

A Multi-objective Mathematical Model for Job Scheduling on Parallel Machines Using NSGA-II

Shahram Saeidi

Department of Computer Engineering, Tabriz Branch, Islamic Azad University, Tabriz, Iran

E-mail: Sh_saeidi@iaut.ac.ir

Abstract—In the current industrial world, Time and cost are two the most important concepts affecting whole our planning, activities and scheduling. Effective use of these factors, will lead to increasing performance and profit. Solving the parallel-machine problem is one of the basic and important problems in industrial and service delivery systems. In this paper, a new mathematical multi-objective linear programming model is proposed for scheduling the parallel machines to minimize the total make-span and total machines cost. The proposed model is implemented in Matlab using the NSGA-II¹ approach and the results are compared with MOPSO² approach. The computational results show the effectiveness and superiority of the proposed model.

Index Terms—Parallel Machines Scheduling, Linear Programming, NSGA-II, MOPSO.

I. INTRODUCTION

In the industrialized world today due to the decrease of production, depreciation of machinery and equipment manufacturing, energy shortages and increased costs of implementing the required production machinery and equipment, the need to optimize the use of resources and time in the industry is increasingly growing. Scheduling is one the most important topics which has raised the interest of the researchers in industrial and engineering fields. Effective and efficient planning for sequencing production is directly related to increasing the efficiency of production facilities. Scheduling problem because of its use in industry and services has great economic importance and its nonlinear nature, increases its scientific importance. So check solutions to this issue in many fields, including industrial engineering and computer science are common.

Job scheduling problem is obtaining a reasonable timetable in which certain operations assigned to particular machines, at acceptable period of time so that the technological limitations have also been respected. The sequence of the jobs on the machines has a significant impact on the volume of work in process, in a timely manner to meet customer demand and the overall cost of the product. Therefore, in the manufacturing and service-delivery systems, we try to schedule operations in

a way that the maximum performance of existing resources, such as labor or machinery is achieved. Simply, the objective is determining the job processing sequence so that the total completion time of all the jobs is minimized.

The machine scheduling problem is divided into two types: single machine, and multiple machines. The former type, schedules all the jobs on a single machine and the latter, schedules the current jobs simultaneously processes on parallel machines. These machines may be homo-gen (doing the same process) or not (have different processing capabilities). In this paper, a job shop multiple machine scheduling problems is considered under the current assumptions:

- Each job is made up of several activities. To complete a job, it is necessary to perform all of its activities. The number of job activities is different.
- Each activity can be processed only by one machine at a specific time.
- A relation between the activities of a job is a prerequisite. So if the prerequisite actions are not completed current activity cannot be started. Relations between jobs are not the prerequisite.
- The number of available machines is fixed. At any moment machine can only do one activity.
- The machines cannot be stopped during execution of a current activity (process) and start it again. A processing machine cannot transfer its current activity to another machine.
- There is no need to perform all the activities of a job done by a specific machine, so that even a machine may not be involved in executing any task.
- When an activity of a job is finished and the machine is ready for processing the next activity of the same job, the machine is bound to the implementation of new activity during the corresponding preparation time. Time to prepare for the first activity of each job is considered zero.
- The machines (and also jobs) are assumed to be as different types (machines are not homo-gen). The machines can take responsibility for processing activities, only for the jobs of the same type.
- There is a priority relation among the activities of a job. The jobs are not prior to each other. There is not any priority between activities of different jobs.
- The processing cost of each activity on a machine

¹ Non-dominated Sorting Genetic Algorithm

² Multi-Objective Particle Swarm Optimization

is also considered.

Considering the above assumptions, the main goal of the proposed model in this paper is optimizing the following objective functions:

- Minimizing the maximum total make-span.
- Minimizing total machines' costs.

