A Genetic Algorithm Approach for Scheduling Jobs on Identical Parallel Machines

Dr. Pravin P. Tambe*
Operations Management & Decision Sciences
Indian Institute of Management Tiruchirappalli, India
pravin@iimtrichy.ac.in, tambepp@gmail.com

Abstract

Scheduling is an essential function in production management. Scheduling determines what is going to be made, when, where and with what resources. Parallel machines scheduling problems have many industrial applications. This paper presents a hybrid approach using a genetic algorithm to investigate the identical parallel machine scheduling problem with an objective to minimize the tardiness penalty cost. A numerical example is presented to demonstrate the approach.

Keywords
Parallel machines, Genetic algorithm. Schedule Penalty Cost

1. Introduction

Scheduling is a decision making process that plays an important role in most manufacturing industries, as it is one of the key factors of improving the manufacturing productivity. Effective production scheduling can improve machine utilization, reduce inventory, reduce lead time and thus improve the on-time delivery. Scheduling in manufacturing systems is typically associated with allocating a set of jobs on a set of machines in order to achieve some objectives.

The parallel machine scheduling problem is an important and difficult problem faced in the industrial world. Traditionally, the problem consists of scheduling a set of independent jobs on identical parallel machines (processors) with the aim of certain performance measure like minimizing maximum job completion, lateness, tardiness penalty, etc. For example, in a bakery there are multiple ovens for baking, in case of printing industry, a publisher will need multiple identical printing facilities otherwise the publisher will lose revenues if a special title (in the worst case the actual bestseller) is not available on the shelves of the bookstores, the pharmaceutical industry usually runs parallel machines to produce drugs, and many more. In line with current trends towards just-in-time manufacturing strategies, where both early and tardy finishing of job processing are undesired, objectives related to earliness and tardiness penalties have become increasingly popular.

There are two essential issues to be dealt for all kind of multiple machine scheduling problems:

1. Partition jobs to machines.
2. Sequence jobs for each machine.

The combinatorial nature of most scheduling problems allows the use of search based and enumerative techniques, such as genetic algorithms, branch and bound, simulated annealing, tabu search, etc. These methods usually offer good quality solutions. Scheduling with fast heuristic algorithms is therefore highly effective, and the only feasible solution, in many instances.

Genetic Algorithms are very effective at performing global search for combinatorial optimization problems. This paper presents a hybrid approach using a Genetic algorithm combined with backward forward heuristic approach (Sule, 2007) for scheduling a set of independent jobs on identical parallel machines with earliness-tardiness, non-common due date scheduling problem. For earliness penalty the inventory holding cost for the batches is considered.

The numerical example of schedulingbatches on three identical parallel machines is presented. The computational results for different problem sizes and schedule tightness factor are also presented.

The rest of this paper is organized as follows: Section 2 presents the literature review. In section 3 problem descriptions is presented. Section 4, describes the proposed genetic algorithm approach and in Section 5 results are presented. Finally, in Section 6 conclusion and future work is discussed.

2. Literature Review

A variety of optimizing criteria and objectives are defined in the literature to determine the most efficient and effective parallel machine schedule. Min and Cheng (1999) 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. Balasubramanian et.al. (2009) proposed 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. Balin(2011) attempted to adapt a GA to the non-identical parallel machine scheduling problem and proposed an algorithm with a new crossover operator and a new optimality criterion. Zhiming and Chunwei(2000) presented a GA approach to solve the job shop scheduling problems in real-time cases. Mönch et al. (2005) investigated 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. Min and Cheng (2006) introduced 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. (2005) presented 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. Demirel et al. (2011) investigated parallel machine scheduling problem in order to minimize total tardiness and developed a genetic algorithm solution procedure for such problems. Also, using problem specific knowledge, an efficient solution improvement scheme and an appropriate crossover operator were developed and integrated into the genetic algorithm. Bilgesu and Koc (2012) analyzed a Parallel Machine Scheduling (PMS) and Flexible Job-shop Scheduling (FJS) problems. They presented a PMS and FJS chromosome structure, crossover and mutation operator from literature in order to guide for new researchers about scheduling with GAs. Biskup et al. (2008) considered the problem of scheduling a given number of jobs on a specified number of identical parallel machines to minimize total tardiness. They proposed a new heuristic approach which is general enough for solving several important types of parallel-machine scheduling problems. They also developed and evaluated an efficient heuristic algorithm for finding optimal or near-optimal schedules for the identical parallel-machine problem to minimize total tardiness. Computational results of empirical experiments involving the proposed and the three best available heuristics are used to identify the most effective heuristic algorithms for minimizing total tardiness. Vallada and Ruiz (2011) proposed a genetic algorithm that includes the crossover operator with a limited local search as well as a fast local search procedure; this method was tested on both small and large problem sets and outperformed the other evaluated methods. Eroglu et al.(2014) proposed a genetic algorithm that was based on random keys. The results showed that the *GA,* which is the foundation of this study, outperformed the other algorithms.Dang et al. (2020) presented the problem of scheduling a set of jobs with tool requirements on identical parallel machines in a work center. They proposed a mathematical model for the problem and a new matheuristic that combines a genetic algorithm and an integer linear programming formulation.

3. Problem Description

The parallel machine manufacturing system is shown in Figure 1. The scheduling decisions involved in parallel machine problem are:

i) Which machine processes each job?
ii) In what sequence the jobs to be processed on the machine.

