers have focussed mainly on the development of heuristics and metaheuristics.

The complexity of the flow shop scheduling problems renders exact solution methods infeasible and impractical for instances of more than a few jobs and machines. This is the main reason why various heuristic methods, which obtain near-optimal solutions, are tried. The heuristic methods can be divided in to constructive heuristics and improvement heuristics. Constructive heuristics are the heuristics that build a feasible schedule from scratch by making a series of passes through the list of unscheduled jobs, where at each pass one or more jobs are selected and improvement heuristics start from an initial solution and apply an improvement procedure (refer Fig.2). In constructive heuristics, a sequence obtained is fixed and cannot be altered i.e., it gives a single solution all the time whereas in improvement heuristics, an initial solution is iteratively improved upon i.e., it may give different solutions every time. Hence the literature review is divided into two sections:

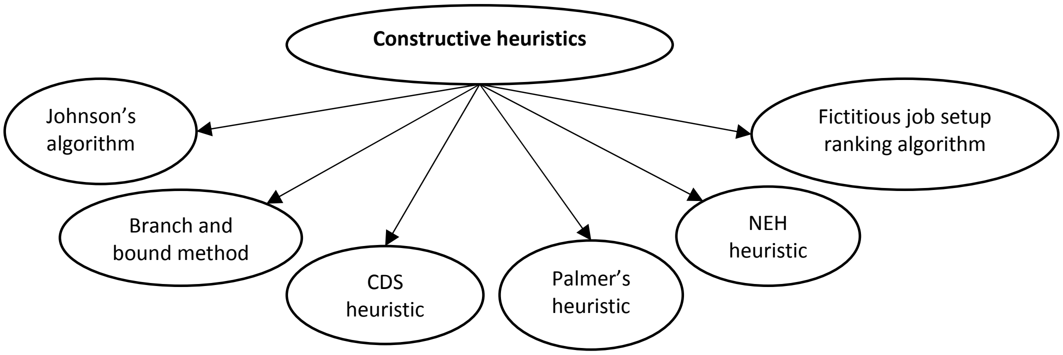

Fig.1. Constructive heuristics for flow shops

A. Constructive heuristics

1. Constructive heuristics for permutation flow shop

Johnson's work was the first significant one on flow shop scheduling [3]. Johnson's algorithm is the earliest known heuristic for the permutation flow shop scheduling problem (PFSP), which provides an optimal solution for two machines. Moreover, it can also be used as a heuristic for the *m* machine problem by clustering the *m* machines into two virtual machines. His pioneering work was the real inspiration for all of the future researches in the flow shop scheduling, in which the general ideas of Johnson's rule have been used.

Another method is assigning a weight or index to every job and then arranging the sequence by sorting the jobs according to the index. This idea was given by Palmer [4]. He developed a simple heuristic, in which a slope index is calculated for every job and then the jobs are arranged in decreasing order of this index. Campbell *et al.* [5] extended Johnson's algorithm to develop a heuristic algorithm (called CDS heuristic). By clustering *m* original machines into two fictitious machines and solving the two machine problem thus generated by repeatedly applying Johnson's algorithm, (*m*-1) schedules are built.

Nawaz, Enscore and Ham's NEH heuristic [6] is regarded as the best heuristic till today for the PFSP. Their idea was that the jobs with highest processing times on all the machines should be scheduled as early in the sequence as possible. The total processing time for each job on all the machines is first calculated. Then the jobs are sorted in the decreasing order of total processing time. The first two jobs are taken and the best schedule is chosen out of the two possible schedules. Next, one job is taken at a time from the sorted list and inserted at every possible position and the best sequence obtained is selected. The process is repeated until the last job is selected from the sorted list.

Chakraborty and Laha [7] modified the NEH heuristic for makespan minimization in PFSP. The modified algorithm was run on 28 different problem sizes. Analysis reveals significant improvement in the quality of the solution while the algorithmic complexity remained the same. Their conclusion was that both original NEH and modified NEH outperform the best competitor to date.