The remainder of this paper is organized as follows: The literature review and previous works are discussed in section 2. The proposed mathematical model is given in section 3. Section 4 presents the proposed Genetic Algorithm (GA) using NSGAI approach; implementation and the computational results are discussed in Section 5. Conclusion and future work are given in the final section.

II. RELATED WORKS

The parallel machine scheduling problem has many applications in real environments and it has raised the researchers' interest during the past years. Due to the NP-hard nature of the problem [1], most of the past works have developed a meta-heuristic approach and have obtained acceptable or better solutions.

Seraj, Tavakkoli-Moghaddam and Julai [2] used mixed integer programming method for open-shop scheduling problem to minimize the average lateness and earliness by a multi-objective fuzzy decision making approach. They also used Taboo search for medium and large scale problems and compared the results with Lingo software. Kacem, Hammadi and Borne [3] developed new methods for flexible job shop scheduling problem consisting localization and Genetic Algorithm. They used new encoding schemes for chromosomes representation used in GA.

Naderi, Zandieh and Fatemi-Ghomi [4] proposed two general techniques which were applicable for scheduling problems too. They used these techniques for job shop scheduling problem considering sequence dependencies. The objective function was defined as total make-span of all the jobs. In this research, four heuristic methods were developed based on GA and Simulated Annealing (SA). Calhoun, Dekcro, Moore, Chrissis and Van Hove [5] used Taboo search method for solving the scheduling problem. They used Java programming language for implementing the proposed model for 100 numbers of jobs over four numbers of machines.

Sels, Coelho, Manuel Dias, and Vanhoucke [6] developed three meta-heuristics for scheduling multiple jobs on non-related parallel machines to minimize total make span. The methods were based on GA, Taboo search and a combination of them including branch and bound method.

Ak and Koc [7] in their research paper discussed the GA approach and its operators, chromosome structure design for parallel machine scheduling and flexible job shop scheduling problems. Teekeng and Tammano [8] developed a modified GA to minimize the total make-

span. Kim, Yun, Yoon, Gen, and Yamazaki [9] used a hybrid fuzzy GA to minimize the total make-span and total lateness penalty cost. Liu [10] also proposed a compound GA to minimize total lateness. He used a heuristic to assign high priority jobs to high priority machines.

Gen and Lin [11] designed a multi-objective evolutionary algorithm for solving different type of scheduling problems. They used the proposed method for solving job shop and flexible job shop scheduling problems.

Rostami, Ebrahimzadeh and Mahdavi [12] proposed a non-identical parallel machine multi-objective scheduling problem considering both the deterioration and learning effects. The processing times and due dates have been considered as uncertain fuzzy values. The authors developed a branch and Bound method for solving the proposed approach. Bandyopadhyay and Bhattacharya developed a modified evolutionary algorithm for solving multi-objective parallel machine scheduling problem [13].

Li, Amdeo, Yalaoeui and Chehade [14] also proposed a new approach for solving the parallel machine scheduling problem consisting n independent jobs on m identical parallel machines with release dates, due dates, and sequence-dependent setup times with no preemption of the jobs. The considered objective functions are minimizing total make-span and total tardiness. They used NSGA-II and SPEA-II evolutionary approaches for solving the proposed model and showed the advantages of NSGA-II approach for the studied problem.

Tavakkoli-Moghaddam, Taheri and Bazzazi [15] developed a novel, multi-objective model of a parallel machines scheduling problem that minimizes the number of tardy jobs and total completion time of all jobs. The proposed model, considers the machines as unrelated parallel units with different speeds. Besides, there is some precedence, relating the jobs with non-identical due dates and their ready times. Sequence-dependent setup times embedded in their proposed model may vary in different machines based on their characteristics.

Baesler and Palma [16] proposed a memetic algorithm combining the Genetic Evolution with local search for solving the parallel machine scheduling problem involving a molding production process in the wood industry. The algorithm was compared against four multi-objective techniques: the multi-objective genetic algorithm (MOGA), the strength Pareto evolutionary algorithm (SPEA), the non-sorting genetic algorithm II (NSGA II), and multi-objective genetic local search (MOGLS). The Authors show that their proposed approach outperformed the benchmark techniques in most of the test problems based on two objectives of industrial interest: minimization of the maximum completion time and minimization of total tardiness.

