Cell Formation Problem: An Multithreading Tabu Search for Setup Time Optimization for Limited Machine Magazines

A New Solution for a Classical Problem

Arthur T. Gómez, Felipe R. Ferrary, José V. Canto dos Santos and Leonardo Chiwiacowsky
Universidade do Vale do Rio dos Sinos – PIPCA, Av. Unisinos, 950, São Leopoldo – RS, Brazil

Keywords: Tabu Search, Machine Setup Time Optimization, Manufacturing Cell Formation.

Abstract: This paper aims to present a solution for the Manufacturing Cell Formation Problem through the use of a multithreading Tabu Search that uses deterministic methods to effectively explore local optimum areas. Manufacturing Cell Formation problem involves the process of analysing parts and groups them according to their similarity. This paper aims maximize the production efficiency, by minimizing the machine setup time in a limited magazine size through the reduction of tool changes by creating clusters of parts that share machining tools and present an initial scheduling based on tool changes reduction. In order to valid the proposed algorithm, the results obtained are compared against other Tabu Search solutions proposed in the literature.

1 INTRODUCTION

The process of batching and automate manufacturing processes is an essential process for companies that want to make competitive products. The Batch manufacturing is estimated to be the most common form of production, constituting more than 50% of the total manufacturing activity in the US. In addition, there is a growing need to make batch manufacturing more efficient and productive (Groover, 2001).

In the batching process, Group Technology (GT) has an important role and offers relevant contribution helping to increase the efficiency of production operations and reducing the requirement of facilities (Xu et al., 2014). As part of GT, the Manufaturing Cell Formation problem aims to define an efficient struture to group machines (James et. al., 2007).

The Manufaturing Cell Formation problem is a NP-Hard (Spiliopoulos and Sofianopoulou, 2008). Thus, simple heuristics has the propensity to not present satisfactory results. Therefore, several methods making use of artificial intelligence techniques are proposed to solve the problem. Being the manufacturing cell formation a problem of combinatorial optimization, metaheuristic like Ant Colony (Li et al., 2011; Spiliopoulos and Sofianopoulou, 2008), Genetic algorithms (Xiaodan et al., 2007), Tabu Search (Gómez et al., 2011) are commonly used to find a good solution. On this paper the metaheuristic Tabu Search is used to solve the problem.

Beyond that, another resource that currently receives increasing focus on academic papers is the parallel processing applied to these methods (Fiechter, 1994; He et al., 2005). In such cases, different processors can perform multiple calculations simultaneously.

Another method gaining increased attention is the hybridization of metaheuristics (Kaur and Murugappan, 2008). The success of the hybrid approach comes from the union factor of the strategic advantages of each method in a single metaheuristic; providing a better performance compared to the pure method (Tsai et al., 2009).

In this paper, a variation of the classical implementation of Tabu Search is proposed. Being part of this modification, concepts of hybridization and parallel programming are used to provide solutions near promising regions.

This paper is divided into 5 sections. Section 2 presents concepts of the Manufaturing Cell Formation problem as well as review from its literature. Section 3 presents details about the Tabu

Gómez A., Ferrary F., Canto Dos Santos J. and Chiwiacowsky L..
Cell Formation Problem: An Multithreading Tabu Search for Setup Time Optimization for Limited Machine Magazines - A New Solution for a Classical Problem.
DOI: 10.5220/0005063403320339
In *Proceedings of the 11th International Conference on Informatics in Control, Automation and Robotics* (ICINCO-2014), pages 332-339
ISBN: 978-989-758-039-0

Search proposed to solve the problem, also further details about the multithreading concept used on this method are provided. Section 4 presents the methods used for general testing, as well as the metric used to compare the result againt other solution found in the literature. Section 5 presents the conclusions.

2 MANUFACTURING CELL FORMATION LITERATURE REVIEW

The cellular formation for a manufacturing system is an application of the Group Technology (GT) – which is a tool to identify similar parts and group them together regarding similarities between them (Selim et al, 1998). The Cell Formation (CF) aims to reduce the setup and flow times – minimizing the inventory and manufacturing lead times (Wemmerlov and Hyer, 1989; Wu et al., 2010).