2. Constructive heuristics for SDST flow shop

Rios-Mercado and Bard [8] developed a branch-and-cut algorithm for SDST flow shop which obtained the optimum solutions for instances of up to eight jobs and six machines. The same authors Rios-Mercado and Bard [9] also developed a branch-and-bound method which was able to solve SDST flow shop problem instances of up to ten jobs and six machines with a maximum deviation of about 1% from the optimal solution. Rios-Mercado and Bard [10] also proposed an extension of NEH heuristic for the SDST flow shop and their heuristic is called NEHRB.

Rajesh Vanchipura and R. Sridharan [11] presented two new constructive heuristics for solving the SDST flow shop scheduling problem with the minimization of makespan as the objective. The first heuristic called as the setup ranking algorithm (SRA) generates the sequence using only the setup times of jobs. The second heuristic algorithm, fictitious job setup ranking algorithm (FJSRA), is developed using concept of fictitious jobs. Fictitious jobs are the pairs of jobs with minimum setup time between them.

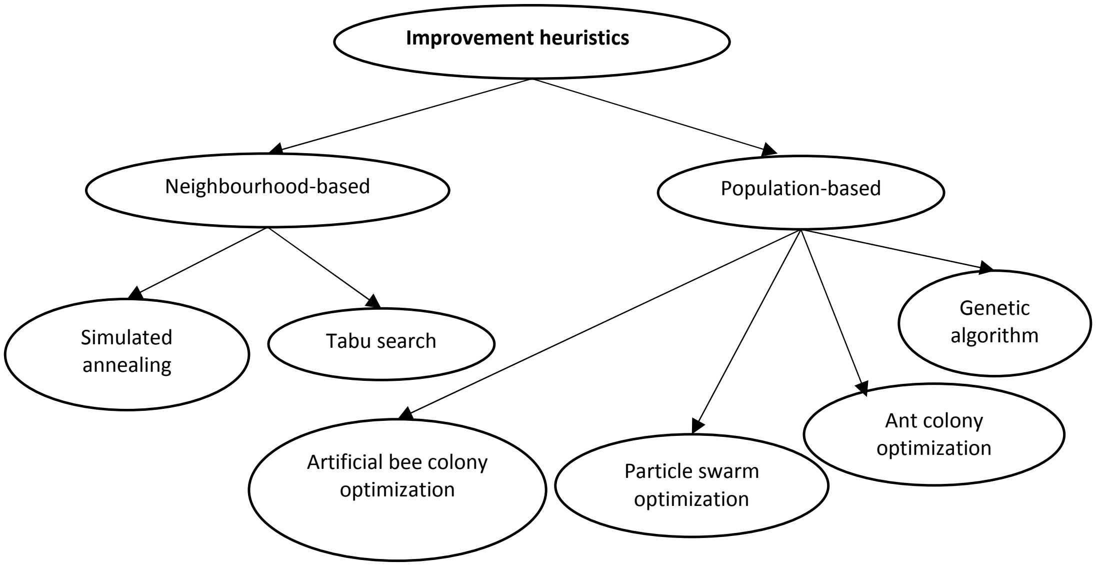

Fig.2. Improvement heuristics for flow shops

A. Improvement heuristics

The improvement heuristics always start with an initial solution and apply some improvement procedure which improves the solution iteratively. Improvement heuristics are invented more than the constructive ones due to their flexibility. These include metaheuristics such as Genetic Algorithm (GA), Ant Colony Optimization (ACO), Bee Colony Optimization, Particle Swarm Optimization (PSO), Tabu Search (TS) and Simulated Annealing (SA) etc. (refer Fig.2). Improvement heuristics have been used to solve both permutation flow shop and SDST flow shop. Hence, the literature of improvement heuristics can be further divided in to:

1. Improvement heuristics for permutation flow shop

Simulated Annealing (SA) algorithm, based on the analogy of annealing process of metals, was worked on by Osman and Potts [12]. They developed a set of four different SA based heuristic algorithms for the PFSP and proved that their algorithm gives better results compared to the NEH heuristic. Rajesh Gangadharan and Chandrasekharan Rajendran [13] considered simulated annealing for permutation flow shop scheduling problem (PFSP) with the twin-objective of minimizing makespan and total flowtime. They proposed two new heuristics to provide the seed sequences for the SA heuristic. Another simulated annealing based heuristic with bi-criteria minimization of makespan and maximum tardiness was developed by Chakravarthy and Rajendran [14].

