Minimizing Total Weighted Tardiness in Identical Parallel Machine with Sequence Dependent Setup Time Using Genetic Algorithm

Sivarhubhen Sivapragasam and Yasothei Suppiah
Faculty of Engineering and Technology, Multimedia University, Melaka, Malaysia.
yasothei.suppiah@mmu.edu.my

***Abstract*—This paper considers a scheduling problem in an identical parallel machine environment to minimize total weighted tardiness with the consideration of sequence dependent setup times. As the scheduling problem is proven to be NP-hard, a genetic algorithm is developed with the aim of providing good solution in a reasonable time to the scheduling problem. Computational experiments were performed to study the effectiveness of the genetic algorithm solution quality and the CPU time. Various dispatch heuristics were developed to provide initial solutions to the genetic algorithm besides comparing their solution quality with the genetic algorithm's solution. The developed genetic algorithm has the capability to provide good results and good improvement compared to all the developed dispatching heuristics.**

***Index Terms*—Dispatching Heuristic; Genetic Algorithm; Parallel Machine; Scheduling; Tardiness.**

I. INTRODUCTION

The Scheduling is considered to be an act of making decision over arrangement of a set of activities or jobs while taking into consideration of some restricted constraints with the aim of achieving some targeted goals. Machines, manpower, and facilities are commonly assumed as critical resources in the production and service activities whereby managing these resources leads to the scheduling process with the objective of increasing efficiency, performance, utilization and finally profitability. This requires a great deal of compromises in achieving these targets. Furthermore, the existence of setup time in the scheduling process increases the complexities of the scheduling problem. Hence, decisions made in scheduling problem are very crucial and play a significant role in accomplishing the goals of the industry.

II. LITERATURE REVIEW

Identical parallel machine environment is one of the most basic machine environments available in the current industries. It is said that the job scheduling in parallel machines against their due dates is a very common setting from a practical perspective (Biskup et al. [3] & Demirel et al [6]). Industries such as the pharmaceutical industry and capital-intensive printing industry are a few from many industries that deal with processing in parallel machines (Biskup et al.[3]). This makes it interesting to consider identical parallel machine as it simulates actual environment where the findings can help improve the performance of such industries. Due-date-related problems are usually much more computationally complex and are classified as strongly NP-hard compared to scheduling problems of optimizing makespan or flow time. (Pinedo [16]). Zhu and Wilhelm [24] concluded in their review paper that due date related objectives are fertile opportunities for future research.

It is said that the effective management of sequence dependent setup is a one of the critical factors for improving the performance of a manufacturing system Krajewski [11]. In many practical production plant s such as chemical, pharmaceutical and automobile, the setup process such as cleaning up and tool replacement are sequence dependent Zandieh [22] & Roshanaei [18]. Allahverdi et al. [1], concluded in their survey paper that scheduling with setup times and cost has great potential for future research. In a recent analysis by Conner [5], half of the 250 industrial projects consist of sequence-dependent setup times. In situations where these setups are applied well, 92% of the customers due dates could be met.

As the parallel machine problem TWT problem is an NP-hard scheduling problem (Pfund et al., [15]) the solution methods for industrial size problem focus on heuristics methods in the literature. The common approach in industry is to use dispatching rules as it is the easiest way to address the parallel machine TWT problem and have been described by Pinedo [17]. With the advancement of computing systems in recent years, dispatching rules continue to be one of the most promising technologies for practical applications (Chen, et al. [4]). The static dispatching rule which requires at most O(n log n) computational time, for example, the earliest due date (EDD) rule, the shortest processing time (SPT) and the weighted shortest processing time (WSPT) are the simplest and most widely used rule. Very often the comparisons of dispatching rules are between the WSPT and EDD for the TWT problem. These can be seen in the work of Vepsalainen and Morton [20], Morton and Pentico [14], Huegler and Vasko [9] and Volgenant and Teerhuis [21] and many others. Regardless of producing fast solutions to the scheduling problems, the dispatching rules are known to be myopic and the solution quality is naturally much inferior compared to the optimal solutions (Pfund et al., [15]).

