

COMPARISON OF METAHEURISTIC METHODS TO MINIMIZE MAKESPAN OF CELL SCHEDULING PROBLEMS

Saravanan M.¹, Nooru Haq A.²

¹Department of Mechanical Engineering, R.V.S College of Engineering and Technology, Dindigul, India

²Department of Production Engineering NIT, Trichirappalli, India

Email: ¹sarandgl2k@yahoo.co.in

Abstract

Scheduling problem in a cellular manufacturing environment is treated as group scheduling problem, which assumes that all parts in a part family are processed in the same cell and no inter-cellular transfer is needed. In a typical CMS environment, however, there could be some exceptional parts, which need to visit machines in the other cells. This fact limits the applicability of group scheduling approaches. These stages are termed as intra-cell scheduling and inter-cell scheduling. Through inter-cell scheduling, the sequence of parts within manufacturing cells is determined. In inter-cell scheduling however, the sequence of cells is obtained. This paper addresses the scheduling of manufacturing cells in which parts may need to visit different cells. A metaheuristic named Scatter Search and Simulated Annealing methods are proposed to solve the intra-cell and SVS-algorithm is to inter cell-scheduling problem. The performances of proposed algorithms are evaluated on benchmark problems selected from literature. The results reveal that the scatter search algorithm performs better than heuristic NEH method in the selected benchmark problems with respect to average makespan.

Keywords: Cell Scheduling, Metaheuristic, Scatter Search, and Makespan.

I. INTRODUCTION

Cellular manufacturing (CM) is a production system in which the parts requiring similar production process are grouped in distinct manufacturing cells. These similarities reduce setup times as similar parts can be processed with similar jigs and fixtures. The major advantages of CM have been reported in literature as reduction in setup time, reduction in throughput time, reduction time in work-in-process inventories, reduction in material handling costs, better quality production control, increment in flexibility, etc (Wemmerlov and Hyer, 1989; Shankar and Vrat, 1999; Olorunniwo and Udo, 2002). Cellular manufacturing system (CMS) also provides a production infrastructure that facilitates successful implementation of modern manufacturing technologies success as Just-in-Time manufacturing, flexible manufacturing systems, computer integrated manufacturing, etc. There are many issues in the design and planning of CMS such as cell formation problem (Joines et al., 1996; Cabrera-Rios et al., 2002; Uddin and Shanker, 2002), layout of CMS (Bazargan-Lari, 1999), production planning in CMS (Riezebos et al., 1998), scheduling in CMS (Wemmerlov and Vakharia, 1991), etc. Of these issues, the cell formation problem is an area that has been widely attempted in literature (Soleymanpour et al., 2002). One of the most important issues to attain the benefits of CMS is effective implementation of its scheduling systems (Hitomi and Ham, 1976). Nevertheless, this area has not been widely attempted in literature as compared to the cell formation problem (Mahmoodi and Dooley, 1992; Logendran et al.,

1995). Due to the similarities in the design, shape, function, etc. parts in a part family generally visit machines in the same sequence with minor differences in setup requirements (Schaller, 2001). Therefore, a part family can be divided into several groups so that each group needs similar setup requirements. In other words, a group is a subset of a part family and all parts in the same group need similar setup requirement. This problem is addressed as flow shop group scheduling or briefly group scheduling in the literature. In group scheduling, it is assumed that each part family can be processed in one cell by duplicating bottleneck machines or subcontracting exceptional parts (Logendran et al., 1995). However, subcontracting exceptional parts may not be practical or duplicating bottleneck machines may not be possible in every CMS environment due to production economics, budget and manufacturing space limitation, etc. Thus, in a typical CMS environment, it is difficult to form independent manufacturing cells and mostly there are some exceptional parts that create inter-cellular moves (Shankar and Vrat, 1998). These constraints limit the applicability of group scheduling methods in real life. This paper considers a scheduling problem in which intercellular moves are allowed and parts may visit machines in the other cells. We propose a heuristic, namely SVS-algorithm to solve the problem. The name SVS is due to the first letter of authors' name as this convention is in tune with some of the related works, namely CDS method (Campbell et al., 1970), NEH method (Nawaz et al., 1983), etc. This paper is organized as

follows: Section 2 includes a review of literature and discussion of the methods developed for group scheduling problem. In Section 3, the cell scheduling problem considered in this paper is described and formulated. Section 4 describes the proposed algorithms. The computational results of SVS-algorithm are reported in Section 5. Section 6 contains conclusions and discussions.

