Journal of Industrial Engineering International *Islamic Azad University, South Tehran Branch*
July 2008, Vol. 4, No. 7, 48-58

A heuristic approach for multi-stage sequence-dependent group scheduling problems

N. Salmasi*
Department of Industrial Engineering, Sharif University of Technology, Tehran, Iran

R. Logendran
School of Mechanical, Industrial and Manufacturing Engineering, Oregon State University, Corvallis, OR, 97331, USA

Abstract

We present several heuristic algorithms based on tabu search for solving the multi-stage *sequence-dependent group scheduling* (SDGS) problem by considering minimization of makespan as the criterion. As the problem is recognized to be strongly NP-hard, several meta (tabu) search-based solution algorithms are developed to efficiently solve industry-size problem instances. Also, two different initial solution generators are developed to aid in the application of the tabu search-based algorithms. A lower bounding technique based on relaxing the mathematical model for the original SDGS problem is applied to estimate the quality of the heuristic algorithms. To find the best heuristic algorithm, random test problems, ranging in size from small, medium, to large are created and solved by the heuristic algorithms. A detailed statistical experiment, based on nested split-plot design, is performed to find the best heuristic algorithm and the best initial solution generator. The results of the experiment show that the tabu search-based algorithms can provide high quality solutions for the problems with an average percentage error of only 1.00%.

Keywords: Scheduling; Group scheduling; Integer Programming; Tabu search; Lower bound

1. Introduction

Manufacturing companies are forced to improve their efficiency and flexibility in order to survive. Cellular Manufacturing (CM) is a concept used since the 70's to increase the productivity and flexibility of production in manufacturing companies. In CM, the parts are assigned to different groups based on their similarities in shape, material, or processing operations. The machines are also assigned to different cells in order to decompose the production line. Each group is then assigned to a particular cell, which includes different machines that have the capability to perform the necessary operations for each part that belong to the group. This decomposition of machines and parts (called jobs in this paper) has several advantages such as significant reduction in set-up time, work-in-progress inventories, and simplified flow of parts and tools.

Sequencing and scheduling are forms of decision making that improve the efficiency of production by finding the best sequence of processing the assigned jobs to a set of machines. In CM, finding the best sequences of jobs as well as groups is called Group Scheduling (GS). In GS, the best sequence of processing the assigned groups to the cell as well as the jobs in each group to optimize some measure of effectiveness is investigated. One of the relevant objectives in the investigation of GS problems, *minimization of makespan*, is considered in this paper.

GS problems based on their required group set-up time are classified into two major groups: sequence dependent, and sequence independent scheduling. If the set-up time of a group for each machine depends on the immediately preceding group that is processed on that machine, the problem is classified as "sequence dependent group scheduling (SDGS);" otherwise, it is called "sequence independent group scheduling".

There are many real world applications of sequence dependent scheduling problems. Schaller et al. [18] discussed an industry case of sequence dependent

* *Corresponding author. E-mail:nsalmasi@sharif.edu*

group scheduling problem in printed circuit boards (PCBs) in which the major set-up is required to switch from a group of PCBs to another. Painting automobiles with different colors in small batch sizes is another example of sequence dependent set-up scheduling problems.

A comprehensive literature review of group scheduling problems was performed by Allahverdi et al. [1] and Cheng et al. [2]. Vakharia et al. [20] and Schaller et al. [17, 18] present branch-and-bound based approaches as well as several heuristic algorithms to solve the SDGS problem with multiple machines by considering minimization of makespan criterion. The highlight of their research is published in Schaller et al. [18]. Franca et al. [3] developed an algorithm based on genetic algorithm and a memetic algorithm with local search to solve the SDGS problem by considering minimization of makespan.

The literature review reveals that there still exist several potential areas worthy of further research on SDGS problems [2]. The industry needs a solution approach with good quality (optimal or near optimal) in a short time. Considering the widespread practical applications of SDGS problems in industry such as auto industry and hardware manufacturing, and the importance of minimizing the makespan criterion, further research on this topic is still required. Indeed, that is the motivation for the research reported in the next several sections.

2. Problem description

In this research, it is assumed that g groups are assigned to a cell that has m machines. Each group includes b_i jobs ($i = 1, 2, \ldots, g$). The set-up time of a group for each machine depends on the immediately preceding group that is processed on that machine (sequence dependent set-up time).

The goal is to find the best sequence of processing the jobs in each group as well as groups themselves by considering minimization of makespan. The assumptions made in this research are:

- The problem belongs to permutation scheduling problems. This is the only available method to produce in some industries. For instance, if a conveyer is used to transfer jobs among machines, then all jobs should be processed in the same sequence on all machines.
- All jobs and groups have the same importance (weight) for the company.
- All jobs and machines are available at the beginning of the planning horizon.