Genetic algorithm (GA) is another prominent metaheuristic in the scheduling literature which have been developed by Holland [8]. It is a search process simulating the natural evolutionary process. Starting with a current population of possible solutions to the scheduling problem, the best solutions are allowed to produce new children by the process of mutation and crossover in the aim of providing better generations that meet the goal or the objective of the scheduling. This approach has been found to quickly generate

good solutions for a wide variety of scheduling problems (Schaller, [19]). Some successful applications of GA can be found in Malve & Uzsoy [13], Zhou et al. [23], Behnamian et al. [2] , Demirel [6], Lin et al. [12] and Schaller [19]. Zhou et al. [23] proposed a hybrid GA which can be viewed as a general approach that is capable of solving a variety of scheduling problems without major redesign. In a very recent article, Joo & Kim [10] developed a hybrid GA with the combination of dispatching rule for the unrelated parallel machine and production availability. The objective of this problem is to determine the allocation policy of jobs and the scheduling policy of machines to minimize the total completion time. To solve the problem, a mathematical model for the optimal solution is derived, and hybrid GAs with three dispatching rules are proposed for large-sized problems.

III. METHODOLOGY

A. Problem Statement

The scheduling environment in this project deals with scheduling of jobs in an identical parallel machines setting. There is a set of ***N*** independent jobs waiting to be processed in the machines. Each job is characterized by its due date, d_i, weight, w_i, and processing time, p_i. The processing time of a job is the same in both machines.

Furthermore, the scheduling problem takes into account of sequence dependent setup times. When a job ***j2*** is processed after job ***j1***, then a setup time s_{1112} incurred, in which $s_{1112} \neq s_{1211}$. The setup time is solely dependent on the jobs ***j1*** and ***j2*** and is independent of the machine.

The aim of this scheduling problem is to find a good sequence of jobs that minimizes total weighted tardiness value which is denoted by:

$$\sum_{i=1}^{N} w_j T_j \tag{1}$$

Where T_1 is the tardiness of the jobs. As the completion time of each job provided by the processing schedule is C_1, therefore the tardiness is defined by:

$$T_i = \max(0, C_i - d_i) \tag{2}$$

B. Dispatch Heuristics

This class of algorithms arranges jobs on a list according to some rule. The next job on the list is then assigned to the first available machine.

There are 6 dispatch heuristics developed in this paper:

1. Earliest Due Date: Jobs are processed in ascending order of their due date, d_1
2. Weighted Earliest Due Date: Jobs are processed in ascending order of: d_1/w_1
3. Shortest Processing Time: Jobs are processed in ascending order of their processing time, p_1
4. Weighted Shortest Processing Time: Jobs are processed in ascending order of: p_1/w_1
5. Longest Processing Time: Jobs are processed in descending order of their processing time: p_1
6. Weighted Longest Processing Time: Jobs are processed in descending order of their processing time: p_1/w_1

C. Genetic Algorithm (GA)

The GA developed in this paper consists of these properties:

1. Initial population
2. Crossover Process
3. Mutation Process
4. Rate of Reproduction
5. Fitness Function
6. Mating Pool Limit
7. Generations

1. Initial Solution

The dispatching heuristics solution quality and the sequence of jobs it produces in each machine will serve as an initial population to the GA.

2. Crossover Process

Crossover process simulates the actual reproduction in the natural eco system. A population of individuals is set at the initial parameters in which the individuals are placed. These individuals are paired randomly based on the number of crossover child set. The paired individuals produce an offspring by transferring and interchanging the chromosomes that is present in the parent individuals. The procedure for crossover process which is used in this paper is listed below.

a) Get stars position (or overall partitioning structure) from one parent
b) Get a randomly selected sub schedule from the same parent in step 1
c) Get remaining jobs from the other parent by making a left to right scan

It is assumed that two parents are present for crossover as shown in Figure 1. The two parents are labelled as Parent 1 and Parent 2 respectively. The crossover process procedure is explained with the aid of Figure 1.