CF is a binary matrix machines versus parts which reorganizes rows and columns with the intention to group parts (part families) and machines (machine cell). The binary element in the Figure 1 represents the relationship between part and machine indicating:

- 1 – the part *p* uses the resource (or machine) *m*;
- 0 – the part *p* can't use the resource *m*;

See the example below:

Machine Number \ Part Number	1	2	3	4	5
1	0	1	0	1	1
2	1	0	1	0	0
3	0	1	0	1	0
4	1	0	1	0	0

Figure 1: Relationship between part and machine.

Then, apply the Cluster Identification Algorithm (Kusiak and Chow, 1987) to find any relationship between parts and machines – rearranging the cells and resulting in two cellular formation for manufacturing systems and two part families, as described in the Figure 2.

		FP-1		FP-2		
		1	3	2	4	5
CM-1	2	1	1			
	4	1	1			
CM-2	1			1	1	1
	3			1	1	

Figure 2: Cells created by Kusiak Algorithm.

3 PROPOSED Tabu SEARCH

Firstly proposed in 1986, Tabu Search is a method that can be used to solve different problems on combinatorial optimization environment (Glover, 1986). According with Glover (1986), Tabu Search may be viewed as a metaheuristic superimposed on another heuristic. This method uses a list in order to forbid movements ("tabu movement") that drives to solution areas already explored.

Sumanta Basu (2012) brings a brief and updated literature review about the Tabu Search being used as a problem resolution tool related to combinatorial area. On this paper, the main methods used to perform movements from one solution to another movements used to generate the neighbourhood, Tabu List size and type, search intensification and diversification, aspiration criteria as well as a couple of other relevant details pointed along the article.

The diffent methods overviewed by this article are evaluated qualitatively (through the quality of the solution) as well as quantitatively (by checking the number of times this method is used in the studied articles). The methods employed in the proposed Multithreading Tabu search take in account of the methods pointed as the most efficiant and appropriate to the resolution of combinatorial problems.

The solution quality is evaluated using by metric named "effective clustering" (Kumar and Chandrasekharan, 1990). This metric is represented by the equation(1).

$$\tau = \frac{e - e0}{e + ev} \quad (1)$$

In the given equation, *e* represents the number of 1's in the given matrix, e_v represents the number of voids (number o 0's inside the clusters in the main diagonal) e_0 is the number of exception (the number of 1's outside the main diagonal).

3.1 Initial Solution

As initial solution, cluster identification is used. This algorithm is an iterative process that selects rows and columns in a matrix in order to simultaneously create Family Parts and Machine Clusters. Kusiak and Chow (1987) have initially proposed an algorithm that identifies the similarity between manufacturing processes and creates separable groups. This method uses a binary part-machine incidence matrix A = [a_{ij}] and decomposes A into sub matrices A_1, A_2, A_n. Each sub matrix can be defined as a machine cell.

However, once this paper aims to control and reduce the number of tool changes in a limited magazine, it is necessary taking in account of the machine magazine limit. For this reason, the algorithm used on this paper to generate the initial solution is a method proposed by Gómez (Gómez, 1996) because this method considers the similarity of tools used in the manufaturing process, as well as the machine magazine limit.

Despite this kind of method hardly returns the optimal solution, it is frequently used because it points to an optimum solution. Usually an exact algorithm points to a local optimum solution instead of a global one. However, this algorithm can in some particular cases point to the optimal solution as well as it can point to a near-optimal solution that can be considered an acceptable result (Black, 2005).

3.2 Search Intensification

Proposed by (Croes, 1958) as a method to solve the travelling salesman problem, the 2-opt is widely used to modify a current solution and generate a new neighborhood until the stop criteria requirements are met (Lim, Yong, Ramli and Khalid, 2011). Also, it is used as a local search method in many other combinatorial problems (Kothari and Ghosh, 2013; Hasegawa, Ikeguchi and Aihara, 1997).

The 2-opt movement occurs through the removal of two non-adjacent parts or machines from a cluster. After the removal of these components, those parts or machines have their position exchanged thus preventing the need of perform an evaluation of a subgroup again. In addition to, 2-opt movement is also used to exchange the families orders, thus changing the current scheduling. This proccess helps in reducing the number of tool change between Family parts, consequently reducing the setup time.