3. Complexity of the problem

Gupta and Darrow [5] proved that the two machine sequence dependent job scheduling (SDJS) problem is a NP-hard problem. Garey et al. [4] also proved that the flowshop job scheduling problem by considering minimization of makespan criterion for more than two machines ($m \geq 3$) is an NP-hard problem. Based on these insights, it is easy to see that the problem investigated in this paper is easily reducible to the ones already proven NP-hard. Thus, the fact that the proposed problem is NP-hard, follows immediately.

4. Heuristic algorithm

Since the research problem is shown to be NP-hard, a heuristic algorithm is needed to solve industry-size problems in a reasonable time. Previous research by Skorin-Kapov and Vakharia [19], Nowicki and Smutnicki [13], Logendran and Sonthinen [11], and Schaller [16], has shown tabu search (TS) to be a promising technique for solving similar scheduling problems.

TS is a metaheuristic algorithm which is developed independently by Glover [6] and Hansen [7] for solving combinatorial optimization problems. It attempts to avoid getting trapped in a local optimal solution. While other hill-climbing heuristics terminate once a local optimum has been found, TS continues searching for a better solution.

The TS method, like other heuristic algorithms, needs an initial solution. At each iteration, the search moves from the current solution to the best solution in the neighborhood, which may have inferior objective function value to the previously found one. The process is continued until one of the stopping criteria is satisfied. In this research the stopping criteria are either the specified number of local optima or the maximum number of iterations without improvement. The moves to the solutions that contain the attributes of recently visited solutions are temporarily forbidden for a number of iterations in order to prevent cycling. If the objective function value of the new solution is better than a value called "the aspiration level", this tabu restriction is overridden. TS also contains short-term memory, which keeps track of the moves that are currently declared tabu.

Besides the short-term memory, the long-term memory is applied to enhance the quality of the solution. The long-term memory keeps the information on the frequency that each attribute appears in the solutions. In this application, the attribute indicates the position of a group or a job within a group. The search can be intensified by shifting the explorations in the neighborhoods of the good solutions with frequently added attributes and diversified by introducing the search to explore new regions that do not contain the frequently added attributes of the inferior solutions.

The process of finding a solution for the research problem involves two levels. The first level investigates to find the best sequence of groups. During the first level, a sequence of groups is chosen. The second level investigates to find the sequence of jobs in each group based on the chosen group sequence by the first level. Thus, a two-level TS is developed to solve the research problem. In the first (outside) level, the best sequence of groups is investigated. It is done by moving from one group sequence to another. When a sequence of groups by the outside level is chosen, the second (inside) level finds the best sequence of jobs that belong to each group by considering minimization of makespan. This is done for the inside search by moving from a sequence of jobs in a group sequence to another sequence of jobs in the same group sequence. The relationship between the outside and inside search is that when the outside search is performed to get a new group sequence, the search process is shifted to inside search. The inside search is performed to find the best sequence of jobs in groups by considering the proposed group sequence by outside search. When the inside search stopping criteria are satisfied, the best found job sequence is considered. Then, the search returns back to the outside search. The outside search stops when the outside search stopping criteria are satisfied. The best found solution during the search is reported as the final solution. The final solution is comprised of the sequence of groups and the sequence of jobs in each group that provides the best makespan for the objective function.

4.1. Construction of initial solution

The quality of the final solutions as well as the efficiency of the search may significantly be improved if a good quality initial solution generator is applied. Schaller et al. [18] developed a heuristic algorithm to solve SDGS problems and suggested that to apply the result of their algorithm as an initial solution for a heuristic algorithm such as TS. In order to evaluate this suggestion, two different initial solution generating mechanisms are developed in this research. One based on a random initial solution and another based on the proposed algorithm by Schaller et al. [18].

4.2. Generation of neighborhood solutions

When a feasible solution is considered as an initial solution, the neighborhoods of the seed are generated to explore the search. During the inside search, a neighborhood of a seed is generated by applying swap moves, i.e., changing the order of two sequenced jobs that belong to a group. The outside neighborhoods can be derived similar to the inside neighborhoods by applying swap moves.

4.3. Steps of TS

As is typically done in TS, the initial solution is considered as the first entry into the outside candidate list (OCL). Then, the neighborhood solutions are explored by perturbing it. The value of each neighborhood solutions is determined by its objective function value. These neighborhoods have to be compared with the tabu-list filter whose goal is to prevent the cycle trap of local optima. This filter is implemented through comparison of neighborhood solutions against a set of restricted moves listed in tabu-list (TL). This list is constructed based on the recent change in previous best solutions. The tabu-list records these changes or moves in the order they are applied. The size of the tabu-list is determined through extensive experimentation.