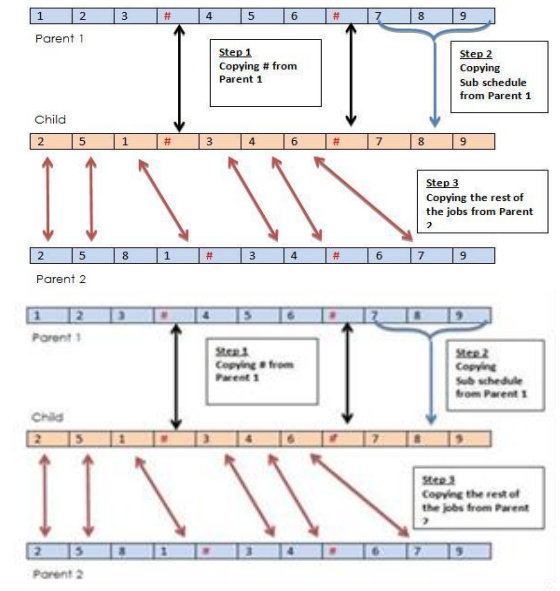

Figure 1: Crossover Process

3. Mutation Process

Mutation process does not involve two individuals such as the crossover process but only a single individual. The changes and evolutions happen within the single cell and is varied into 3 types of mutations that is explained more detail in the following parts below.

Type 1

The changes of chromosomes happen within a selected sub schedule of the sequence. Notice that two jobs were selected in Figure 2; both the jobs are interchanged while other jobs remain in their positions. This type of mutation is labelled as mutation type 1.

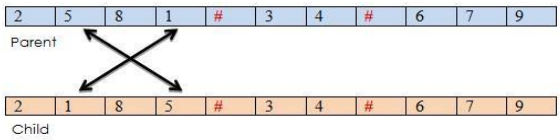

Figure 2: Mutation Process Type 1

Type 2

The type 2 mutation is similar to type 1 mutation but however with a major difference. Notice that job 8 belongs to the first sub schedule where else job 6 belongs to the last sub schedule in Figure 3. Both the jobs are interchanged and this produces the new offspring.

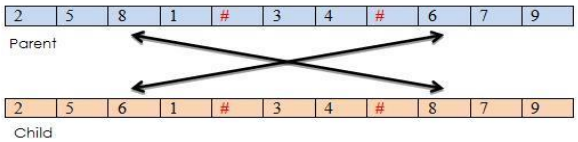

Figure 3: Mutation Process Type 2

Type 3

Type 3 mutations are the interchange of portioning position of the schedule with chromosomes. Partitioning positions is what varies the sequence in terms of sub schedule. For example from Figure 4, the jobs 2, 5,8 and 1 belong to the first sub schedule that will be processed in machine 1. The same follows for the remaining jobs. Now for type 3 mutation, the portioning position is swapped with a job as shown.

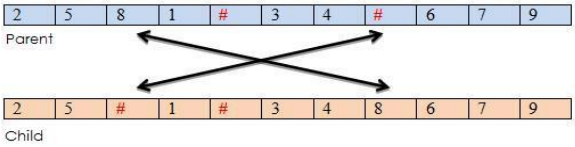

Figure 4: Genetic Algorithm Method Outline

Type 3 mutation is the only type of mutation that is allowed to swap partitioning positions. The type 3 mutations is the only function that allows the position of partitioning to be changed as this allows the GA to break free from confinement formed by the initial populations and allows evolution to take place in much wider space. In this research, all three types of mutation processes are used recursively in the order of Type 1, 2 and 3.