Chen *et al.* [15] developed a simple Genetic Algorithm (GA) for the PFSP with various improvements. The initial population was developed with CDS heuristic and RA heuristic. Only the crossover operator was used with no mutation and the crossover used was partially mapped crossover or PMX. Reeves [16] also developed a GA in which the offsprings generated do not replace their parents but individuals that have fitness value below average. He used a crossover called C1 or one-point order crossover and used a shift mutation. Two new hybrid genetic algorithms with minimization of makespan as objective has also been proposed by Ruiz *et al*. [17] for PFSP. Their algorithms use new genetic operators, improved techniques like that of the hybridization with a local search and an efficient population initialization and also a new generational scheme.

Nowicki *et al*. [18] proposed a fast tabu search algorithm for finding minimum makespan using a modified NEH algorithm to obtain initial solution. Their algorithm is based on tabu search technique with a specific neighborhood definition which employs a 'block of jobs' notion. Computational experiments of up to 500 jobs and 20 machines show its excellent numerical properties.

Suliman [19] proposed a two-phase improvement heuristic. In the first phase an initial job sequence is generated using one of the available, well-known and efficient heuristics, while the sequence generated is improved in the second phase in terms of makespan using a pair exchange mechanism coupled with directionality constraint. The resulting algorithm is found to have performance comparable to NEH which runs faster.

Ant colony optimization (ACO) is another approach for solving the flow shop problem. Rajendran and Ziegler [20] used two ACO algorithms with the objective of minimizing makespan and total flow time of jobs. The effectiveness of the algorithms was evaluated by considering benchmark problems and values of makespan given by Taillard [25].

Particle swarm optimization (PSO) by Tasgetiren *et al.* [21] is another significant work done on the permutation flow shop scheduling problem. They considered objectives of minimizing makespan and the total flowtime of jobs and applied PSO to the 90 benchmark instances provided by Taillard. Artificial bee colony algorithm (ABC) was worked on by Tasgetiren *et al*. [22] to solve the PFSP. They presented a discrete artificial bee colony algorithm (DABC) hybridized with a variant of the iterated greedy algorithms to find the permutation with the smallest total flowtime

Backtracking Search Algorithm (BSA) is one of the new-born algorithms that was proposed first by Civicioglu [23] and used in continuous numerical optimization problems. It is a kind of evolutionary algorithms (EA) which mimics the natural evolution process. BSA, which is basically used for continuous numerical optimization problems, was applied to the PFSP by Qun Lin *et al*. [24] with makespan criterion.

Taillard [25] gives benchmark problems for job shop, permutation flow shop and open shop scheduling problems. In his paper, he proposed 260 randomly generated scheduling problems whose sizes are greater than sizes of the rare examples published and correspond to real dimensions of industrial problems.

2. *Improvement heuristics for SDST flow shop*

Parthasarathy and Rajendran [26] considered simulated annealing to minimize total weighted tardiness in SDST flow shop. They proposed a perturbation scheme called random insertion perturbation scheme, which resulted in a superiority of about 70% over the existing heuristic. Rios-Mercado and Bard [10] put forward a metaheuristic based on a greedy randomized adaptive search procedure (GRASP) for solving the SDST flow shop.

One of the most important works done on SDST flow shop is the hybrid genetic algorithm developed by Ruiz *et al*. [27]. The hybrid version includes a special initialization of population (a modification of NEH heuristic) and two selection schemes, tournament selection and roulette wheel selection and also a local search. Four types of crossover and a powerful restart scheme were also utilized.