When all neighborhood solutions of a seed are generated, the best local move among them is compared against the TL. If the move is restricted, it is normally ignored and the second best move is considered. If a restricted move has a better value than the best global value found so far, namely the aspiration level, the tabu restriction is ignored. The best move, after filtering against TL and aspiration criterion, is compared with the current members of the candidate list. If the chosen neighborhood does not belong to the current candidate list, it is selected for the next perturbation and generation of new neighborhood. Otherwise, the next best neighborhood is chosen. This move is recorded into the TL. This process is repeated until the search is terminated by one of the stopping criteria. When the outside search is completed, the solution with the best objective function value is reported as the result of the search. The steps of performing TS for outside search are depicted in Appendix 1. The process for inside search is the same as well.

4.4. Two-machine SDGS problem with minimization of makespan criterion

For the two machine SDGS problems with minimization of makespan criterion, Logendran et al. [10] showed that the optimal sequence of jobs in each group conforms to Johnson's algorithm [9]. Thus, the TS algorithm for these problems can be relaxed to a one level search in order to find the best sequence of groups. During the search, the sequence of processing jobs belonging to each group is identified based on Johnson's [9] algorithm.

4.5. Parameters for the research problem

The sizes of problems investigated in this research include 2 to 16 groups in a cell and 2 to 10 jobs in a group. The maximum number of total jobs for the problems considered is at most 120 jobs in all groups. Extensive experimentation was performed to develop the empirical formulae for evaluating the parameter values used for these research problems, and the formulae so developed are presented in Appendix 2. In some cases a formula for a range can be generated and in some of them a value for a parameter in a range is offered. If a formula does not provide an integer value for a parameter, the result is rounded down. The two-machine problems, as noted in Section 4.4, require only a one level search.

In this research, three different versions of TS are applied to solve the problem. The first one (TS1) is TS with short term memory. The second one (TS2) is TS with long term memory and intensification (LTM-max) and finally, the third one (TS3) is TS with long term memory and diversification (LTM-min).

5. Lower bound

A lower bounding technique was previously developed to evaluate the quality of the heuristic algorithms for the two-machine SDGS problem [10]. This lower bounding technique was modified by Salmasi [14] by adding a couple of new constraints in order to develop a more enhanced lower bound. The lower bound is based on relaxing the problem from SDGS to Sequence Dependent Job Scheduling (SDJS) problem. Every group is considered as an independent job. The run time of these independent jobs (groups) on each machine is considered equal to the summation of the run time of its jobs on each machine. The optimal solution of this problem is a lower bound for the original problem because the possible idle times between processing jobs that belong to a group on all machines are ignored. In this paper, this lower bound is applied in order to evaluate the quality of the developed heuristic algorithms.

6. Experimental design

An experiment is designed to evaluate the performance of three developed heuristic algorithms based on TS. The factors considered for this design are as follows:

Number of groups: Problems up to 16 groups are investigated in this research. The levels of this factor are defined in three different categories: small, medium, and large. Small size problems are problems including 2 to 5 groups. Problems with 6 through 10 groups are considered as medium size problems, and problems with 11 through 16 groups are classified as large size problems.

Number of jobs in a group: The maximum number of jobs of a group in a problem is considered as a factor. For instance, if in a group scheduling problem with three groups, groups have 3, 6, and 9 jobs respectively, then the problem is classified as a 9-jobs problem. In this research the maximum number of jobs that belong to a group is limited to 10. This factor has also three levels. Level 1 includes problems with at most 2 to 4 jobs in a group. Problems with at most 5 to 7 jobs in a group are classified as level 2, and finally if one of the groups of a problem includes 8 to 10 jobs, then the problem belongs to the third level based on its number of jobs factor.

The ratio of set-up times: The experiments performed indicate that the quality of solutions strongly depends on the ratio of set-up times of groups on consecutive machines. Thus, this factor is considered as a factor. Three levels are defined for this factor. In a sequential machine pair, if the set-up time of the first machine is significantly less than the set-up time of the second machine, the problem belongs to the first level. If both machines have almost the same set-up times, the problem belongs to the second level. Finally, if the set-up time of the first machine is significantly greater than the second machine, the problem is classified as the third level of this factor. This factor should be applied to all sequential machine pairs. For instance, in a three machines problem, this ratio for "M_1/M_2" and "M_2/ M_3" should be compared. Thus, this can be considered as two separate factors in this problem.

Initial solution: The initial solution for the heuristic algorithms is considered as a factor. Each of two different techniques of generating initial solution is considered as a level for this factor.