Yin, Ling, Wu and Chiang [17] proposed a high-performance algorithm for solving the multi-objective unrelated parallel machine scheduling problem. The proposed approach is based on the iterated Pareto greedy (IPG) algorithm combined with Taboo list (TL) to enhance its performance. Liu and Tsai [18] developed a

model for solving the multi-objective parallel machine scheduling problem, considering controllable processing times. In this study, the processing time of each job can be between lower and upper bounds, and a cost is associated with the processing of a job on a machine. The processing time of a job can be decreased, which may lower the cycle time, although doing so would incur additional costs.

III. THE PROPOSED MATHEMATICAL MODEL

The main objective of solving the job shop scheduling problem, is finding the best feasible solution from the problem space which satisfies all the constraints. The time needed to find such a solution, highly depends on the program complexity and dimension. The scheduling problem must be first formulated under a given assumptions and next be solved using a proper approach. In this section, the proposed mathematical model for solving the parallel machine scheduling problem under the assumptions mentioned in section 1 is formulated and described. They developed two multi-objective mathematical models, which consist of two and three inconsistent objective functions, respectively. The first model minimizes the total manufacturing cost and the total weighted tardiness simultaneously, while the second uses make-span an additional objective function.

3.1 Indexes and Parameters

The proposed model consists of the following indexes and parameters:

- I : Set of used machines.
- i : Index used for machines.
- J : Set of available jobs.
- j : Index used for jobs.
- l : Index used for processes (activities) of jobs.
- M : Total number of machines.
- N : Total number of jobs.
- P_j : Set of processes of j^{th} job.
- N_j : Number of processes of j^{th} job.
- l_j : l^{th} process of j^{th} job.
- D_{lji} : Processing time of l^{th} process of j^{th} job on i^{th} machine.
- $R_{l'ji}$: Setup time of i^{th} machine, if it will perform the process l' after process l of job j .
- C_i : Total cost of i^{th} machine.
- C_{lji} : Cost of l^{th} process on i^{th} machine.
- $CostM$: Machine investment fixed cost.
- K_i : Type of i^{th} machine.
- K_{l_j} : Type of l^{th} process of j^{th} job.
- O_{l_j} : A positive integer representing the priority of the l^{th} process of j^{th} job.
- $u_{ljl'j}$: Boolean parameter representing the precedence of the processes, equal to 1 if l^{th} process of the j^{th} job is prior to the l'^{th} process.
- ST_{lji} : Starting time of l^{th} process on i^{th} machine.

3.2 Decision Variables

The main items of a linear mathematical model are decision variables that their final values are known as the final solution. The proposed model consists of two Boolean decision variables which are defined as below:

- x_{lji} : Equal to 1, if the l^{th} process of the j^{th} job should be processed by the i^{th} machine, otherwise zero.
- $y_{ljl'ji}$: Equal to 1, if l^{th} process of j^{th} job will be processed by i^{th} machine before the l'^{th} process of the same job, otherwise zero.

3.3 Objective Functions

There are two objective functions in the proposed model defined as the following equations.

The make-span is the total length of the schedule (that is, when all the jobs have finished processing). The first objective function is defined to minimize the total make-span which is calculated by Equation 1.

$$\text{Min } T = \sum_{j=1}^n FT_j \quad (1)$$

$$FT_j = \text{Max } FT_{l_j} \quad \forall L_j \in P_j \quad (2)$$

$$FT_{l_j} = x_{lji} * (ST_{lji} + D_{lji}) \quad \forall L_j \in P_j, i \in I \quad (3)$$

The second objective function is defined equation (4) in order to minimize total machine costs which include the processing costs plus the machine investment costs.