II. REVIEW OF LITERATURE

A majority of the algorithm developed for group scheduling problem has two stages. The first stage determines the sequence of parts within the groups and the second stage determines the sequence of groups. Hitomi and Ham (1976) define a lower bound for the optimum makespan and propose a branch and bound technique to determine the optimum sequence of parts and groups. Since the group scheduling problem is NP-hard (Logendran and Srisankarajah, (1993), several researchers have attempted to develop heuristics for the group scheduling problem. Yoshida and Hitomi (1979) provided an algorithm for optimum solution of two-machine flow shop scheduling problem with setup times. Sekiguchi (1983) and Baker (1990) extended the work of Yoshida and Hitomi (1979) to two-machine group scheduling problem in which each group requires different setup on machines. Logendran and Nudtasomboon (1991) propose a heuristic referred to a LN method for solving the intra-group scheduling problem, which is very similar to the NEH algorithm proposed by Nawaz et al (1983). The only difference between the NEH algorithm and LN method is that in the NEH algorithm jobs are initially sorted according to the descending order of total processing time, but this is done in descending order of average processing time in LN method.

Wemmerlov and Vakharia (1991) compared the performance of eight part family scheduling procedures and reported that the family-based scheduling approaches perform superior with respect to minimum flow time and lateness. Logendran et al (1995) have studied the performance of different combination of Petrov's (PT) method 1968, Logendran and Nudtasomboon (LN) method (1991) and Campbell, Dudek and Smith (CDS) method Campbell et al (1970) and reported that the LN-PT method, i.e. use of LN method at the first stage and PT method at the second stage performs superior over the PT-LN, PT-CDS and CDS-PT combinations. The PT and CDS algorithms are respectively single-and multiple-pass heuristics, which simplify an N-jobs M-machine flow shop scheduling problem into an N-job two-machine problem and then use Johnson's algorithm Johnson's, (1954) to determine the sequence of jobs. A detailed description of these methods can be found in Logendran et

al., (1995). Yang and Liao (1996) consider a group scheduling problem with two cells and inter-cellular moves and propose a branch and bound technique and a heuristic to solve this problem.

Yang and Chern (2000) consider two-machine flow shop group scheduling problem in which each group requires same setup and removal times on both machines. Schaller (2001) reports a new lower bound tighter than the one proposed by Hitomi and Ham (1976) that is used to evaluate partial sequences in the branch and bound procedure for the flow shop group-scheduling problem. M.Solimanpur, Prem Vrat and Ravi Shankar (2004) proposed a heuristic algorithm (SVS-algorithm) to minimize makespan in cell scheduling problem. The results reveal that the SVS-algorithm performs better than LN-PT method in all the selected problems with respect to average makespan. We propose a two stage method as reported in the literature, M.Solimanpur et al,2004. This work proposes an efficient metaheuristic method called Scatter Search, instead of NEH method for scheduling of parts within the cells (intra-cellular) and we use the same algorithm (SVS algorithm) used by M.Solimanpur et al,2004 for inter-cellular moves to minimize the makespan. The literature shows Scatter Search method perform better for scheduling problems, refer Saravanan M , Noorul Haq et al,2007 .

III. PROBLEM STATEMENT AND FORMULATION

The scheduling problem attempted in this paper concerns two distinct sequencing problems. These are: sequencing of parts within the cells and sequencing of cells. In this paper, these problems are referred to as intra-cell and inter-cell scheduling, respectively. The following assumptions are made for the considered scheduling problem:

- All parts in all part families are available for processing at time zero.
- Once an operation starts on a machine, it cannot be interrupted before getting completed (non-preemption).
- All parts in a part family require the same setup time on each machine.
- Setup times are independent of the sequence of parts or cells.
- There is no backtracking in the sequence of machines required by a part family. Once a part is completed on a machine, either it continues processing in the same cell or leaves that cell and joins another cell. As this part returns back to its primary cell, it does not go to the machines already visited.