Algorithm: Each of three heuristic algorithms is considered as a level for this factor.

The group, job, and the set-up ratio factors are the ones which are used to generate a test problem. Then, each test problem is solved by the heuristic algorithms by applying one of the two initial solution generators. Based on this explanation, each experimental unit of the first three factors (which generate a test problem) is split into six different parts to be solved by one of the combinations of the heuristic algorithms and the initial solution generators. Thus, the split plot design is the most appropriate model to compare the results [12]. As the test problems are created based on the groups, jobs, and set-up ratio factors, these factors are put in the whole-plot and the remaining factors, i.e., the algorithm and the initial solution generator factors, are put in the sub-plot. The factors in the whole plot are considered nested to generate a test problem. A problem instance, which is considered as a block for the sub-plot factors, is generated for specific levels of whole-plot factors. The problems (blocks) are treated as a random factor. The model is a mixed model, because it includes fixed factors (groups, jobs, set-up ratios, algorithms, and initial solutions) as well as random factor (problem instances). The model of the experiment for a 3-machine problem can be represented as:

$$Y_{ijklmnr} = \mu + G_i + J_j + R1_k + R2_l + (G*J)_{ij} + (G*R1)_{ik} + (G*R2)_{il} + (J*R1)_{jk} + (J*R2)_{jl} + (R1*R2)_{kl} + T_{t(ijkl)} + \alpha_m + I_n + \text{subplot interactions} + \varepsilon_{ijklmnr},$$

where,

μ The overall mean,

G_i The effect of group factor, $i = 1, 2, 3$,

J_j The effect of job factor, $j = 1, 2, 3$,

$R1_k$ The ratio of set-up time of M_1/M_2 factor,

$R2_l$ The ratio of set-up time of M_2/M_3 factor,

$\varepsilon_{ijklmnr}$ The error term,

T_t The block factor (a random factor),

α_m The effect of algorithm factor $m = 1, 2, 3$,

I_n The effect of initial solution factor $n = 1, 2$,

k 1, 2, 3,

l 1, 2, 3.

The goals of performing the experimental design are:

- Find the heuristic algorithm with the best performance.
- Identify if there is a statistically significant difference between the performances of initial solution generators.

The hypothesis tests to investigate for these goals are:

H_0: $\alpha_1 = \alpha_2 = \alpha_3 = \ldots = \alpha_m$

H_1: if any of the α's is different from the others

H_0: $I_1 = I_2$ H_1: $I_1 \neq I_2$

As noted, a cell with more than six machines is highly unlikely in industry. In the interest of time, an experiment which includes the minimum and the maximum number of machines is considered. Thus, the comparison is performed for 2, 3, and 6 machine problems separately by solving the test problems generated with the heuristic algorithms.

7. Results obtained from TS algorithms

The generated test problems are solved by three different versions of TS by applying two different initial solution generators. The TS algorithm is coded in C programming language. The lower bounding technique is also applied to provide a lower bound for test problems. The ILOG CPLEX (version 9.0) is used to solve the lower bounding model. All of the heuristic algorithms and the lower bounding problems are run on a Power Edge 2650 with 2.4 GHz Xeon, and 4GB RAM. The results for two, three, and six machine problems are shown in Table 1. More detailed results are given in Salmasi [14]. This percentage error is calculated based on: (The heuristic algorithm - The lower bound) / The lower bound.

Table 1. The results for the test problems.

Number of Machines	Average Time to solve LB (seconds)	Percentage error					
		Initial 1			Initial 2		
		TS1	TS2	TS3	TS1	TS2	TS3
Two	10.2	1.3%	0.9%	1.2%	1.3%	1.1%	1.2%
Three	65.4	1.7%	1.4%	1.6%	1.6%	1.4%	1.4%
Six	4720	2.0%	1.8%	2.1%	2.0%	1.8%	1.9%

The experimental design is coded with Statistical Analysis System, SAS, release 9.1, to find the best heuristic algorithm as well as the best initial solution generator. A significance level of 5% is used in all of the tests reported below.

The results of the experiment for two machine problems are shown in Appendix 3. It shows that there is a significant difference among the objective function values of TS heuristic algorithms (the result of F test is equal to 0.0048). To find the difference among the TS heuristic algorithms, the Tukey test is performed. The result of Tukey's test shows that TS2 has a better performance compared to the others. The results of the experimental design also show that there is no difference between the initial solution generators for two machine problems (the result of F test is equal to 0.4975).

