A Genetic Algorithm Approach for Minimizing Total Tardiness in Parallel Machine Scheduling Problems

Tufan Demirel, Vildan Özkır, Nihan Çetin Demirel, and Belgin Taşdelen.

***Abstract*—This study investigates parallel machine scheduling problem in order to minimize total tardiness and we developed a genetic algorithm solution procedure for such problems. Also, using problem specific knowledge, an efficient solution improvement scheme and an appropriate crossover operator are developed and integrated into the genetic algorithm.**

***Index Terms*— Genetic algorithms, Scheduling, Parallel Machines, Total Tardiness.**

I. Introduction

SCHEDULING problems involve obtaining the optimal schedule under various circumstances including various types of objectives, machine environments and job characteristics. The common objectives, which are often studied in the literature, are the minimization of total (weighted) completion time, (weighted) mean flow time, mean waiting time, total (weighted) lateness, tardiness, and make-span.

In most of today's industries, meeting the due dates is very important to sustain competitive strength and scheduling jobs on parallel machines against their due dates become a very common setting from a practical perspective [6]. A variety of optimizing criteria and objectives are defined in the literature to determine the most efficient and effective parallel machine schedule. Minimizing total tardiness is the most interesting criteria for production systems, especially in the current situation where competition is becoming more and more intensive [31].

Parallel machine scheduling problem includes scheduling a set of independent jobs, say $N = \{1,2 \dots, n\}$, ona set of identical parallel machines, say $M = \{1,2 \dots, m\}$, while minimizing an objective function such as total weighted job tardiness, maximum completion time, the maximum tardiness or total weighted flow time. The minimizing total tardiness machine scheduling problem for single machine is proved to be strongly NP-hard by Du and Leung [7] and similarly, with the same objective, the parallel machine scheduling problem is extremely NP hard [15,18,19].

Manuscript received March 4, 2011; revised March 22, 2011.

Tufan Demirel is with the Yildiz Technical University, Department of Industrial Engineering, 34349, Istanbul-Turkey (Phone: 090 212 3832867 e-mail: tdemirel@yildiz.edu.tr).

Vildan Özkır is with the Yildiz Technical University, Department of Industrial Engineering, 34349, Istanbul-Turkey (e-mail: cvildan@yildiz.edu.tr).

Nihan Çetin Demirel is with the Yildiz Technical University, Department of Industrial Engineering, 34349, Istanbul-Turkey (e-mail: nihan@yildiz.edu.tr).

Belgin Taşdelen is with the Yildiz Technical University, Department of Industrial Engineering, 34349, Istanbul-Turkey.

II. Literature

In the literature, evolutionary and search algorithms are widely used for solving the machine scheduling problems. Many authors have illustrate that genetic algorithm (GA) performs well for solving the scheduling problems [20,31, 33] as well as for parallel machine scheduling problems [2, 28].

Min and Cheng [21] study on the application of GA in solving identical parallel machine scheduling problem for minimizing the makespan and developed machine-code based genetic algorithm method which performs better for large scale scheduling problems. Zhiming and Chunwei [34] present a GA approach to solve the job shop scheduling problems in real-time cases. Mönch et al. [24] investigate two different approaches (batching before job assignment / assigning jobs before batching) based on GA for scheduling jobs with incompatible job families and unequal ready times on parallel machines while minimizing the total weighted tardiness. Chang et al. [8] develop a two phase subpopulation GA for solving the multi-objective parallel machine scheduling problem. First phase includes the decomposition of the population into subpopulations which is designed for a scalar multi-objective whereas the second phase combines modest solutions following the first phase and unifies all subpopulations as one big population. Min and Cheng [22] introduce three different heuristics based GA methods to determine the optimal common due date and the optimal scheduling policy for minimizing the total cost of assignment of due date, earliness and tardiness and discussed the efficiency of the methods for parallel machine earliness/tardiness scheduling problems. Chang et al. [9] illustrate the efficiency of the proposed method, which is a combination of mining gene structure technique and the subpopulation GA, in solving multi objective flow shop scheduling problems. They proposed a gene mining procedure for creating artificial chromosomes to strengthen the solution convergence effectively by helping GA to search better solution spaces and find better solution. Balasubramanian et.al. [5] propose bi-criteria GA for scheduling of interfering jobs on identical parallel machines where jobs belong to two disjoint sets; the makespan criterion needs to be minimized for one of the sets, while the total completion time needs to be minimized for the other. Another multi-objective parallel machine scheduling

ISBN: 978-988-19251-4-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

problem is solved by Moghaddam et al. [23] who propose two-level 0-1 linear programming model to minimize the number of tardy jobs and the total completion times, considering a set of jobs that have non-identical due dates and ready times with some precedence relations on a set of unrelated parallel machines.