4. Rate of Reproduction

The rate of reproduction for GA can be separated into two categories that are named mutation rate and crossover rate. These categories determine the number of offsprings produced in each generation. The formula for both categories is listed below;

$$No\ of\ Mutation\ Child = Mutation\ Rate \times Mating\ Pool\ Limit \quad (2)$$

$$No\ of\ Crossover\ Child = Crossover\ Rate \times Mating\ Pool \quad (3)$$

5. Fitness Function

The fitness function is referred as the function that evaluates the fitness of a sequence of job and determines if the sequence is good enough to be included in the mating pool. The survival of fittest is implied as the only the fittest sequences will be selected and brought to the next generation while the rest will be discarded. The fitness function formula is given below;

$$Fitness = \frac{1}{(Total\ Weighted\ Tardiness)} \quad (4)$$

6. Mating Pool Limit

The mating pool limit means the number of sequence that is allowed to be brought forward to the next generation of evolution. These values are user determined.

7. Generations

The number of generations indicates the number of iterations that are needed for the GA to deliver its result. This would be the termination criteria of the GA algorithm as once the number of generations needed has reached than the algorithm will stop and present its obtained results. The flow chart for the GA is presented in Figure 5.

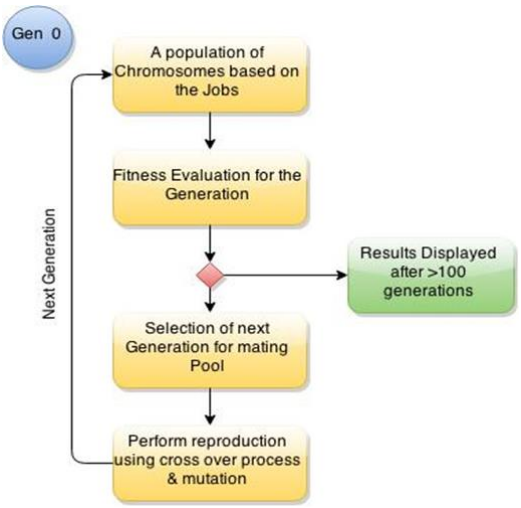

Figure 5: Genetic Algorithm Method Outline

IV. COMPUTATIONAL RESULTS

In order to test the efficiency of the developed heuristics, random instances were generated in a manner similar to Dunstall and Wirth [7] and Schaller [19]. In Table 1, the r value denotes the proportion of tardy jobs for a specific problem (25% and 50%). On the other hand, the R value denotes the extent the due dates of jobs is spread (0.5 and 1.0). More specifically, the tardiness factor r characterizes how loose or tight the due date is. Larger value of this factor contributes to tighter due dates whereas smaller values leads to loose due dates. The due date range R controls the variability of due dates. Each combination of parameters is run for 10 times, hence, a total of 200 samples was tested overall for all the developed heuristics.

Table 1
Parameter Setting

Problem Parameter	Values Used	Total Values
No of jobs ***N***	10, 20, 30, 40 and 50	5
Processing Time p_j	Uniform between [1,50]	1
Weight w_j	Uniform between [1,10]	1
Due Date d_j	d_j = Uniform [AP(1-r-(r/2)), AP(1-r+(r/2))] where AP = (Sum of all p_j)/2 Set 1: (R = 0.50); (r = 0.50) Set 2: (R = 1.00); (r = 0.50) Set 3: (R = 0.50); (r = 0.25) Set 4: (R = 1.00); (r = 0.25)	4
Setup Time s_{ij}	Uniform between [1,50]	
Total parameters combinations		20
Number of problem instances per combination		10
Total problems		200

The parameters for GA were extracted from various resources in the literature to suit to the problem nature of the scheduling. The mating pool limit was set to be 20 while the mutation rate and crossover rate were set to be 8 (Cheng et al. (1995)). The number of generations was maintained to be 150 (Demirel et al. (2011)) All these parameters were held constant throughout the testing.

The developed heuristics were later programmed in the programming language of C# using the Visual Basic Studio version 6. The computational experiments were tested on a Sony Vaio VGN-CS33G which operates on a Pentium (R) Dual Core CPU T4300@2.10GHz.