The result of the experiment for three machine problem is shown in Appendix 4. It shows that there is a significant difference among the objective function values of heuristic algorithms (the result of F test is less than 0.0001). To find the best heuristic algorithm, a Tukey test is performed. The result of Tukey's test shows that TS2 has a better performance compared to the other two heuristic algorithms. The results of the experimental design show that there is no difference between applying different initial solution generators for three machine problems (the result of F test is equal to 0.2732). For six machine problems (Appendix 5), there is a significant difference among the objective function values of heuristic algorithms (the result of F test is equal to 0.0004).

To find the best heuristic algorithm, a Tukey test is performed. The result of Tukey's test shows that TS2 has a better performance compared to the other two heuristic algorithms. The results show that there is no difference between the initial solution generator for six machine problems (the result of F test is equal to 0.3344).

8. Comparison of the best TS algorithm with other algorithms

Table 2 presents the percentage error obtained for each size of the test problem instances solved with the Schaller et al. [18] initial solution generator as well as the best TS algorithm (TS2 with the first initial solution generator). A paired t-test is performed between the results of the best TS and the results of Schaller et al. [18] algorithm for two, three and six machine problems, separately. The results show that in all three cases (two, three & six machine problems), there is a significant difference between the performance of the TS algorithm and Schaller et al. [18] algorithm and in all of them TS shows a superior performance compared to Schaller et al. [18] algorithm.

9. Conclusion

In this research, for SDGS problems, three different versions of TS with two different initial solution generators are developed. The first initial solution generator is a random sequence generator and the second one is developed based on the result of Schaller et al. [18] algorithm. Test problems with two, three, and six machines are solved by these algorithms. A lower bounding technique is also applied to evaluate the quality of the heuristic algorithms. The results of the experiment show that TS2 (LTM_Max) has the best performance compared to the other heuristic algorithms.

Table 2. The results of current available algorithms.

Number of machines	Percentage error for The best TS	Percentage error for Schaller et. al. (2000)
Two	0.9%	9.1%
Three	1.4%	8.8%
Six	1.8%	7.1%

In other words, TS2 provides a better sequence for groups as well as jobs in each group. Based on the results, there is no statistically significant difference between the objective function values of the heuristic algorithms by applying different initial solutions. It means that applying Schaller et al. [18] algorithm as the initial solution generator does not help to improve the quality of solution.

The feasible solution space of the problem has too many local optima. Thus, starting with a good quality local optimal solution as the initial solution does not guarantee of obtaining a better final solution by the heuristic algorithm. This may be the reason for not improving the quality of solutions by applying Schaller et al. [18] algorithm as an initial solution generator. The comparison of results of the TS algorithm and the lower bound shows that the average percentage error of TS for the test problems with two, three, and six machines is around 1.0%. The detailed experimentation has also proved the fact that TS is clearly superior to Schaller et al. [18] algorithm.

Acknowledgments

This research is funded in part by the National Science Foundation (USA) Grant No. DMI-0010118. Their support is gratefully acknowledged.

References

[1] Allahverdi, A., Gupta, J. N. D. and Aldowaisan, T., 1999, A review of scheduling research involving setup considerations. *Omega, Int. Journal of Management Science,* 27, 219-239.

[2] Cheng, T. C. E., Gupta, J. N. D. and Wang, G., 2000, A review of flowshop scheduling research with set-up times. *Production and Operations Management,* 9(3), 262-282.

[3] Franca, P. M., Gupta, J. N. D., Mendes, P. M. and Veltink, K. J., 2005, Evolutionary algorithms for scheduling a flowshop manufacturing cell with sequence dependent family setups. *Computers and Industrial Engineering,* 20, 1-16.

[4] Garey, M. D., Johnson, D. S. and Sethi, R., 1976, The Complexity of flowshop and jobshop scheduling. *Mathematics of Operations Research*, 1(2), 117-129.

[5] Gupta, J. N. D. and Darrow, W. P., 1986, The two-machine sequence dependent flowshop scheduling problem. *European Journal of Operational Research*, 439-446.

[6] Glover, F., 1986, Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, 13, 533-549.

[7] Hansen, P., 1986, *The Steepest Ascent Mildest Decent Heuristic for Combinatorial Programming*. Conference on Numerical Methods in Combinatorial Optimization, Capri, Italy.

[8] ILOG, CPLEX, 2003, Release 9.0, Paris, France.

[9] Johnson, S. M., 1954, Optimal two and three stage production scheduling with set-up times included. *Naval Research Logistic Quarterly*, 1(1), 61-68.

[10] Logendran, R., Salmasi, N., and Sriskandarajah, C., 2006, Two-machine group scheduling problems in discrete parts manufacturing with sequence-dependent setups. *Journal of Computers and Operations Research,* 33, 158-180.