Gajpal *et al.* [28] presented an ant colony optimization algorithm for flow shop scheduling with sequence dependent setups for the makespan objective. A tabu search heuristic with makespan and weighted tardiness criteria for SDST flow shop was developed by Fred Choobineh *et al.* [29]. They showed that proposed heuristic gives optimal or near to optimal solutions in a reasonable time.

Rajesh Vanchipura *et al.* [30] presented the application of variable neighbourhood descent (VND) approach for solving SDST flow shop scheduling problem. They integrated the VND approach with two constructive heuristics, NEHRB and FJSRA. The analysis reveals that the performance of VND-based algorithms depends on the constructive algorithm used for providing the initial solution.

IV. CONCLUSION

From the literature survey, it is clear that some exact approaches for SDST flow shop have been developed, but they are only applicable to two machine case and m machine case can be solved using these methods only for very small instances. Furthermore, the general heuristics presented have high CPU time requirements and hence not suitable for medium and large instances. Moreover, there are very few work considering the SDST flow shop with the makespan criterion. In addition, sequence dependent setup time (SDST) is one of the most recurring additional complications in the scheduling problem. Most of the existing work have incorporated the setup time in the processing time, but to improve the performance of manufacturing system setup time has to be considered separate since it is one of the critical factors. The constructive heuristics are extremely difficult to be formulated for these problems. Only few constructive heuristics are available till now. More researches are done on improvement heuristics such as heuristic algorithms and hybrid metaheuristics in flow shop scheduling. SDST flow shop scheduling is an area where the number of existing researches is limited. Future research on flow shops operating under sequence dependent setup times environment is desirable.

REFERENCES

[1] Garey M.R, Johnson D.S and Sethi R.R. ,"The complexity of flow shop and job shop scheduling", Operations Research. vol 1, pp. 117-129, 1976.

[2] Gupta J.N.D. ," Flow shop schedules with sequence dependent setup times, Journal of the Operations Research Society of Japan", vol 29, no.3, pp. 206-219, 1986.

[3] Johnson S.M. ,"Optimal two-and three-stage production schedules with setup times included", Naval Research Logistics Quarterly, vol. 1, pp. 61-68, 1954.

[4] Palmer D.S. ,"Sequencing jobs through a multi-stage process in the minimum total time a quick method of obtaining a near optimum", Operations Research Quarterly, vol. 16, no. 1, pp. 101-107, 1965.

[5] Campbell H.G, Dudek R.A, Smith M.L. ,"A heuristic algorithm for the n-job m-machine sequencing problem", Management Science vol.16, no. 10, pp. 169-174, 1970.

[6] Nawaz M., Enscore E., Ham I., "A heuristic algorithm for the m-machine n-job flow shop sequencing problem", Omega, vol. 11, no. 1, pp. 91-95, 1983.

[7] Chakraborty U.K., Laha D., "An improved heuristic for permutation flow shop scheduling", International Journal of Information and Communication Technology, vol. 1, no. 1, pp. 89-97, 2007.

[8] Rios-Mercado R.Z, Bard J., "Computational experience with a branch-and-cut algorithm for flow shop scheduling with setups", Computers and Operations Research, vol. 25, no. 5, pp. 351-366, 1988.

[9] Rios-Mercado R.Z, Bard J., "A branch-and-bound algorithm for permutation flow shops with sequence-dependent setup times", IEE Transactions, vol. 31, pp. 721-731, 1999.

[10] Rios-Mercado R.Z, Bard J. ,"Heuristics for the flowline problem with setup costs", European Journal of Operational Research, vol. 110, pp. 76-98, 1998.

[11] Rajesh Vanchipura, Sridharan R., "Development and analysis of constructive heuristics algorithms for flow shop scheduling problems with sequence dependent setup times", International Journal of Advanced Manufacturing Technology, vol. 67, pp. 1337-1353, 2012.

[12] Osman I.H., Potts C.N., "Simulated annealing for permutation flowshop scheduling", OMEGA, vol. 17, no. 6, pp. 551-557, 1989.