III. PROBLEM DESCRIPTION

The problem considered in this paper can be formally described as scheduling n independent jobs $N = \{1,2,\dots,n\}$ on m identical parallel machines $M = \{1,2,\dots,m\}$ within a time window $[r_i, d_i]$.

In order to minimize the total tardiness time of scheduling grouped jobs on identical parallel machines, the optimal solution satisfies following conditions:

–All jobs are ready at time zero

TABLE I
PARAMETERS AND INDEX OF THE PROBLEM

Symbol	Definition
M	Number of machines
N	Number of jobs
Nm	Population Number
i,j	Job index
k	Machine index
S_i	Setup time for job i
p_i	Process time for job i
t_i	type of job i
d_i	due date of job i

–Each job has only one operation and cannot be processed on different machines at the same time.

–Placement and transportation time are not considered.

–Each machine can only process one job at the same time and machines can be idle.

–Jobs can wait for processing and each job has the same priority.

Technological constraints are constant and the number of the jobs, the number of the machines, the process times, and job set-up times are all known. We handle the scheduling problem that job preemption is not allowed and machine breakdown is not considered.

IV. THE MODEL STRUCTURE

In order to maintain long-term customer satisfaction, it is essential for companies to be reliable and consistent on delivering the orders on the exact due dates. In many cases, there is no benefit for customers in case of any delay and that may result unsatisfactory and loss the orders [23].

In this paper, we propose a zero–one MIP model for scheduling jobs on parallel machines to minimize total tardiness. Based on the definition and notations described above, the objective function can be formulated as follows:

$$\begin{aligned}\text{Minimize } Z &= \sum_{i=1}^{N} L_i \\ &= \sum_{k=1}^{M}\left[\sum_{i=1}^{index}\left(d_{i,k} - C_{i,k}\right)\right] \\ &= \sum_{k=1}^{M}\left[\sum_{i=1}^{index}\left(d_{i,k} - \left(C_{i,k} + P_{i,k} + v_{k,i}\,S_i\right)\right)\right]\end{aligned} \quad (1)$$

In the above formulation, objective function represents the minimization of the total tardiness which can be shown in three different forms.

To ensure that any job should be scheduled and can only be scheduled once, we formulate the following constraint in the following:

$$\sum_{k=1}^{M}\sum_{i=1}^{index} u_{(k,i,x)} = 1 \qquad i = 1,2,\dots,N \quad (2)$$

To ensure each position on the list can hold only one job at the same time, we consider the constraint as shown below.

$$\sum_{k=1}^{M} u_{(k,i,x)} = 1 \qquad i = 1,2,\dots,N \quad (3)$$

We need to reveal that the first job should have setup time on each machine.

$$v_{(k,1)} = 1 \; if \;\; u_{(k,i,x)} = 1 \qquad k = 1,\dots,M \quad (4)$$

A setup time is required when a job is followed by another job from a different group and vice versa.

$$v_{(k,i)} = \begin{cases} 0 & t_{i-1} = t_i \\ 1 & t_{i-1} \neq t_i \end{cases} \qquad i = 1,\dots,N \quad (5)$$

Finally, we need to define the characteristics of local variables.

$$\begin{matrix} v_{(k,i)} \in \{0,1\} \\ u_{(k,i,x)} \in \{0,1\} \end{matrix} \qquad \begin{matrix} i = 1,2,\dots,N \\ k = 1,2,\dots,M \\ x = 1,\dots,index \end{matrix} \quad (6)$$

where the local variables, say $u_{(k,i,x)}$ and $v_{(k,i)}$, can be defined as following:

$$u_{(k,i,x)} = \begin{cases} 1, & \text{job } i \text{ is the } x^{th} \text{ position of machine } k \\ 0, & \text{otherwise} \end{cases}$$

$$v_{(k,i)} = \begin{cases} 1, & t_{i-1} \neq t_i \text{requires setup time} \\ 0, & t_{i-1} = t_i \text{no setup time} \end{cases}$$

In order to solve the identical parallel machine scheduling problem, highly sophisticated branch and bound algorithms are presented recently by Yalaoui and Chu [31], Shim and Kim [29] and Tanaka and Araki [30]. Since the problem is NP hard, optimal solutions are often not available even for mid-size problems. Therefore, we focus on developing a GA based solution procedure to yield efficient solutions in terms of computation time, applicability and solution quality.