The formula used to calculate the percentage of improvement is as given below.

$$\frac{(Dispatch\ Heuristic_{TWT} - GA_{TWT})}{Dispatch\ Heuristic_{TWT}} \times 100\% \quad (5)$$

Table 2
Overall percentage (%) of improvement of GA for 10 Jobs

Set	EDD	WEDD	SPT	WSPT	LPT	WLPT	Avg[a]	BD[b]
1	70.03	51.20	66.91	50.18	76.78	81.69	66.13	WSPT
2	73.41	58.52	63.20	50.43	76.35	78.96	66.81	WSPT
3	77.75	56.83	78.93	64.59	83.07	85.33	74.42	WEDD
4	66.52	58.82	78.30	67.61	81.09	84.99	72.89	WEDD

[a]Average improvement of GA with respect to all Dispatch Rules
[b]The best Dispatch Rule in the given Set

Table 3
Overall percentage (%) of improvement of GA for 20 Jobs

Set	EDD	WEDD	SPT	WSPT	LPT	WLPT	Avg[a]	BD[b]
1	74.52	51.78	73.00	56.88	80.08	84.60	70.14	WEDD
2	70.07	55.04	67.06	59.08	78.72	82.54	68.75	WEDD
3	81.38	70.75	84.83	77.53	89.60	91.40	82.58	WEDD
4	83.03	71.16	86.18	79.30	89.79	91.62	83.51	WEDD

[a]Average improvement of GA with respect to all Dispatch Rules
[b]The best Dispatch Rule in the given Set

Table 4
20 Jobs Overall percentage (%) of improvement of GA for 30 Jobs

Set	EDD	WEDD	SPT	WSPT	LPT	WLPT	Avg[a]	BD[b]
1	77.90	62.88	73.10	61.37	82.13	85.17	73.76	WSPT
2	73.31	57.60	70.65	59.49	81.27	82.72	70.84	WEDD
3	87.38	72.31	85.35	76.92	95.27	92.60	84.97	WEDD
4	83.24	73.73	84.72	79.28	91.67	92.78	84.24	WEDD

[a]Average improvement of GA with respect to all Dispatch Rules
[b]The best Dispatch Rule in the given Set

Table 5
Overall percentage (%) of improvement of GA for 40 Jobs

Set	EDD	WEDD	SPT	WSPT	LPT	WLPT	Avg[a]	BD[b]
1	71.65	57.86	69.73	57.94	79.43	82.61	69.87	WEDD
2	73.18	55.99	71.50	61.01	81.96	83.58	71.20	WEDD
3	85.95	70.57	85.71	77.49	90.44	92.67	83.81	WEDD
4	85.06	71.61	85.29	78.82	91.15	92.88	84.14	WEDD

[a]Average improvement of GA with respect to all Dispatch Rules
[b]The best Dispatch Rule in the given Set

Table 6
Overall percentage (%) of improvement of GA for 50 Jobs

Set	EDD	WEDD	SPT	WSPT	LPT	WLPT	Avg[a]	BD[b]
1	69.69	51.13	67.35	55.80	77.90	81.60	67.42	WEDD
2	72.88	58.36	69.67	60.53	80.02	83.21	70.78	WEDD
3	81.59	65.70	81.57	71.14	88.76	90.52	79.88	WEDD
4	85.44	70.95	86.60	79.56	90.93	92.57	84.34	WEDD

[a]Average improvement of GA with respect to all Dispatch Rules
[b]The best Dispatch Rule in the given Set

Table 2 provides the average improvement of the GA for all the sets (Set 1-4 (Refer Table 1 for the combinations of R and r)) for the case of 10 jobs. All sets seem to have an average improvement of more than 66% which can be considered as a huge improvement of GA with respect to the dispatch heuristics overall for the problem 10 jobs problem.