[11] Logendran, R. and Sonthinen, A., 1997, A Tabu search-based approach for scheduling job-shop type flexible manufacturing systems. *Journal of the Operational Research Society,* 48, 264-277.

[12] Montgomery, D. C., 2001, *Design and Analysis of Experiments*. New York, John Wiley & Sons.

[13] Nowicki, E. and Smutnicki, C., 1996, A fast Tabu Search algorithm for the permutation flowshop problem. *European Journal of Operations Research*, 91, 160-175.

[14] Salmasi, N., 2005, *Multi-Stage Group Scheduling Problems with Sequence Dependent Set-ups*. Doctoral Dissertation. Oregon State University, Corvallis, Oregon,

[15] SAS Release, 9.1, 2002-2003, SAS Institute Inc., Cary, North Carolina, USA.

[16] Schaller, J. E., 2000, A comparison of heuristic for family and job scheduling in a flow-line manufacturing cell. *International Journal of Production Research,* 28(2), 287-308.

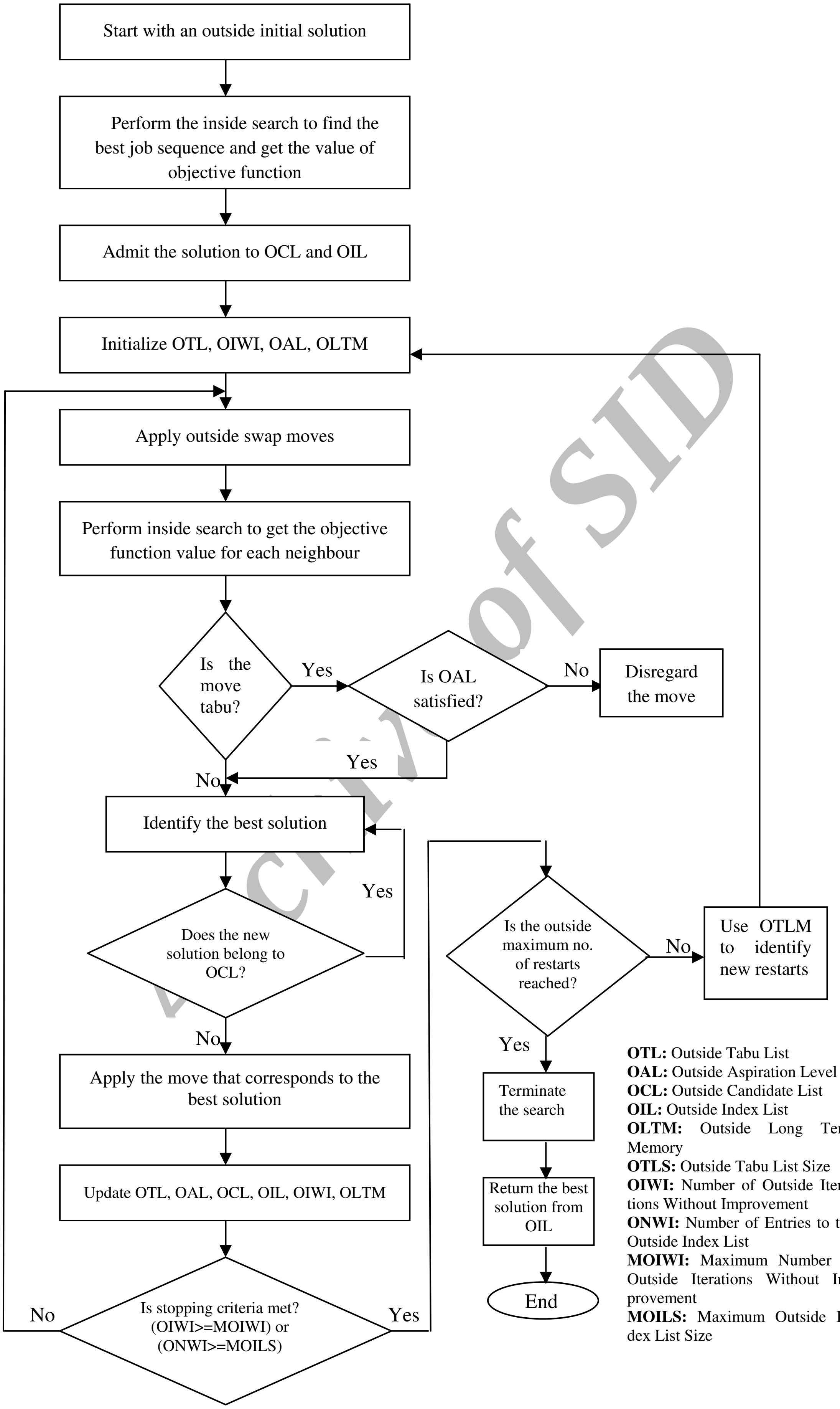
[17] Schaller, J. E., Gupta, J. N. D. and Vakharia, A. J., 1997, *Group Scheduling with Sequence Dependent Set-ups*. Proceedings of the Annual Decision Science Institute Meeting, San Diego, CA, 1141-1143.