[13] Rajesh Gangadharan, Chandrasekharan Rajendran, "A Simulated Annealing Heuristic for Scheduling in a Flowshop with Bicriteria", Computers ind. Engng, vol. 27, no. 1-4, pp. 473-476, 1994.

[14] Chakravarthy K., Rajendran C., "A heuristic for scheduling in a flowshop with the bi-criteria of makespan and maximum tardiness minimization", Production Planning and Control, vol. 10, pp. 707-714, 1999.

[15] Chen C.L, Vempati V.S, Aljaber N., "An application of genetic algorithms for flow shop problems", European Journal of Operational Research, vol. 80, pp. 389-396, 1995.

[16] Reeves C.R., "A genetic algorithm for flowshop sequencing", Computers and Operations Research, vol. 22, no. 1, pp. 5-13, 1995.

[17] Ruiz R, Maroto C, Alcaraz J., "Two new robust genetic algorithms for the flowshop scheduling problem", Technical Report, Universidad Politecnica de Valencia, Valencia, Espana, Grupo de Investigation Operativa GIO, OMEGA, The International Journal of Management Science, 2003.

[18] Nowicki E, Smutnicki C., "A fast tabu search algorithm for the permutation flow-shop problem", European Journal of Operational Research, vol. 91, pp. 160-175, 1996.

[19] Suliman S.M.A., "A two-phase heuristic approach to the permutation flow-shop scheduling problem", International Journal of Production Economics, vol. 64, pp. 143-152, 2000.

[20] Rajendran C., Ziegler H., "Ant-colony algorithms for permutation flowshop scheduling to minimize makespan/total flowtime of jobs", European Journal of Operational Research, vol. 155, pp. 426-438, 2004.

[21] M. Fatih Tasgetiren , Yun-Chia Liang, Mehmet Sevkli, Gunes Gencyilmaz, "A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem", European Journal of Operational Research Volume, vol. 177, no. 3, pp. 1930-1947, 2007.

[22] M. Fatih Tasgetiren , Quan-Ke Pan , P.N. Suganthan, Angela H-L Chen., "A discrete artificial bee colony algorithm for the total flowtime minimization in permutation flow shops", Information Sciences, vol. 181, no.16, pp. 3459-3475, 2011.

[23] Pinar Civicioglu, "Backtracking Search Optimization Algorithm for numerical optimization problems", Applied Mathematics and Computation, vol. 219, pp. 8121-8144, 2013.

[24] Qun Lin, Liang Gao, Xinyu Li, Chunjiang Zhang, "A hybrid backtracking search algorithm for permutation flow-shop scheduling problem", Computers & Industrial Engineering, vol. 85, pp. 437-446, 2015.

[25] Taillard E., "Benchmarks of basic scheduling problems", European Journal of Operation Research, vol. 64, pp. 278–285, 1993.

[26] Parthasarathy S., Rajendran C.A., "A simulated annealing heuristic for scheduling to minimise weighted tardiness in a flow shop with sequence dependent setup times of jobs - a case study", Production Planning and Control, vol. 8, no. 5, pp. 475-483, 1997.

[27] Ruiz R., Maroto C., Alcatraz J., "Solving the flow shop scheduling problem with sequence dependent setup times using advanced meta-heuristic", European Journal of Operational Research, vol. 165, no. 1, pp. 34-54, 2005.

[28] Gajpal Y., Rajendran C., Ziegler H., "An ant colony algorithm for scheduling in flowshops with sequence-dependent setup times of jobs", International Journal of Advanced Manufacturing Technology, vol. 30, no. 5-6, pp. 416-424, 2006.

[29] F. Fred Choobineh, Esmail Mohebbi, Hansen Khoo, "A multi-objective tabu search for a single-machine scheduling problem with sequence-dependent setup times", European Journal of Operational Research, vol. 175, no. 1, pp. 318-337, 2006.

[30] Rajesh Vanchipura, R. Sridharan, Subash Babu, "Improvement of constructive heuristics using variable neighbourhood descent for scheduling a flow shop with sequence dependent setup time", Journal of Manufacturing Systems, vol. 33, pp. 65-75, 2014.