ISBN: 978-988-19251-4-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

V. PROPOSED ALGORITHM

Since genetic algorithm is proved to be an effective approach for solving optimization problems, there are many situations where the simple GAs does not perform particularly well [10]. Therefore, many authors (as mentioned in Section 2) developed different kinds of operators and defined various additional processes for GAs to perform better solutions in terms of solution quality and computational time. In this study, we present an additional mechanism to embed in GA for solving parallel machine scheduling problem with setup times.

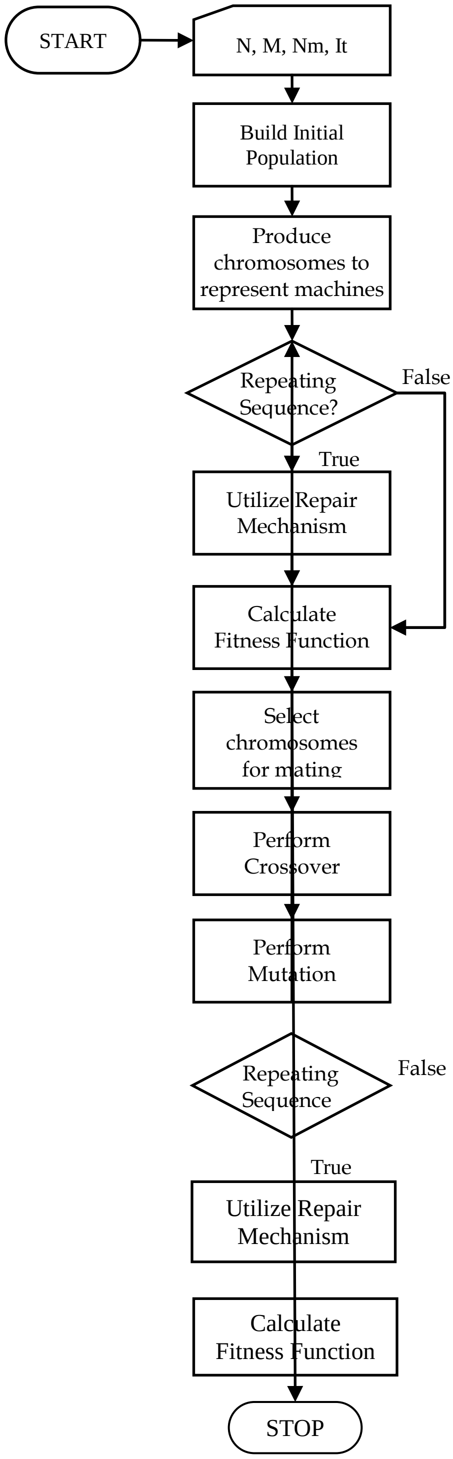

Figure 1 Algorithm

The first step involves supplying data for job definitions, machines, population parameters to the proposed model. Then, the initial population of chromosomes is created randomly and each chromosome is partitioned to represent each machine schedule. After initial repair phase, we calculate the fitness function which has already been the objective function of minimizing total tardiness. Fitness function is related to minimization of total tardiness while scheduling same type of two jobs to the same machine and eliminating the set up time of latter job. Then, selection operation is utilized to produce successive populations with better fitness values. In selection phase, roulette wheel and Monte Carlo simulation methods are used. These methods help us to select chromosomes randomly while avoiding local optimal solutions. In addition to the selection operation, we defined a rule which is used to save the successive solution sequence which has the best fitness value and to transfer this chromosome to the next generation without any operation, as an elitism procedure. As for crossover function, circular crossover is selected in order to avoid repeating sequence, shorten the solution procedure, and reduce the complexity by simplifying the control mechanism. Mutation operation is utilized to help scheduling by randomly changing different jobs to same machine as well as changing the jobs sequence on the same machine. The proposed algorithm reports each iteration results until the stopping criterion is satisfied.

VI. RESULTS AND DISCUSSION

In previous section, we present a kind of genetic algorithm including repair mechanism. We coded this algorithm in C# programming language. The proposed algorithm is executed for randomly generated parallel machine scheduling problems with varying number of both machines and jobs. The number of jobs varies between 5 and 15 and the number of machines varies between 2 and 4.

The parameters of crossing over and mutation are set to 0,05 and 0,01 respectively. The number of population varies problem to problem. We obtained the results by limiting the iteration number by 50 and 100 iterations.