To clarify the problem attempted in this paper, we consider a typical CMS with three cells. In this system, 16 parts are to be scheduled on 11 machines. The processing time of each part on each machine is given in Table 1. Table 2 shows the sequence of machines for processing each part family. As seen in Table 4.2, machines H, D, B and G are bottleneck. For example, two cells share machine D, though the setup requirement of each part family on this machine is different.

Table 1. Processing time of each part on each machine

J / M	Cell1					Cell2					Cell3					
	5	6	9	10	11	12	1	2	4	13	8	3	7	14	15	16
A	8	5	7	11	12											
C	7	15		6	7											
E	8	11	16		20											
F	9		20	14												
H		14	5	12	6	5				13						
D	11					7	8	9	10	11	12					
G						11	6	8	10	6				8		
I						13	5	7		5						
B	4											2	5	11	8	14
J												6	3	14	9	5
K												7	12	6	10	

Table 2. Sequence of machine

Cell1	A-C-E-F-H-D-B
Cell2	H-D-G-I
Cell3	G-B-J-K

The sequence of machines reported in the last column of Table.2 is the sequence by which the associated part family is processed. In this sequence, the machines not required by a particular part are skipped. For example, part 5 which belongs to cell 1 needs processing on machines A, C, E, F and H. Therefore, part 5 initially goes to machine A, C, E, F, H, D and then skips to machine B. Two conditions are to be satisfied in order to process a part on a machine. These are: (1) the machine has to finish processing of earlier job, and (2) the part has to get released from preceding machine.

Table 3. Calculation of Makespan for Intra cell Scheduling

part s	5		6		9		10		11	
	IN	OUT	IN	OUT	IN	OUT	IN	OUT	IN	OUT
A	0	8	8	13	13	20	20	31	31	43
C	8	15	15	30			30	37	43	50
E	15	23	30	41	41	57			57	77
F	23	32			57	77	77	91		
H			41	65	77	82	91	103	103	109
D	32	43								
B	43	47								

Table 3 shows a grid graph representation of the scheduling problem in cell 1. In this figure, the columns stand for parts and the rows stand for machines. As seen in this figure, the largest r is 109 and therefore, the makespan in cell 1 is 109 for Permutation $\Pi = (5,6,9,10,11)$.

IV. THE PROPOSED METHODOLOGY

The scheduling problem addressed in this paper consists of two problems, namely intra-cell and inter-cell scheduling problems. Sequencing parts within cells is called intra-cell scheduling and sequencing part-families/cells is termed inter-cell scheduling. The scheduling approach adopted in this paper is of exhaustive type in which all machines process parts and part families in the same order. In other words, no part/part family can seize a machine before its preceding part/part family. Yang and Liao (1996) pointed out that although non-exhaustive scheduling can make a better tradeoff between the processing and setup times, several advantages such as learning effects and reduction of the defect rate may be obtained by adopting exhaustive scheduling. In addition, Mahmoodi and Dooley (1991) conducted a simulation study and reported that the performance of exhaustive scheduling is generally superior to that of non-exhaustive scheduling.

The proposed approach performs as follows: at the first stage (intra-cell scheduling), the sequence of parts is determined for each part family. The inter-cellular moves are ignored in this stage and it is assumed that all the machines required to process a part family is available in the relevant cell. Therefore, at the first stage we have C independent sequencing problems, where C is the number of cells. At the second stage (inter-cell scheduling), bottleneck machines are taken into account and part families are sequenced using a proposed heuristic procedure.

A. Intra-cell scheduling

In this Stage, we propose a meta-heurist method to sequence the parts within the cell. Metaheuristic is an algorithmic approach to approximate the optimal solution for problems in combinatorial optimization. Heuristic can be used to find good solutions to hard problems, metaheuristic drives to produce better solutions. A metaheuristic is described in as “an iterative master process that guides and modifies the operations of subordinate heuristics to efficiently produce high-quality solutions. It may manipulate a complete (or incomplete) single solution or a collection of solutions at each iteration. The subordinate heuristics may be high (or low) level procedures, or a simple local search, or just a construction method”.

A.1. Scatter Search

Scatter search (SS) is a metaheuristic method in which solutions are combined to yield better solutions. Scatter search (SS) is an evolutionary or population-based method in which solutions are combined to yield better solutions using convex or non-convex linear combinations. Strategies for diversification and intensification are typically added to enhance the search.