The intensification occurs due the non-substantial change of only two parts or machines, thus creating a solution near of the current solution. Considering it is a simple movement, consequently it has a low computational complexity – $\theta(n^2)$ for 2-opt algorithm (Croes, 1958). For this reason, this method can be commonly find in Tabu Search implementations. In this paper, the 2-opt movement is used to generate 20 neighbours and potential solution candidates.

Another mehod used in the implementation and on search intensification is a column insertion method. In this method, as initial step an part or machine is randomly selected and then, this component is removed from its cluster. In a next step, the algoritm should find the best cluster to insert this part or machine back (Semet and Taillard, 1993).

Suppose that the Families FP-1, FP-2 and FP-3 (Figure 3(a)) are clusters that represent the current solution for a machine supports 4 tools and the part 4 that belongs to the Cluster FP-3 is removed. Considering the, algorithm should find the best Family to insert this part back aiming to minimize the associated objective function as well as its restrictions. On this particular example, once the part 4 shares resources with the FP-1 the part 4 must be moves this cluster reducing the FP-3(Figure 3(b)). In a second iteration the part 7 could be moved to FP-2 cutting the FP-3 off.

	Machine Number	FP-1			FP-2			FP-3	
		1	5	3	6	2	8	7	4
CM-1	1	1		1					
	2	1	1						
CM-2	3					1			
	4				1	1	1		
CM-3	5				1		1	1	
	6			1					1

Figure 3(a): Incidence matrix [aij] before insertion movement.

	Machine Number	FP-1				FP-2			FP-3
		1	5	3	4	6	2	8	7
CM-1	1	1		1					
	2	1	1						
CM-2	6			1	1				
	3						1		
CM-3	4					1	1	1	
	5					1		1	1

Figure 3(b): Incidence matrix [aij] after insertion movement.

When comparing both proposed intensification methods, it is possible to notice that both methods are very similar. More than this, it is possible to affirm that the insert column method is an improved 2-opt once in both cases a randon part or machine is selected. However, on insert column algorithm, when returning this part to a cluster an objective function must be respected giving always the best solution. Due the second method requires more computational resources than the 2-opt method, it is used to generate only 5 neighbours and potential solution candidates.

3.3 Search Diversification

Different from the movements proposed in order to intensify the search (usually have simpler and subtle movements), in situations where it is necessary diversify the search and then to reach a solution far from the current solution then diversification movements are used. The diversification movements should perturb the neighbourhood increasing its candidate solution range consequently reaching solutions areas not explored before.

In the proposed model, the diversification movements occur only after a pre-defined number of iterations without improve of the current value of the actual solution or the value of the objective function. By default, this value is set as half of the value proposed as stopping criteria (The stopping criteria is defined as a finite number of iteration without improvement the value of objective function).

Once the iteration counter reaches half of the stopping criteria, automatically diversification movements are made perturbing the current neighbourhood. In this paper the chosen stopping criteria is 500 rounds without improvement, consequently the diversification movements only creates solution candidates after 250 iterations. Two different methods are proposed in order to generate solution candidates.

As previously cited in this paper, the Tabu Search is a method that improves the results though several iterations using a list of movements that should not be repeated and uses these resources to avoid becoming stuck in local optima area. However, the performance and the time to obtain an optimal or near-optimal solution are strongly affected by the proposed initial solution (Liu, Xiong and Liu, 2009).

Bearing in mind that the diversification algorithm is only used after several rounds without improvements, we can assume that it is pottentially stucked in a local optimal. If so, the result hardly will be improved. For this reason, the implemented Tabu Search proposes the usage of the same deterministic method previously used to generate the initial solution again, generating a unique solution candidate that points to a new optimal or suboptimal region, region which can possibly be the global optimal solution (Black, 2005).

It is known that the deterministic algoritm used to find an initial solution has a higher complexity - $\theta(2mn)$ (Kusiak, 1987) - than the standard algorithm used during the search intensification like 2-opt ($\theta(n^2)$). Nevertheless, with the advancement from studies related to the parallel processing area in addition to technological advancement of the computer hardware, it becomes possible to use algorithms with higher complexity instead of save processing time in combinatorial problems. Thus, it is proposed the use of the modified kusiak algoritm in parallel to the Tabu Search algoritm to create a solution candidate. The Figure 4 shows the architeture used to run the initial solution method in parallel to the intensification and diversification movements. Once it is a multithread processing method, an iteration should not be finished until both processes are completely done, otherwise there is a risk of loose the synchronization between both.