Among the performance of the dispatch heuristics, the WSPT rule had dominated in Set 1 (tight instances) by being the best dispatch heuristic while WEDD rule for Set 4 (loose instances).

For the cases of 20-50 jobs, the average improvements of the GA to other dispatch heuristics are revealed in Tables 3-6.

From Tables 3-6, it is shown that the GA has improved the solution quality of all the samples compared to their respective dispatch heuristics for all cases. The average percentage improvement of the GA compared to the dispatch heuristics is 67.42% at the minimum while the maximum was found to be 84.97%. This proves that GA has given a good improvement to the value of the total weighted tardiness overall.

Another observation that can be made is that, among the dispatch heuristics, the WEDD rule have dominated in all the instances for all the sets except Set 1(R =0.5 ; r =0.5) and Set 2(R =1.0 ; r =0.5) in Table 2 where WSPT rule emerged to be better. Therefore it could be said that the WEDD rule outperformed other dispatch heuristics despite varying the tightness factor and due date range. On the other hand WLPT rule have proven to be the worst dispatch rule in all the instances except Set 3(R =0.5 ; r =0.25) of Table 4 where LPT rule proven to be the rule with worst performance.

The dispatch heuristics takes a very short time to produce results. It only takes at most 0.048 seconds to solve the problem of 50 jobs (which is the largest size problem). On the other hand, the CPU time for the GA falls in the range of 2.346 -3.496 seconds for all cases which is also very quick and reasonable.

V. CONCLUSION

This paper investigates the scheduling problem which exists for the identical parallel machine with the consideration of sequence dependent setup times. Our aim is to find a good schedule to minimize the total weighted tardiness. Given that the problem is NP-hard, several dispatch heuristics and genetic algorithm were developed. Besides providing initial solutions to the genetic algorithm, the

dispatch heuristics solution were compared with the solution quality of the genetic algorithm. Among the dispatch heuristics, WEDD provided the best solution quality. From the computational experiments, the genetic algorithm provided good results for all the 200 instances tested with an average improvement in the range of 66.13% - 84.97% compared to other dispatch heuristics. Future work can be in the direction of developing other metaheuristics such as tabu search, simulated annealing, etc.

REFERENCES

[1] Allahverdi, A., Ng, C.T., Cheng, T.C.E. and Kovalyov Mikhail, Y., "A Survey of Scheduling Problems with Setup Times or Costs", *European Journal of Operational Research*, vol. 187, 2008, pp. 985-1032.

[2] Behnamian, J., Ghomi, F. and Zandieh, "M. A Multi-Phase Covering Pareto-Optimal Front Method to Multi-Objective Scheduling in a Realistic Hybrid Flowshop using a Hybrid Metaheuristic", *International Journal of Production Research*, vol. 48, 2009, pp. 4949–4976.

[3] Biskup, D., Herrmann, J. and Gupta, J.N.D., "Scheduling Identical Parallel Machines to Minimize Total Tardiness", *International Journal of Production Economics*, vol. 115, no. 1, 2008, pp. 134-142.

[4] Chen, T., Rajendran, C. and Wu, C. W. "Advanced Dispatching Rules for Large-Scale Manufacturing Systems", *International Journal of Advanced Manufacturing Technology*, 2013. Doi 10.1007/s00170-013-4843-y.

[5] Conner, G., "10 questions", *Manufacturing Engineering Magazine*, 2009.

[6] Demirel, T., Özkır, V., Demirel, N. C. and Taşdelen, B., "A Genetic Algorithm Approach for Minimizing Total Tardiness in Parallel Machine Scheduling Problems",*Proceedings of the World Congress on Engineering London, U.K*, Vol II, pp. 6 – 8, July 2011.

[7] Dunstall, S. and Wirth, A., "Heuristic Methods for the Identical Parallel Machine Flowtime Problem with Set-Up Times", *Computers and Operations Research*, vol. 32, no. 9, 2005, pp. 2479-2491.