An algorithm that is able to obtain good solutions in short times is proposed for this problem. This method is based on the recent metaheuristic strategy named scatter search (SS). This method also incorporates others procedures based on others methods, such as local search, tabu search, GRASP and path relinking, Joaquin, A. Pacheco. This SS approach is analyzed and compared with other recent techniques. In all cases, our proposed technique gives adequate solutions, compared with others recent techniques, in reasonable time SS has been successfully implemented in a variety of settings including combinatorial optimization and nonlinear optimization in continuous variables. Scatter search embodies principles and strategies that are still not emulated by other evolutionary methods, and that prove advantageous for solving a variety of complex optimization problems. More about the origin and multiple applications of SS can be found in Glover (1998), Glover et al. (1999) and Laguna (2000).

The basic steps involved in the static scatter search are:

Step 1: Use the Diversification Generator to generate diverse trial solutions from the seed solution(s)

Step 2: Use the Improvement Method to create one or more enhanced trial solutions

Step 3: With these initial solutions update the Reference Set (RefSet)

Step 4: Repeat

4.1 Generate subsets of the RefSet.

4.2 Combine these subsets and obtain new solutions.

4.3 Use the Improvement Method to create a more enhanced trial solution.

4.4 Continue to maintain and update the Reference Set until Refset is stable (No new solutions are included)

Step 5: if iterations (Steps 1 – 4) elapse without improvement stop else return to step 1.

1. A *Diversification Generation Method* to generate a collection of diverse trial solutions, using an arbitrary trial solution (or seed solution) as an input.

2. An *Improvement Method* to transform a trial solution into one or more enhanced trial solutions. (Neither the input nor the output solutions are required to be feasible, though the output solutions will more usually be expected to be so. If no improvement of the input trial solution results, the “enhanced” solution is considered to be the same as the input solution.)
3. A *Reference Set Update Method* to build and maintain a *reference set* consisting of the b “best” solutions found (where the value of b is typically small, e.g., no more than 20), organized to provide efficient accessing by other parts of the method. Solutions gain membership to the reference set according to their quality or their diversity.
4. A *Subset Generation Method* to operate on the reference set, to produce a subset of its solutions as a basis for creating combined solutions.
5. A *Solution Combination Method* to transform a given subset of solutions produced by the Subset Generation Method into one or more combined solution vectors. Specific processes for carrying out these steps are described in Glover.

A.2. SA Algorithm

The efficiency of this method depends on the following parameters, which have to be selected precisely:

1. Initial solution,
2. Cooling scheme (how temperature changes during the search process),
3. Neighbourhood value and
4. Stopping condition.

The parameters and steps of the Simulated Annealing Algorithm are as follows.

Step 1: Initialization

Set initial temperature, $T = 1000$ c

Reduction of temperature factor $(\alpha) = 0.95$

Number of Iterations $(n) = 25$

Step 2: Generation of initial temporary seed ('S') and setting its objective value as 'Ms'

Step 3: Generate number of nearer neighborhood sequences to S using pair wise exchange

Step 4: Find objective function of all sequences generated in step3. Sort the minimum value and store it in (Ms').

Step 5: Calculate Acceptance parameter Delta ()

If $(= Ms - Ms') \leq 0$ then proceed to step (5). Else go to step (6).

Step 6: $S = S'$ and $Ms = Ms'$ go to step (7)

Step 7: $\frac{1}{2}0$, means

Probability, $P = \exp^{-T} = e^{-2/1000} = \text{say, } 0.99$. Now select a random number between 0 and 1. Preferably with two decimal points say, $U = 0.65$

If $P < U$, select $S' = S$ otherwise select $S = S'$

Step 8: Again do the neighbourhood generation for new value

$n = n + 1$; Do the same procedure for 25 times.

Step 9: If $n \geq 25$, reduce the temperature value 'T'

New, $T = \alpha T = 0.95 \times 1000 = 950$

Repeat the procedure up to the temperature becomes Zero. Go to step (3). Then the algorithm is frozen and the best sequence is obtained.