The implemented alternative Kusiak method can be defined as a deterministic algorithm, being non deterministic only when selecting the initial part represented by a row in the matrix. For this reason, depending of the initial part selected, the same result matrix can be obtained more than once. It occurs every time the initial part previously selected is selected again. For this reason, the implementation of second Tabu List is proposed in order to prevent that the same initial part be selected several times. This secondary Tabu List is responsible for storing a set of initial parts that were already used. This special list size is proportional to the problem size and has a size equivalent to 10% of the prblem size.

3.4 Tabu List

When generating a collection of solutions, it can be observed that the new neighbourhood can present only worse solutions in comparison with the best solution found. The Tabu Search has resources that can prevent the method to visit these worse solutions again, as well as it has resources avoid becoming stuck in a local optimal region (Sorlin and Solnon, 2005). A solution is added to the Tabu List every time it was already visited.

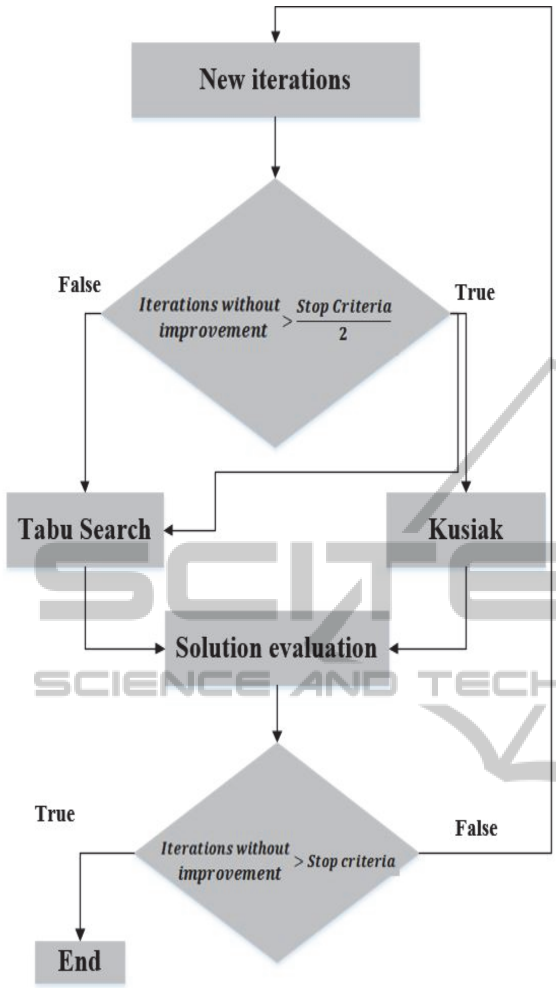

Figure 4: Multithreading Tabu Search architecture.

The Tabu Search is a method with memory structures. This memory structure is used in order to store a finite collection of movementes made during its processing. Therefore, a list is necessary to store these movementes, being this list known as the Tabu List (Glover, 1986). Bearing in mind that whether it is an finite size list, each movement added to the Tabu List must remain in this list for a limited and pre-defined time. This limitation occurs through the definition of the Tabu List Size (Basu, 2012).

The size of the Tabu List allows the user to intensify or diversify the search. The Tabu List memory types can be divided into three different categories Short-term, Intermediate-term or Long-term.

The Tabu List size can also be static, dynamic or random (Tang and Hooks, 2005). According with (Basu, 2012) survey, lists with static size are the most common and its size usually varies from 0 to 4 like as proposed in (Crainic, Gendreau, Soriano and Toulouse, 1993). On the other hand, there are articles that also present static lists, however with a larger list allowing more than 30 values (Lau, Sim and Teo, 2003). In this paper as well as presented by (Ting, Li and Lee, 2003; Greistorfer, 2003) the Tabu List size is proportional the number of parts and machines being analyzed, the chosen proportion on this implementation is 25% of the problem size.