[8] Holland J.H., "Adaptation in Natural and Artificial Systems", *Ann Arbor: University of Michigan Press*, 1975.

[9] Huegler, P.A. and Vasko, F.J., "A Performance Comparison of Heuristics for the Total Weighted Tardiness Problem", *Computers and Industrial Engineering*, vol. 32, no. 4, 1997, pp. 753-767.

[10] Joo, C.M. and Kim, B.S. "Hybrid Genetic Algorithms with Dispatching Rules for Unrelated Parallel Machine Scheduling with Setup Time and Production Availability", *Computers and Industrial Engineering*, vol. 85, 2015, pp. 102-109.

[11] Krajewski, L.J., King, B.E., Ritzman, L.P. and Wong, D.S., "Kaban, MRP and Shaping the Manufacturing Environment", *Management Science*, vol. 33, 1987, pp. 39-57.

[12] Lin, Y. K., Pfund, M. E. and Fowler, J. W., "Heuristics for Minimizing Regular Performance Measures in Unrelated Parallel Machine Scheduling Problem", *Computers & Operations Research*, vol. 38, no. 6, 2011, pp. 901–916.

[13] Malve S. and Uzsoy, R., "A Genetic Algorithm for Minimizing Maximum Lateness on Parallel Identical Batch Processing Machines with Dynamic Job Arrivals and Incompatible Job Families", *Computers and Operations Research*, vol. 34, 2007, pp. 3016-3028.

[14] Morton, T. E. and Pentico, D. W., "Heuristic Scheduling Systems: With Applications to Production Systems and Project Management", *John Wiley and Sons*, 1993.

[15] Pfund, M., Fowler, J. W., Gadkari, A. and Chen, Y., "Scheduling Jobs on Parallel Machines with Setup Times and Ready Times", *Computers and Industrial Engineering*, vol. 54, 2008, pp. 764-782.

[16] Pinedo, M., "Scheduling: Theory, Algorithms and Systems", *Englewood Cliffs, NJ: Prentice Hall*, 1995.

[17] Pinedo M., "Scheduling Theory, Algorithms and Systems", *Edition 2. New Jersey : Prentice Hall*, 2002.

[18] Roshanaei, V., Seyyed Esfehani, M.M. and Zandieh, M., "Integrating Non-Preemptive Open Shops Scheduling with Sequence-Dependent Setup Times using Advanced Metaheuristics", *Expert Systems with Applications*, vol. 37, no. 1, 2010, pp. 259-266.

[19] Schaller, J.E., "Minimizing Total Tardiness for Scheduling Identical Parallel Machines with Family Setups", *Computers and Industrial Engineering*, vol. 72, 2014, pp. 274-281.

[20] Vepsalainen, A. P. J. and Morton, T. E., "Priority Rules for Job Shops with Weighted Tardiness Cost", *Management Science*, vol. 33, no. 8, 1987, pp. 1035-1047.

[21] Volgenant, A. and Teerhuis, E., "Improved Heuristics for the N-Job Single-Machine Weighted Tardiness Problem", *Computers and Operations Research*, vol. 26, no. 1, 1999, pp. 35-44.

[22] Zandieh, M., Fatemi,Ghomi, S, M, T. and Moattar Husseini, S.M., "An Immune Algorithm Approach to Hybrid Flow Shops Scheduling with Sequence-Dependent Setup Times", *Applied Mathematics and Computation*, vol. 180, no. 1, 2006, pp. 111-127.

[23] Zhou, H., Cheung, W. and Leung, L.C., "Minimizing Weighted Tardiness of Job-Shop Scheduling using a Hybrid Genetic Algorithm", *European Journal of Operational Research*, vol. 194, no. 3, 2009, pp. 637-649.

[24] Zhu, X. and Wilhelm, W. E., "Scheduling and Lot Sizing with Sequence-Dependent Setup; A Literature Review", *IIE Transactions*, vol. 38, 2006, pp. 987-1007.