Results show that there is a parallel trend between number of jobs and total tardiness. If the number of jobs increases, total tardiness of parallel machine scheduling problem increases too. The increase in population and number of iterations lets us observe working of algorithm better. Also it is noticed that problems are generated randomly and the results should be considered in this way.

The results of the proposed method indicate these events:

When the total number of jobs is low;

ISBN: 978-988-19251-4-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

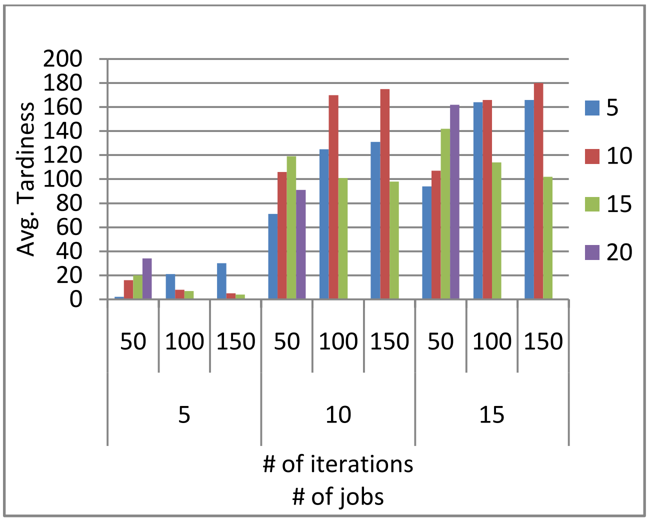

Figure 2 Average tardiness with respect to varying population sizes

- Proposed method yields better solution for lower iteration numbers and lower population sizes.

- Proposed method yields better solution for higher iteration numbers and higher population sizes.

When the iteration number is high;

- Proposed method yields better solutions for all problem sizes and population sizes. It is observed that it has positive effects on quality of the results when number of iterations increases.

Finally, two repair mechanisms is embedded to genetic algorithm for solving parallel machine scheduling problems. This repair mechanism presents an efficient approach to producing efficient chromosomes.

REFERENCES

[1] Anghinolfi, D. and Paolucci, P. (2007), Parallel machine total tardiness scheduling with a new hybrid metaheuristic approach, Computers and Operations Research, 34(11), 3471-3490.

[2] Abdekhodaee A. H., Wirth A., and Gana H. S., Scheduling two parallel machines with a single server: the general case, Computers & Operations Research 33, 994–1009.

[3] Allahverdi, A., Gupta, J.N.D., and Aldowaisan, T., (1999), A review of scheduling research involving setup considerations. OMEGA The International Journal of Management Sciences 27, 219–239. In: Allahverdi A., Ng C.T., Cheng T.C.E., and

[4] Allahverdi A., Ng C.T., Cheng T.C.E., and Kovalyov M.Y., (2008), A survey of scheduling problems with setup times or costs, European Journal of Operational Research 187, 985–1032.

[5] Balasubramanian H., Fowler J., Keha A., and Pfund M., (2009), Scheduling interfering job sets on parallel machines, European Journal of Operational Research, 199(1), 55-67.

[6] Biskup, D., Herrmann, J. and Gupta, J.N.D., (2008), Scheduling identical parallel machines to minimize total tardiness, International Journal of Production Economics, 115(1), 134-142.

[7] Du, J. and Leung, J.Y.T., (1990),Minimizing total tardiness on one machine is NP-hard, Mathematics of Operations Research, 15(3), 483-494.

[8] Chang P.C., Chen S.H., Lin K.L., (2005), Two-phase sub population genetic algorithm for parallel machine-scheduling problem, Expert Systems with Applications 29, 705–712.

[9] Chang, P.C.,Chen, S.H., Liu, C.H., (2007) Sub-population genetic algorithm with mining gene structures for multi objective flowshop scheduling problems, Expert Systems with Applications 33, 762–771.

[10] Cheng, R. and Gen, M., (1997), Parallel machine scheduling problems using memetic algorithms, Computers & Industrial Engineering, 33(3-4), 761-764.

[11] Chiang, T-C, Cheng, H-C, and Fu, L-C., (2010), A memetic algorithm for minimizing total weighted tardiness on parallel batch machines with incompatible job families and dynamic job arrival, Computers & Operations Research, 37(12), 2257-2269.

[12] Cochran J. K., Horng S. M., ve Fowler J. W., (2003), A multi-population genetic algorithm to solve multi-objective scheduling problems for parallel machines, Computers & Operations Research 30, 1087–1102.