[18] Schaller, J. E., Gupta, J. N. D. and Vakharia, A. J., 2000, Scheduling a flowline manufacturing cell with sequence dependent family setup times. *European Journal of Operational Research,* 125, 324-339.

[19] Skorin-Kapov, J. and Vakharia, A. J., 1993, Scheduling a flow-line manufacturing cell: A Tabu Search approach. *International Journal of Production Research,* 31(7), 1721-1734.

[20] Vakharia, A. J., Schaller, J. E. and Gupta, J. N. D., 1995, *Designing and Scheduling Manufacturing Cells*. Proceeding of the INFORMS National Meeting, New Orleans, LA.

Appendix 1. Flow chart for tabu search



OTL: Outside Tabu List
OAL: Outside Aspiration Level
OCL: Outside Candidate List
OIL: Outside Index List
OLTM: Outside Long Term Memory
OTLS: Outside Tabu List Size
OIWI: Number of Outside Iterations Without Improvement
ONWI: Number of Entries to the Outside Index List
MOIWI: Maximum Number of Outside Iterations Without Improvement
MOILS: Maximum Outside Index List Size

Appendix 2. The parameter values for TS for different problems

	Index list			Iterations without improvement			Tabu list size		
	Number of groups (G)		Parameter value/formula	Number of groups (G)		Parameter value/formula	Number of groups (G)		Parameter value/formula
	From	To		From	To		From	To	
Outside search for two machines	2	3	2	2	9	G*1.25	2	10	(G/4)+1
	4	6	G*3	10	16	G*2	11	15	(G/4)+2
	7	10	G*10				16	16	5
	11	16	G*50						
	From	To		From	To		From	To	
Outside search for three and six machines	2	3	2	2	2	1	2	12	(G/5)+1
	4	4	G	3	5	(G/2)+1	13	15	(G/4)+1
	5	6	G*2	6	12	G	16	16	(G/4)
	7	9	G*10	13	16	12			
	10	16	G*50						
	Job Numbers								
	From	To		From	To		From	To	
Inside search for three and six machines	2	30	2	2	29	1	2	64	1
	31	80	3	30	39	2	65	120	2
	81	120	4	40	49	3			
				50	59	4			
				60	79	5			
				80	99	6			
				100	120	7			

Appendix 3. The ANOVA table for the two machine problem by considering minimization of makespan

Effect	Num DF	Den DF	F Value	Pr > F	Effect	Num DF	Den DF	F Value	Pr> F
G	2	0	434444	<.0001	G*J*R1	8	0	1976.72	0.9140
J	2	0	122878	<.0001	G*J*A	8	135	0.58	0.7895
R1	2	0	16974.6	0.0488	G*J*I	4	135	1.24	0.2962
A	2	135	5.56	0.0048	G*R1*A	8	135	0.18	0.9937
I	1	135	0.46	0.4975	G*R1*I	4	135	1.93	0.1083
G*J	4	0	14163.4	0.0445	G*A*I	4	135	0.48	0.7482
G*R1	4	0	4021.26	0.5347	J*R1*A	8	135	0.53	0.8285
G*A	4	135	2.69	0.0339	J*R1*I	4	135	0.64	0.6352
G*I	2	135	1.81	0.1671	J*A*I	4	135	1.02	0.4005
J*R1	4	0	7523.18	0.2300	R1*A*I	4	135	0.18	0.9460
J*A	4	135	0.90	0.4682	G*J*R1*A	16	135	0.89	0.5868
J*I	2	135	4.37	0.0145	G*J*R1*I	8	135	1.07	0.3889
R1*A	4	135	0.55	0.7023	G*J*A*I	8	135	0.77	0.6318
R1*I	2	135	0.72	0.4869	G*R1*A*I	8	135	0.39	0.9229
A*I	2	135	0.51	0.5992	J*R1*A*I	8	135	0.43	0.9036
					G*J*R1*A*I	16	135	0.38	0.9844

Appendix 4. The ANOVA table for the three machine problem by considering minimization of makespan