B. Inter Cell Scheduling

Once the sequences of parts within cells are determined, the next problem is to sequence cells so as to minimize maximum completion time of all part families on all machines. Let σ denote the sequence of cells in inter-cell scheduling problem. In the proposed method, a pair wise comparison of cells is performed to sequence different cells. There are two possible partial sequences for cells c and c' , viz. $\sigma_1 = (c, c')$ and $\sigma_2 = (c', c)$ In the proposed method, the makespan is calculated for both partial sequences σ_1 and σ_2 . Of sequences σ_1 and σ_2 , the sequence that results into smaller makespan is selected and the score of the cell appearing first in the parenthesis is increased by 1. This comparison is done for every pair of cells and thus the final sequence of cells is obtained in the sequence of decreasing order of scores. Solimanpur, M., Vrat, P., Shankar proposes SVS method. (Refer for complete details)

The stepwise procedure of the second stage of the proposed SVS-algorithm is as follows:

Step 1: Set $c = 1$, $i = 1$ and $w(k) = 0$; $\forall k = 1, 2, \dots, C$.

Step 2: Set $c^0 = c + i$.

Step 3: Calculate the makespan for sequences $\sigma_1 = (c, c')$ and $\sigma_2 = (c^0, c)$.

Let $C_{\max}(\sigma_1)$ and $C_{\max}(\sigma_2)$ are the makespan associated with sequences σ_1 and σ_2 respectively.

Step 4: If $C_{\max}(\sigma_1) < C_{\max}(\sigma_2)$, set $w(c) = w(c) + 1$.

Step 5: If $C_{\max}(\sigma_1) > \frac{1}{2} C_{\max}(\sigma_2)$, set $w(c^0) = w(c^0) + 1$.

Step 6: If $c = C - 1$, go to step 8.

Step 7: If $c < C$, set $i = i + 1$ and go to step 2.

Otherwise, set $c = c + 1$, $i = 1$ and go to step 2.

Step 8: Arrange cells by descending order of Values for w and then STOP.

V. COMPUTATIONAL RESULTS

As mentioned in literature, solimanpur (2004) have experimentally verified that the application of NEH-SVS method performs superior over other combinations including PT-LN, PT-CDS, and LN-PT and CDS-PT. Thus, the results obtained through Scatter Search algorithm are compared to those obtained by NEH-SVS algorithm. The SA and SS-method were coded in C language and run on a 2.56GHz Pentium IV PC. We have selected benchmark problems from literature to evaluate the performance of the proposed metaheuristic methods. The size of each problem in terms of total number of parts, machines, and cells has been reported. The average makespan obtained by NEH-SVS method and SS-SVS and SA-SVS algorithm is reported in the table 4 for all the problems. As seen in this table, the SS-method obtains lesser makespan in problems as compared to NEH-SVS and SA-SVS methods. The last column in the last table shows the percentage of reduction in average makespan (PRAM %) as compared to NEH-SVS method. This measure is calculated as follows.

$\text{PRAM}\% = \left(\frac{\text{The average makespan obtained by NEH-SVS method} - \text{The average makespan obtained by SS-SVS algorithm}}{\text{the average makespan obtained by NEH-SVS method}} \right) \times 100$

Table 4. Performance Evaluation of Scatter Search

SL. NO	Problem	Size			NEH-SVS C _{max}	SS-SVS C _{max}	SA-SVS C _{max}	PRAM %
		P ¹	M ²	C ³				
1	PSPLIB	16	11	3	90	81	83	10
2	Solimanpur et al., (2003)	10	8	3	76	76	76	0
3	PSPLIB	20	16	3	148	133	131	10.13
4	Waghodekar et al (1984a)	7	5	2	21	21	21	0
5	Waghodekar et al (1984b)	7	5	2	62	62	60	0

1. P is Parts, 2. M is Machines, 3. C is Cell, 4. C_{max} is Makespan, and 5. PRAM (%) is the percentage of reduction in average makespan.

VI. DISCUSSIONS AND CONCLUSION

This paper concerns the scheduling problem in CM environment. In many cases, the conventional flow shop

group scheduling approach cannot be applied to CM-based systems. The major shortcomings of the group scheduling problem have been discussed in this paper. This paper considers the cell scheduling problem in the presence of intra-cellular moves and inter-cellular moves in CMS. In this sense, the considered problem is more general than the conventional flow shop group-scheduling problem. We propose metaheuristic methods SS and GA to solve the intra-cell problem and SVS-algorithm to solve the inter-cell problem with an aim to minimize the makespan. The obtained results indicate that SS has ability to reduce the makespan as compared to heuristic and Metaheuristic algorithms. The conclusion is that this metaheuristic scatter search can be considered as a stable one.