4 EXPERIMENTS AND RESULTS

In order to test the Tabu Search implementation, a set of 11 problems extracted from the literature is used. This set of problems was previously tested by James et. al. (2007) and Gómez et. Al. (2011). The best solutions found in both papers are compared against the solutions obtained on this paper. The list of test instances was extracted from James et. al. (2007).

Table 1: Minimum, maximum and average efficiency.

ID	Problem Source	Min	Max	Average	σ
1	King and Nakornchai (1982)	82,35	82,35	82,35	0,00
2	Waghodekar and Sahu (1984)	70,00	80,00	74,08	2,44
3	Seifoddini (1989)	79,59	79,59	79,59	0,00
4	Kusiak and Cho (1992)	80,00	80,00	80,00	0,00
5	Kusiak and Chow (1987)	58,62	60,71	58,97	0,75
6	Boctor (1991)	70,37	73,08	71,18	1,31
7	Seifoddini and Wolfe (1986)	76,00	76,00	76,00	0,00
8	Chandrasekharan and Rajagopalan (1986a)	66,67	67,44	66,89	0,36
9	Chandrasekharan and Rajagopalan (1986b)	18,18	26,77	22,47	4,52
10	Mosier and Taube (1985)	70,59	76,00	71,87	2,26
11	Chan and Milner (1982)	92	92	92	0,00

Once the proposed Tabu Search uses multithreading processing – a multi core processor must be used to evaluate the processing time gain, therefore the tests were made in a computer with an Intel Core i5-3210M processor with 2.5Ghz and 6GB of RAM memory.

Each problem was submitted to 75 executions considering a machine magazine limited to 4 tools. In the Table 1 are shown the obtained: worse and best efficiency result, average efficiency result and its standard deviation. Like in James et. al. (2007) the problems are organized by ID and problem source.

An relevant point observed during the tests are the low processing time. In most part of the cases, the processing time have not exceed 2 seconds. The lowest processing time is observed on problem 1 – average processing time of 1 second and 200 milisenconds – and the biggest processing time is observed on problem 8 taking 2 seconds and 79 miliseconds. This low processing time even using the Kusiak algorithm to diversificate the search proves that with the new hardware improvements, methods with a higher complexity can be used in metaheuristics without performance reduction.

The method proved it self powerful when grouping machines that do not have to proccess more parts than the magazine limit. Once we are imposing a restriction, on this particular case imposing a magazine limit of 4 tool. Consequently, if a machine have to process more than 4 parts – it tends to create a bottleneck because this part have to be processed in more than one step. This situation is observed on problem 9, this problem has several parts that requires 5 tools on its production. For this reason, it is expected to have a result above the found on literature where this restriction is not impose.

In other hand, in problems where 4 or less tools are used during the process, like in problems 1, 2, 3, 4 and 5, the results are better or the same when compared to the literature. Table 2 shows the results in the literature in comparison to the obtained on this paper. Proposed Tabu Search represents the results found on this paper. Results better than the ones found in literature are in bold.

Though the Table 2 it is possible to observe that the solutions found to the problems 2, 4, 6, 7 and 10 are better than the solutions presented by the literature. On problems 1, 3 and 11 the same result was found. On problems 5, 8 and 9 the results found in the literature are better than the ones found by the proposed Tabu Search.

Table 2: Comparison between literature results and proposed Tabu Search results.

ID	Best Solution in source Problem	James et. al. (2007)	Gómez et. Al. (2011)	Proposed Tabu Search
1	73,68	82,35	82,35	**82,35**
2	68	69,57	69,57	**74,08**
3	79,59	79,59	79,59	79,59
4	76,92	76,92	76,92	**80,00**
5	53,13	60,87	60,87	58,97
6	70,37	70,83	70,83	**71,18**
7	68,3	69,44	69,44	**76,00**
8	85,25	85,25	85,25	66,89
9	58,72	58,72	56,70	22,47
10	72,79	75	70,35	71,87
11	92	92	92	92

5 CONCLUSIONS

This paper aims to present a solution for the Manufacturing Cell Formation Problem through the use of a multithreading Tabu Search, that uses deterministic methods to effctively explore loca optimum areas. On this paper, the magazine is considered with limited capacity. The magazine capacity considered is four tools.

The method obtained better solutions when compared against other solutions found in the literature when using 4 or less tools on the manufaturing proccess. proving to be a powerfull method to create manufaturing cells for limited number of tools or resources.