[13] Figielska E., (2009), A genetic algorithm and a simulated annealing algorithm combined with column generation technique for solving the problem of scheduling in the hybrid flowshop with additional resources, Computers & Industrial Engineering 56, 142–151.

[14] Graham, R.L., E.L. Lawler, J.K. Lenstra, and A.H.G. RinnooyKan,(1979), Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey, Annals of Dis. Math. 5, 169-231.

[15] Ho, J.C. and Y.L. Chang, (1991), Heuristics for minimizing mean tardiness for m parallel machines, Naval Research Logistics, 38, 367–381.

[16] Jenabi M., Ghomi S.M.T. F., Torabi S.A., and Karimi B., (2007), Two hybrid meta-heuristics for the finite horizon ELSP in flexible flow lines with unrelated parallel machines, Applied Mathematics and Computation 186, 230–245.

[17] Jou C., (2005), A genetic algorithm with sub-indexed partitioning genes and its application to production scheduling of parallel machines, Computers & Industrial Engineering 48, 39–54.

[18] Karp, RM. (1972), Reducibility among combinatorial problems: complexity of computer computations. New York: Plenum Press, 85–103.

[19] Koulamas, C., (1997), Decomposition and hybrid simulated annealing heuristics for the parallelmachine total tardiness problem, Naval Research Logistics, 44, 109–125.

[20] Martino V. D., and Mililotti M., (2004), Sub optimal scheduling in a grid using genetic algorithms, Parallel Computing 30, 553–565.

[21] Min L., and Cheng W., (1999), A genetic algorithm for minimizing the makespan in the case of scheduling identical parallel machines, Artificial Intelligence in Engineering 13, 399–403.

[22] Min L., and Cheng W., (2006), Genetic algorithms for the optimal common due date assignment and the optimal scheduling policy in parallel machine earliness/tardiness scheduling problems, Robotics and Computer-Integrated Manufacturing 22, 279–287.

[23] Moghaddam R. T., Taheri F., Bazzazi M., Izadi M., and Sassani F., (2009), Design of a genetic algorithm for bi-objective unrelated parallel machines scheduling with sequence-dependent setup times and precedence constraints, Computers and Operations Research, 36 (12), 3224-3230.

[24] Mönch L., Balasubramanian H., Fowler J. W., and Pfund M. E., (2005), Heuristic scheduling of jobs on parallel batch machines with incompatible job families and unequal ready times, Computers & Operations Research, 32, 2731–2750.

[25] Liu, M. and Wu, C., (2003), Scheduling algorithm based on evolutionary computing in identical parallel machine production line, Robotics and Computer-Integrated Manufacturing ,19(5), 401-407.

[26] Peng J., ve Liu B., (2004), Parallel machine scheduling models with fuzzy processing times, Information Sciences 166, 49–66.

[27] Rossi A., and Boschi E., (2009), A hybrid heuristic to solve the parallel machines job-shop scheduling problem, Advances in Engineering Software 40, 118–127.

[28] Serifoglu F. S., and Ulusoy G., (1999), Parallel machine scheduling with earliness and tardiness penalties, Computers & Operations Research 26, 773-787.

[29] Shim, S. and Kim, Y., (2007), Scheduling on parallel identical machines to minimize total tardiness, European Journal of Operational Research, 177(1), 135-146.

[30] Tanaka, S. and Mituhiko,A., (2008), A branch-and-bound algorithm with Lagrangian relaxation to minimize total tardiness on identical parallel machines, International Journal of Production Economics, 113(1), 446-458.

[31] Yalaoui, F., and Chu, C., (2002), Parallel machine scheduling to minimize total tardiness, International Journal of Production Economics, 76(3), 265-279.

[32] Vallada E., Ruiz R., and Minella G., (2008), Minimizing total tardiness in the m-machine flowshop problem: A review and evaluation of heuristics and metaheuristics, Computers & Operations Research 35, 1350 – 1373.

[33] Zhiming W., and Chunwei Z., (2000), Genetic algorithm approach to job shop scheduling and its use in real-time cases, Int. J. Computer Integrated Manufacturing, Vol. 13, No. 5, 422-429.

[34] Zhou H., Cheung W., and Leung L. C., (2009), Minimizing weighted tardiness of job-shop scheduling using a hybrid genetic algorithm, European Journal of Operational Research 194, 637–649.

ISBN: 978-988-19251-4-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)