$$\text{Min } C = \max C_i \quad \forall i \in I \quad (4)$$

$$C_i = \text{CostM} + \sum_{j=1}^n \sum_{\forall l_j \in P_j} x_{lji} * C_{lji} \quad \forall l_j \in P_j, i \in I \quad (5)$$

3.4 Constraints

The rest of the model includes some relations to guarantee the problem constraints and assumptions are considered and all obtained solutions are feasible. These relations are given as the follows.

$$\sum_{i=1}^m x_{lji} = 1 \quad (6)$$

$$\sum_{i=1}^m \sum_{\forall l_j \in P_j} x_{lji} = 1 \quad (7)$$

$$l_j \neq l'_j$$

$$y_{ljl'ji} * (ST_{l'ji} - FT_{lji}) \geq x_{lji} * x_{l'ji} * R_{ljl'ji} \quad (8)$$

$$\forall l_j, l'_j \in P_j, l_j \neq l'_j, i \in I$$

$$(O_{l_j} - O_{l'_j}) * (ST_{lji} - ST_{l'ji}) \geq 0 \quad \forall l_j, l'_j \in P_j, \quad (9)$$

$$u_{ljl'j} * (ST_{l'ji} - FT_{lji}) \geq 0 \quad \forall l_j, l'_j \in P_j, \quad (10)$$

$$\min \left(x_{l_j i} * \left(1 - \left(|k_i - k_{l_j}| \right) \right), 0 \right) \geq 0$$

$$\forall i \in I, l_j \in P_j \exists k_i, k_{l_j} \quad (11)$$

The Equation 6 enforces processes of the jobs to be process just by one machine. Equations 7 and 8 show that one machine cannot perform more than one process at a time. Equation 9 makes the processes with higher priority to be performed before other processes of the same job, in case if equal priority, this constraint will have no effect. Equation 10 demonstrates the precedence relation between processes. The type conformity obligation between jobs and machines is considered by Equation 11.

IV. THE PROPOSED GENETIC ALGORITHM

The proposed mathematical model is a non-linear programming model and besides the NP-Hard nature of the problem, it will take a long time to solve the medium and large scale problems in real environment. Hence, a multi-objective GA based meta-heuristic approach is also developed for solving the proposed model. The pseudo code of the standard NSGA-II algorithm is given in Figure 1.

The proposed GA first generates the initial population randomly. Then, the fitness of each chromosome (solution) is calculated and is sorted based to create the Pareto frontiers. Non dominating solutions are placed on a front and the better solution among them is selected using the crowding distance concept. The main loop of the program generates next populations by producing Offsprings, merging with parents, sorting them and choosing the best solutions.

```

Initialize Population
Generate random population - size M
Evaluate Objective Values
Assign Rank (level) Based on Pareto Dominance - "sort"
Generate Child Population
    Binary Tournament Selection
    Recombination and Mutation
For i = 1 to Number of Generations
    With Parent and Child Population
        Assign Rank (level) Based on Pareto - "sort"
        Generate sets of non-dominated fronts
        Loop (inside) by adding solutions to next generation
        starting from the "first" front until M individuals found
        determine crowding distance between points on each front
        Select points (elitist) on the lower front (with lower rank)
        and are outside a crowding distance
    Create next generation
        Binary Tournament Selection
        Recombination and Mutation
    Increment generation index
End of Loop
    
```

Fig.1. The NSGA-II pseudo code [18]

4.1 Chromosome Structure

The chromosomes are the main core structures of the Genetic algorithm which encode the solutions and the proper representation of them is very important to ensure

the efficiency of the algorithm. The chromosomes used in the proposed GA, are constructed from concatenating smaller chromosome structures (sub-chromosomes) for each job which are depicted in Figure 2. In the first generation, the value of the genes is non-repetitive permutation integers which are randomly assigned.