However, due the magazine limit restriction added to the problem the better results are limited only to parts that use 4 or less tools on its processing. Thus, an independent method to deal with this restriction could be implemented in order to improve the result effiiency. Consequently reducing the number of tool changes and setup time. Future works will present other studies on the problem addressed in this article.

REFERENCES

Basu, S., 2012. *Tabu Search Implementation on Traveling Salesman Problem and Its Variations: A Literature Survey*, American Journal of Operations Research. Vol. 2 No. 2, pp. 163-173.

Black, P. E., 2005. *Greedy algorithm, in Dictionary of Algorithms and Data Structures [online]*, Vreda Pieterse and Paul E. Black, eds.

Boctor, F. F., 1991. *A linear formulation of the machine-part cell formation problem*. International Journal of Production Research; 29(2):343–56.

Chan, H. M., Milner, D. A., 1982. *Direct clustering algorithm for group formation in cellular manufacture*. Journal of Manufacturing Systems; 1:65–75.

Chandrasekharan, M. P. , Rajagopalan, R., 1986. *An ideal seed non-hierarchical clustering algorithm for cellular manufacturing*. International Journal of Production Research;24(2):451–64.

Chandrasekharan M. P., Rajagopalan R., 1986. *MODROC: an extension of rank order clustering for group technology*. International Journal of Production Research; 24(5):1221–64.

Crainic, T. G., Gendreau, M., Soriano P., Toulouse, M., 1993. *A Tabu Search Procedure for Multicommodity Location/Allocation with Balancing Requirements*, Annals of Operations Research, Vol. 41, pp. 359-383.

Croes, G. A., 1958. *A method for solving traveling salesman problems*. Operations Res. 6. , pp., 791-812.

Fiechter, C. N., 1994. *A parallel tabu search algorithm for large traveling salesman problems*, Discrete Applied Mathematics, Volume 51, Issue 3.

Glover, F., 1986. Future paths for integer programming and links to artificial intelligence. Computers and Operations Research, 1, p. 533-549.

Gómez, A. T., 1996. *Modelo para o sequenciamento de partes e ferramentas em um sistema de manufatura flexível com restrições às datas de vencimento e à capacidade do magazine*, PhD thesis, Instituto Nacional de Pesquisas Espaciais, São José dos Campos, São Paulo, Brasil.

Gómez, A. T., Linck, I., Galafassi, C., Wickert, T., 2011. *A Study for Manufacturing Cell Formation Approach Considering Setup*, 8th International Conference, ICINCO 2011 Noordwijkerhout, The Netherlands, p. 43-48.

Greistorfer, P., 2003. *A Tabu Scatter Search Metaheuristic for the Arc Routing Problem*, Computers and Industrial Engineering, Vol. 44, No. 2.

Groover, M. P., 2001. *Automation, Production Systems, and Computer-Integrated Manufacturing*. Upper Saddle River, New Jersey: Prentice-Hall, USA.

Hasegawa, M., Ikeguchi, T., Aihara, K., 1997. *Combination of chaotic neurodynamics with the 2-opt algorithm to solve traveling salesman problem*. Physical Review Letters, pp. 2344-2347.

He, Y., Qiu, Y., Liu, G., Lei, K., 2005. *A parallel adaptive tabu search approach for traveling salesman problems*, Natural Language Processing and Knowledge Engineering.

James, T. L., Brown, E. C. and Keeling, K. B., 2007. *A hybrid grouping genetic algorithm for the cell formation problem*, Computers and Operations Research, 34, 2059-2079.

Kaur, D., Murugappan, M. M., 2008. *Performance enhancement in solving traveling salesman problem using hybrid genetic algorithm*, in Proceedings of the 2008 annual meeting of the North American fuzzy information processing society, New York, pp.1–6.

King, J. R., Nakornchai V., 1982. *Machine-component group formation in group technology: review and extension*. International Journal of Production Research; 20(2):117–33.

Kothari, R., Ghosh, D., 2013. *Tabu search for the single row facility layout problem using exhaustive 2-opt and insertion neighbourhoods*, European Journal of Operational Research, Volume 224, Issue 1, Pages 93-100.