REFERENCES

- [1] A.Vakharia, and Y. Chang,1977, "Cell Formation in Group Technology: A Combinatorial Search Approach". International Journal of Production Research, 35(7),pp.2025-2043
- [2] A.Noorul haq and M.Saravanan, 2007, "A scatter search algorithm for general flowshop scheduling problem". International journal of Advanced manufacturing Technology, 31,pp.731-736.
- [3] C.Fleurent,F.Glover, P. Michelon and Z. Valli, 1996. "A Scatter Search Approach for Unconstrained Continuous Optimization," Proceedings of the 1996 IEEE International Conference on Evolutionary Computation, 643-648.
- [4] CC. Gallagher, W.A.Knight, 1986. Group Technology Production methods in Manufacturing. Ellis Horwood Limited, England.
- [5] D-L Yang, M-S. Chern, 2000. Two-Machine flow shop group scheduling problem, Computers & Operations Research 27, pp.975-985.
- [6] F. Glover, 1994a. "Genetic Algorithms and Scatter Search: Unsuspected Potentials," Statistics and Computing, 4, pp.131-140
- [7] F. Glover, M.Laguna, R. Marti, 2000. "Fundamentals of scatter search and path relinking. Control and Cybernetics", 39(3), pp.653-84,
- [8] F.Glover, 1998, A template for scatter search and path relinking. In: Hao J-K, Lutton E, Ronald E, Schoenauer M, Snyers D, editors. Artificial evolution. Lecture notes in computer science, vol. 1363, pp. 13-54, Berlin: Springer
- [9] F.Olorunniwo, G.Udo, 2002. The Impact of management and employees on cellular manufacturing implementations. International Journal of Production Economics 76,pp. 27-38. 21
- [10] H.G.Campbell, R.A. Dudek,, M.L Smith, 1970. A heuristic algorithm for the n job and m machine sequencing problem. Management Science 16, B630 B637
- [11] J. Black,1998, Cellular Manufacturing Systems Reduce Setup Time, Make Small Lot Production Economical_.Industrial Engineering, pp.36-48
- [12] J.Schallar, , 2001. A new Lower bound for the flow shop group scheduling problem. Computers and Industrial Engineering 41, pp.151-161.
- [13] K Hitomi, I. Ham, 1976. Operations scheduling for group technology applications. Annals of the CIRP 25(1),pp. 419-422
- [14] K R, Baker,1990. Scheduling groups of jobs in the two machine flow shop, Mathematical and computer modeling 13, pp.29-36.
- [15] M Groover, 1987, Automation, Production System and Computer Integrated Manufacturing, Prentice Hall, Engle wood NJ.
- [16] M. Nawaz,,J E E Ensore, I.Ham, 1983. A heuristic algorithm for the m-machine, n-job flow shop sequencing problem. OMEGA: International Journal of Management Science. 11,pp. 91-95.
- [17] M.Saravanan ,A. Noorul Haq, 2007,"A Performance Evaluation of Scatter Search Method for Permutation Flow-Shop Sequencing Problems" International journal of Advanced manufacturing Technology.Available in online.
- [18] M.Soleymanpour, P.Vrat, C. R.Shankar, 2002. A transiently chaotic neural network approach to the design of cellular manufacturing. International Journal of Production Research 40, pp. 2225-2244.
- [19] M.Solimanpur, Prem Vrat, C.Ravi Shankar, 2004, A heuristic to minimize makespan of cell scheduling problem, International journal of Production Economics, 88,231-241.
- [20] Manuel Laguna, Rafael Vicente Campos, 1998. "Intensification and Diversification with Elite Tabu Search Solutions for the Linear Ordering Problem", to appear in computers and operations research,
- [21] N.A.Petrov, 1968, "Flow line Group Production Planning", Business Publications Ltd, London. Production, Planning and Control, Panneer Selvam, Prentice-Hall India