Machine 1			Machine 2			Machine 3			Machine 4		
1	3	7	4	8	2	9	5			6	

Fig.2 Sub-chromosome structure for each job

These sub-chromosomes demonstrate the machines assigned for process of a job. The length of the chromosome for job *j* is $N_j + M - 1$. The genes having values greater than N_j are used as delimiters (black fields in Figure 1). For example, assuming $N_1=9$ and $M=4$ (job #1 consists of 9 processes and four machines are available), the Figure 2 shows that processes {1,3,7} are assigned to Machine 1, {4,8,2} to Machine 2, etc.

The main chromosome structure is obtained by concatenating N numbers of sub-chromosomes which is shown in Figure 3.

Sub-chromosome of Job1	Sub-chromosome of Job2	...	Sub-chromosome of Job N
------------------------	------------------------	-----	-------------------------

Fig.3. Chromosome structure of proposed GA

4.2 The Crossover Operator

The crossing over operator is needed to generate new chromosomes from the current population. Generally, good chromosomes (solutions) with a higher probability are good candidates to be mated hoping to generate better Offsprings. In the proposed GA, a one-point crossover operator is applied for every sub-chromosome of the selected parents. This may lead to infeasible off-springs if repetitive genes occur in a sub-chromosome which is not a permutation case. In order to solve this problem, after generating the Offsprings, the proposed algorithm checks for repetitive values and substitutes them with missing numbers. The details are depicted in Figures 4, 5 and 6.

Parent 1	2	3	4	1	5	6
Parent 2	3	6	2	4	1	5

Fig.4. One-point crossover operator

Offspring 1	2	3	4	1	1	5
Offspring 2	3	6	2	4	5	6

Fig.5. Offsprings having repetitive values

The crossover position in Figure 4 is selected between genes 4 and 5 and the children chromosomes (Offsprings) are generated. As it can be seen in Figure 5, the gene with value "1" is repetitive in offspring 1 and the value "6" is missing. The second Offspring also has a similar problem. This issue is solved by replacing the repetitive genes with missing value and is represented in Figure 6.

Offspring 1	2	3	4	6	1	5
Offspring 2	3	1	2	4	5	1

Fig.6. Modified Offsprings having repetitive genes

4.3 The Mutation Operator

The second operator applied on some chromosomes of the population is mutation. This operator randomly selects a chromosome and exchanges its value (the machine assigned to a process of a job) with another value (machine) randomly. This exchange is applied on every sub-chromosome of the selected chromosome by changing the gene values. The feasibility of the mutated chromosome is always checked and in case of generating bad chromosomes, the changes are omitted. The probability of mutation is defined as 30 percent.

4.4 Calculating the Fitness

Calculating the value of the objective function(s) is the main part of the optimizing algorithms. The objective function computes the fitness (quality) of the obtained chromosomes (solutions). In the proposed GA algorithm, the fitness of each chromosome is computed in two points of view: time and cost. The details of each are described below.

4.5 Time Fitness

To calculate the fitness of a chromosome, first it is decomposed to sub-chromosomes and genes are decoded to identify which processes of the jobs are assigned to which machines. When the machines assigned to processes of a job are distinguished, next it will be checked that whether the precedence relations between processes is maintained or not. If the answer is positive, the completion time of each job is calculated. The greatest (maximum) job completion time (make-span) is chosen as the first objective function value.

4.6 Cost Fitness

The calculation of the cost function is performed in a similar manner at the same time. While calculating the process time of the jobs, the computation cost is also considered and calculated. The maximum computed machine cost is chosen for the second objective function value.

4.7 The NSGA-II Implementation

The main ideas in multi-objective NSGA-II approach are dominance and crowding distance. Dominance is a public concept that is used in many multi-objective approaches. The solution x dominates solution y , if the value of all objectives for x solution are better (or at least equal to) than that of y solution. In this case, the x solution exists on a better Pareto Frontier than y and absolutely should be preferred. Otherwise, x neither dominates nor is dominated by y solution which means both solutions are on the same frontier (like x and z solutions is Figure 7). In this case, the crowding distance concept helps the decision makes to choose better solution [12].