Effect	Num DF	Den DF	F Value	Pr>F	Effect	Num DF	Den DF	F Value	Pr >F
G	2	0	360.42	.0001	J*R1*A	8	405	0.74	0.6580
J	2	0	57.00	<.0001	J*R1*I	4	405	1.63	0.1654
R1	2	0	16.96	<.0001	J*R2*A	8	405	0.48	0.8678
R2	2	0	10.54	<.0001	J*R2*I	4	405	5.85	0.0001
A	2	405	14.97	<.0001	J*A*I	4	405	0.45	0.7754
I	1	405	1.20	0.2732	R1*R2*A	8	405	0.36	0.9430
G*J	4	0	3.87	0.0063	R1*R2*I	4	405	0.92	0.4542
G*R1	4	0	1.74	0.1490	R1*A*I	4	405	0.26	0.9014
G*R2	4	0	2.56	0.0444	R2*A*I	4	405	0.37	0.8271
G*A	4	405	2.77	0.0270	G*J*R1*R2	16	0	0.81	0.6686
G*I	2	405	4.58	0.0108	G*J*R1*A	16	405	0.69	0.8090
J*R1	4	0	0.46	0.7651	G*J*R1*I	8	405	1.60	0.1242
J*R2	4	0	0.16	0.9576	G*J*R2*A	16	405	0.17	0.9999
J*A	4	405	0.67	0.6122	G*J*R2*I	8	405	3.36	0.0010
J*I	2	405	0.94	0.3911	G*J*A*I	8	405	0.78	0.6205
R1*R2	4	0	0.58	0.6813	G*R1*R2*A	16	405	0.51	0.9402
R1*A	4	405	1.14	0.3354	G*R1*R2*I	8	405	3.73	0.0003
R1*I	2	405	4.71	0.0095	G*R1*A*I	8	405	0.17	0.9951
R2*A	4	405	0.90	0.4623	G*R2*A*I	8	405	0.32	0.9589
R2*I	2	405	1.04	0.3546	J*R1*R2*A	16	405	0.31	0.9953
A*I	2	405	0.15	0.8649	J*R1*R2*I	8	405	1.65	0.1078
G*J*R1	8	0	0.28	0.9720	J*R1*A*I	8	405	0.13	0.9979
G*J*R2	8	0	0.13	0.9979	J*R2*A*I	8	405	0.53	0.8365
G*J*A	8	405	0.33	0.9539	R1*R2*A*I	8	405	0.11	0.9990
G*J*I	4	405	6.37	<.0001	G*J*R1*R2*A	32	405	0.33	0.9998
G*R1*R2	8	0	0.94	0.4906	G*J*R1*R2*I	16	405	2.14	0.0063
G*R1*A	8	405	0.21	0.9898	G*J*R1*A*I	16	405	0.33	0.9943
G*R1*I	4	405	0.45	0.7722	G*J*R2*A*I	16	405	0.22	0.9995
G*R2*A	8	405	0.90	0.5130	G*R1*R2*A*I	16	405	0.48	0.9569
G*R2*I	4	405	5.01	0.0006	J*R1*R2*A*I	16	405	0.67	0.8243
G*A*I	4	405	0.20	0.9399	G*J*R1*R2*A*I	32	405	0.53	0.9837
J*R1*R2	8	0	0.39	0.9253					

Appendix 5. The ANOVA table for the six machine problem by considering minimization of makespan

Effect	Num DF	Den DF	F Value	Pr >F	Effect	Num DF	Den DF	F Value	Pr > F
G	2	0	253.77	<.0001	G*J*R1	8	0	0.83	0.5810
J	2	0	6.97	0.0036	G*J*A	8	135	0.75	0.6474
R1	2	0	270.48	<.0001	G*J*I	4	135	2.00	0.0987
A	2	135	8.41	0.0004	G*R1*A	8	135	0.54	0.8264
I	1	135	0.94	0.3344	G*R1*I	4	135	4.09	0.0037
G*J	4	0	3.99	0.0114	G*A*I	4	135	0.46	0.7668
G*R1	4	0	37.48	<.0001	J*R1*A	8	135	0.31	0.9610
G*A	4	135	4.02	0.0041	J*R1*I	4	135	3.42	0.0107
G*I	2	135	5.57	0.0047	J*A*I	4	135	0.08	0.9873
J*R1	4	0	2.34	0.0804	R1*A*I	4	135	0.51	0.7270
J*A	4	135	0.35	0.8408	G*J*R1*A	16	135	0.45	0.9662
J*I	2	135	0.82	0.4443	G*J*R1*I	8	135	3.08	0.0032
R1*A	4	135	1.02	0.3992	G*J*A*I	8	135	0.18	0.9935
R1*I	2	135	0.37	0.6915	G*R1*A*I	8	135	0.58	0.7968
A*I	2	135	0.75	0.4763	J*R1*A*I	8	135	0.37	0.9336
					G*J*R1*A*I	16	135	0.21	0.9995