Kumar, K. R., Chandrasekharan, M. P., 1990. *Grouping efficacy: A quantitative criterion for goodness of block diagonal forms of binary matrices in group technology*. International Journal of Production Research, 28(2), 233–243.

Kusiak, A., Cho, M., 1992. *Similarity coefficient algorithm for solving the group technology problem,* International Journal of Production Research; 30(11):2633–46.

Kusiak, A., Chow, W. S., 1987. *Efficient solving of the group technology problem*, Journal of Manufacturing Systems, v. 6(2), p. 117–24.

Lau, H., Sim M., and Teo K., 2003. *Vehicle Routing Problem with Time Windows and a Limited Number of Vehicles*, European Journal of Operational Research, Vol. 148, No. 3.

Li, X., Baki, M. F., Aneja, Y. P., 2011. *Flow shop scheduling to minimize the total completion time with a permanently present operator: Models and ant colony optimization metaheuristic*, Computers & Operations Research, Volume 38, Issue 1, Pages 152-164.

Lim, Y. F., Hong, P. Y., Ramli, R., Khalid, R., 2011. *An improved tabu search for solving symmetric traveling salesman problems*, Humanities, Science and Engineering (CHUSER), 2011 IEEE Colloquium on , vol., no., pp.851,854.

Liu, Y., Xiong, S., and Liu, H., 2009. *Hybrid simulated annealing algorithm based on adaptive cooling schedule for TSP*, Proceedings of GEC'09, Shanghai, pp.895-898.

Mosier, C. T., Taube, L., 1985. *The facets of group technology and their impact on implementation*. OMEGA; 13(5):381–91.

Seifoddini, H., Wolfe, P. M., 1986. *Application of the similarity coefficient method in group technology*. IIE Transactions; 18(3):271–7.

Seifoddini, H., 1989. *A note on the similarity coefficient method and the problem of improper machine assignment in group technology applications*. International Journal of Production Research; 27(7):1161–5.

Selim, M. S., Askin, R. G., Vakharia, A. J., 1998. *Cell formation in group technology: review evaluation and directions for future research*. Computers and Industrial Engineering 34 (1), 3–20.

Semet, F., Taillard, E. D., 1993. *Solving real-life vehicle routing problems efficiently using Tabu Search*, Annals of Operations Research. Vol. 41, pp. 469-488.

Sorlin, S., Christine, S., 2005. *Reactive Tabu Search for Measuring Graph Similarity*, Graph-Based Representations in Pattern Recognition, pp. 172-182.

Spiliopoulos, K., Sofianopoulou , S., 2008. *An efficient ant colony optimization system for the manufacturing cells formation problem*, International Journal of Advanced Manufacturing Technology, 36, 589-597.

Tang H., Hooks E., 2005. *A Tabu Search Heuristic for the Team Orienteering Problem*, Computers & Operations Research, Vol. 32, No. 6.

Ting, C., Li, S. and Lee, C., 2003. *On the Harmonious Mating Strategy through Tabu Search*, Information Sciences, Vol. 156, No. 3-4.

Tsai, C. W., Tseng, S. P., Chiang; M. C., Yang, C. S., 2009. *A Time-Efficient Method for Metaheuristics: Using Tabu Search and Tabu GA as a Case*, Hybrid Intelligent Systems, 2009. HIS '09. Ninth International Conference on, vol.2, no., pp.24-29.

Waghodekar, P. H., Sahu S., 1984. *Machine-component cell formation in group technology MACE*. International Journal of Production Research;22:937–48.

Wemmerlöv, U., Hyer, N. L., 1989. *Cellular manufacturing in the U.S. industry: a survey of users*, International Journal of Production Research 27(9), 1511-1530.

Wu, T. H., Chung, S. H., Chang, C. C., 2010. *A water flow-like algorithm for manufacturing cell formation problems*. European Journal of Operational Research 205(2): 346-360.

Wu, X., Chu, C.H., Wang, Y., Yue, D., 2007. *Genetic algorithms for integrating cell formation with machine layout and scheduling*, Computers & Industrial Engineering, 53, Issue 2, Pages 277-289.

Xu, Y. T., Zhang, Y., Huang, X., 2014. *Single-machine ready times scheduling with group technology and proportional linear deterioration*, Applied Mathematical Modelling, Volume 38, Issue 1, pp. 384-391.