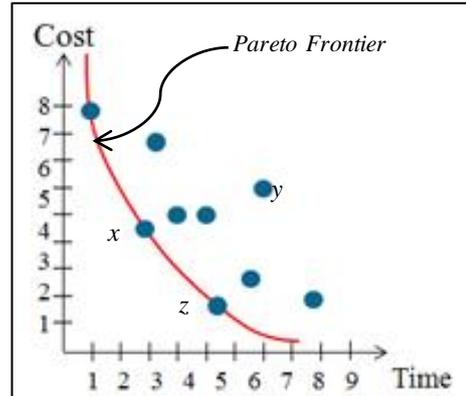


Fig.7. The dominance concept

The crowding distance is a criterion for comparing the solution on the same frontier. Naturally, solutions covering larger spaces of the Pareto frontier are better than the solution with smaller coverage because the probability of ignoring other solutions is less. This criterion is defined as the largest rectangular consisting the current solution which does not contain other solutions [12]. This concept is depicted in Figure 8.

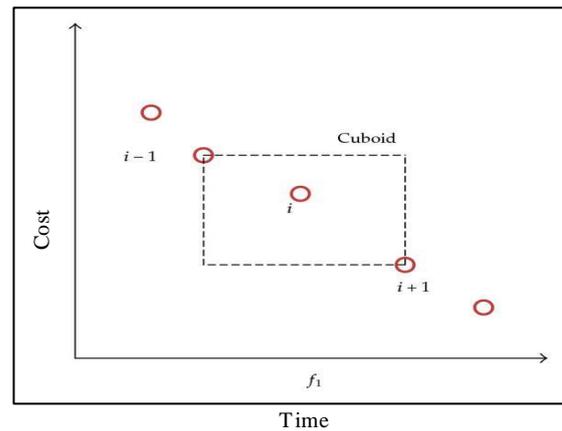


Fig.8. The solution i has a larger crowding distance than its neighbor solutions

V. COMPUTATIONAL RESULTS

The proposed model is implemented in Matlab on a PC running Windows 7 with 4.2 GHz processor. The simulation parameters and the specification of test cases are given in Table 1 and Table 2 respectively.

Table 1. The simulation parameters

Parameter	Value
Population Size	50
Number of Iterations	200
Crossover Probability	0.8
Mutation Probability	0.3

Table 2. Specification of implemented examples

Example #	Number of Jobs	Number of Machines
1	5	5
2	5	10
3	10	5

Each of the examples listed in Table 2 have been executed 30 times and the minimum completion time, average of minimum completion times, minimum cost and the average of minimum costs are shown in Table 3.

Table 3. Obtained results of running the examples for 30 times obtained by NSGA-II approach

	Minimum Completion Time	Average of Minimum Completion Times	Minimum Cost	Average of Minimum Costs
Example 1	139	141.2	537	543.6
Example 2	115	120.4	332	341.6
Example 3	425	439.4	966	982.3

The simulation results for one of the final solution selected from the Pareto frontier of Example #1 are depicted in Figures 9 and 10. The Figure 9 shows the Gantt chart of the processing time of the activities of the jobs (horizontal axis) and assigned machines (vertical axis).

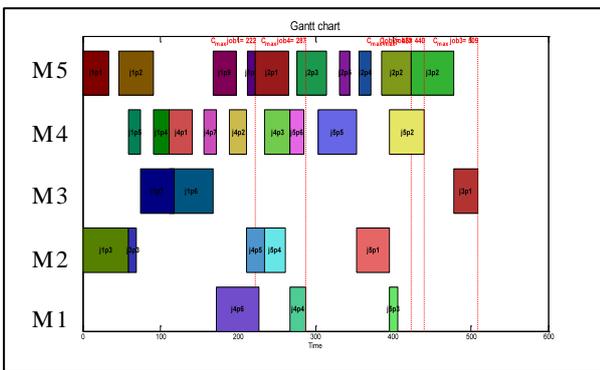


Fig.9. Gantt chart of final solution obtained for Example 1

The cost Gantt chart for the same solution is illustrated in Figure 10.

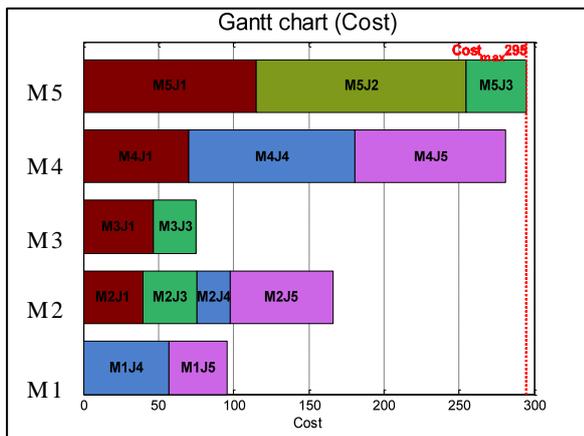


Fig.10. Gantt chart of machine cost of the same solution

The proposed model is also implemented using MOPSO approach. The obtained results are given in Table 4. Comparing the results obtained by two approaches, shows that NSGA-II method has a slightly better performance in solving the proposed algorithm

Table 4. Obtained results of running the examples for 30 times obtained by MOPSO approach

	Minimum Completion Time	Average of Minimum Completion Times	Minimum Cost	Average of Minimum Costs
Example 1	142	146.7	566	581.1
Example 2	117	123.1	335	341.0
Example 3	433	440.9	977	981.5

VI. CONCLUSION

In this paper a new mathematical linear programming model for scheduling the jobs in a parallel environment to minimize the total machine cost and the completion time is proposed and due to NP-Hard nature of the problem and the complexity of the proposed algorithm, a multi-objective meta-heuristics based on NSGA-II and MOPSO are developed for solving the model. The proposed method is implemented in Matlab R2009a (in a PC running Microsoft Windows 7 with 2 GB of RAM and 4.2 GHz main processor) and tested for three case studies of different sizes. The obtained results show the efficiency of the proposed mathematical model and the developed GA algorithm. Besides, the better performance of NSGA-II in comparison with MOPSO approach for solving the proposed model is identified.

REFERENCES

- [1] J. Blazewicz, J.K. Lenstra, A.H. G. Kan, Scheduling Subject to Resource Constraints: Classification and Complexity. *Discrete Applied Mathematics*, 1983, 5(1), 11-24.
- [2] O. Seraj, R. Tavakkoli-Moghaddam, F. Jolai, A Fuzzy Multi-objective Tabu-search Method for a New bi-objective Open Shop Scheduling Problem. Paper presented at the Computers & Industrial Engineering, 2009. CIE 2009. International Conference on
- [3] I. Kacem, S. Hammadi, P. Borne, Approach by Localization and Multiobjective Evolutionary Optimization for Flexible Job-shop Scheduling Problems. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions*, 2009, 32(1), 1-13.
- [4] B. Naderi, M. Zandieh, S.M.T. Fatemi-Ghomi, S. Scheduling Sequence-dependent Setup Time Job Shops with Preventive Maintenance. *The International Journal of Advanced Manufacturing Technology*, 2009, 43(1-2), 170-181.
- [5] M. Calhoun, K., R.F. Deckro, J.T. Moore, J.W. Chrissis, J.C. Van Hove, J. C.. Planning and Re-planning in Project and Production Scheduling. *Omega*, 2002, 30(3), 155-170.
- [6] V. Sels, J. Coelho, A. Manuel Dias, M. Vanhoucke, Hybrid Tabu Search and a Truncated Branch-and-bound for the Unrelated Parallel Machine Scheduling Problem. *Computers & Operations Research*, 2015, 53, 107-117.
- [7] B. Ak, E. Koc, . A Guide for Genetic Algorithm Based on Parallel Machine Scheduling and Flexible Job-Shop

- Scheduling. *Procedia - Social and Behavioral Sciences*, 2012, 62, 817-823.
- [8] W. Teekeng, A. Thammano, Modified Genetic Algorithm for Flexible Job-Shop Scheduling Problems. *Procedia Computer Science*, 2010, 12, 122-128.
- [9] K. Kim, Y. Yun, J. Yoon, M. Gen, G. Yamazaki, Hybrid Genetic Algorithm with Adaptive Abilities for Resource-constrained Multiple Project Scheduling. *Computers in Industry*, 2005, 56(2), 143-160.
- [10] C. Liu, A Hybrid Genetic Algorithm to Minimize Total Tardiness for Unrelated Parallel Machine Scheduling with Precedence Constraints. *Mathematical Problems in Engineering*, 2013, 11.
- [11] M. Gen, L. & Lin, L. Multiobjective Evolutionary Algorithm for Manufacturing Scheduling Problems: state-of-the-art survey. *Journal of Intelligent Manufacturing*, 2014, 25(5), 849-866. doi: 10.1007/s10845-013-0804-4
- [12] M. Rostami, A. Ebrahimzadeh, M. Mahdavi, Multi-objective parallel machine scheduling problem with job deterioration and learning effect under fuzzy environment. *Computers and Industrial Engineering*, 2015, 85, 206-215.
- [13] S. Bandyopadhyay, R. Bhattacharya, Solving multi-objective parallel machine scheduling problem by a modified NSGA-II. *Applied Mathematical Modeling*, 2013, 37(10-11), 6718-6729.
- [14] X. Li, L. Amdeo, F. Yalaoeui, H. Chehade, A Multi-objective Optimization Approach to Solve a Parallel Machines Scheduling Problem. *Advances in Artificial Intelligence*, 2010, Article ID 943050, 10 pages, 2010. doi:10.1155/2010/943050.
- [15] R. Tavakkoli-Moghaddam, F. Taheri, M. Bazzazi, Multi-Objective Unrelated Parallel Machines Scheduling with Sequence-Dependant Setup Times and Precedence Constraints. *IJE Transactions*, 2008, 21 (3), 269-278.
- [16] F. Baesler, C. Palma, Multi-objective parallel machine scheduling in the sawmill industry using memetic algorithms. *The International Journal of Advance Manufacturing Technology*, 2014, 74 (5), 757-768.
- [17] Sh. Yin, K. Ling, W. Wu and Y. Chiang, Multi-objective unrelated parallel machine scheduling: a Tabu-enhanced iterated Pareto greedy algorithm. *International Journal of Production Research*, 2015, DOI: 10.1080/00207543.2015.1047981.
- [18] Ch. Liu, W. Tsai, Multi-objective parallel machine scheduling problems by considering controllable processing times. *Journal of the Operational Research Society*, 2015, doi:10.1057/jors.2015.82.
- [19] C.C. Coello, D.A.V. Veldhuizen, Lamont G.B. Evolutionary Algorithms for solving Multi-Objective problems. Springer Science and Business Media, 2013.

How to cite this paper: Shahram Saeidi, "A Multi-objective Mathematical Model for Job Scheduling on Parallel Machines Using NSGA-II", *International Journal of Information Technology and Computer Science (IJITCS)*, Vol.8, No.8, pp.43-49, 2016. DOI: 10.5815/ijitcs.2016.08.05

Authors' Profiles



Shahram Saeidi: Assistant professor at computer engineering department, Faculty of Engineering, Islamic Azad University, Tabriz branch. Holding Ph.D. and M.Sc. in Industrial engineering and B.Sc. in computer engineering.