The problem considered in this paper can be formally described as scheduling n independent jobs N={1,2,…,n} on m identical parallel machines M={1,2,…,m} in order to minimize the total tardiness time of scheduling grouped jobs on identical parallel machines

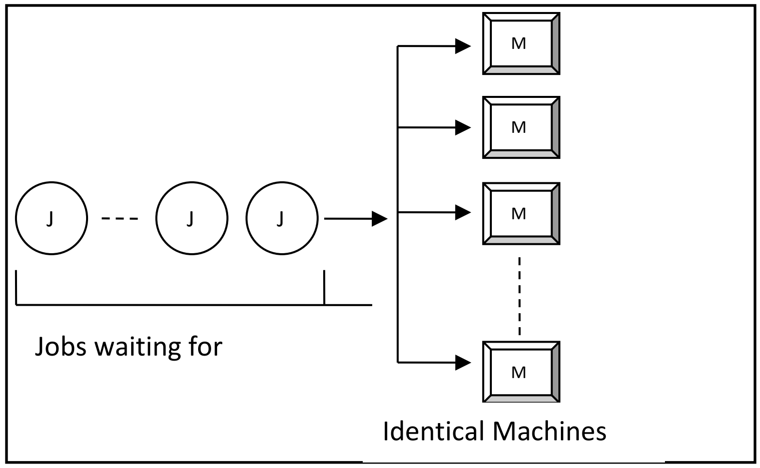

Figure1. Parallel machine scheduling problem

The production unit under study consists of 'm' identical machines producing different product type. We address a batch scheduling problem where a set of identical parallel machines is available for processing the jobs in batches in the continuous mode. Formally, there is a set of jobs n available from time zero onwards for processing on m identical parallel machines in batches. Each batch 'k' has a processing time and a due date; before which the batch is expected to complete. The batches are to be grouped for processing with a sequence-independent setup time; the jobs in the same batch are continuously processed after the setup operation and the processing length of a batch is the sum of the setup time and the processing times.

Following assumptions are made for the batch scheduling problem.

1. Each machine has the ability for processing each job.
2. The batch is available at time zero.
3. The machine can process no more than one batch at a time.
4. Each batch is independent of each other.
5. A batch cannot be pre-empted by another batch.
6. The setup time is included in the processing time for the job.

Let P_k denote the processing time of the batch 'k', let DD_k denote the due date; before which the batch is expected to complete and W_k be the penalty cost for the batch k, if it is late. Also, letCT_k denote the completion time of the kth batch, where k=1,2,……….n, and 'n' is the number of jobs to be processed on the jth machine; j=1,2 ………m.

The lateness of the job can be defined as,

$$\mathrm{LT_k} = CT_k - DD_k \quad (1)$$

and the tardiness penalty of the kth job is,

$$T_k = W_k \times \{max[0, LT_k]\} \quad (2)$$

Let 'ST'$_k$ denote the start time of the kth batch in the production schedule. Then, the inventory holding cost is calculated as follows:

i) If $DD_k \geq CT_k$, then

$$IHC_k = \begin{Bmatrix} \int_0^{(CT-ST)} (CT-ST-t)\times PR\times Ch\times dt + \\ (DD-CT)\times PR\times Ch\times (CT-ST) \end{Bmatrix}_k \quad (3)$$

ii) If $DD_k < ST_k < CT_k$

$$IHC_k = \begin{Bmatrix} \int_0^{(CT-ST)} (CT-ST-t)\times PR\times Ch\times dt \end{Bmatrix}_k \quad (4)$$

iii) If $ST_k < DD_k < CT_k$

$$IHC_k = \begin{Bmatrix} \int_0^{(DD-ST)} (DD-ST-t)\times PR\times Ch\times dt + \\ \int_0^{(CT-DD)} (CT-DD-t)\times PR\times Ch\times dt \end{Bmatrix}_k \quad (5)$$

The objective of production scheduling is to find an optimal sequence of batch such that total cost of batch delay and inventory holding cost for all jobs is minimized. The total schedule penalty cost is given by,

$$\text{Schedule Penalty Cost} = \sum_{j=1}^{M}\sum_{k=1}^{n} W_k \times (\max[0, LT_k]) + IHC_k \quad (6)$$

where 'n' is the number of batches on the allocated machine.
Therefore, the objective function is to minimize the total penalty cost.
Next section presents the proposed genetic algorithm approach.

4.0 Genetic Algorithm

Genetic Algorithm (GA) was developed and introduced by Holland (1975). A genetic algorithm (GA) is a search technique that imitates the natural selection and biological evolutionary process. GA has been used in a wide variety of applications, particularly in combinatorial optimization problems and they were proved to be efficient to provide the near optimal solutions in a reasonable time. Genetic algorithm works on the principle of survival of the fittest by progressively accepting better solutions to the problems. It operates on the population of potential solutions to the problem and iterates towards the optimum solution. During the iterations of the search process, the population includes the fitter solutions converging towards the near optimality. The basic elements of genetic algorithm are solution representation, population, evaluation (Fitness), selection, crossover and mutation. The algorithm starts with a randomly generated initial set of population consist of chromosomes, which represent the solution of the problem. The chromosomes are evaluated for the fitness function and selected according to their fitness value. The crossover operator is the primary operator of GA and the generation of new chromosomes is handled by the crossover operator.

After generating an initial population, selection, crossover and mutation will be iteratively used to search for the best solution. For implementation, the important steps of the GA algorithm are explained below.

1. Solution encoding

Before applying a genetic algorithm on any problem, a method is needed to encode the potential solution to the problem in a form so that the algorithm can be implemented using a computer program. Since the problem is to schedule the batches, a permutation encoding, where a chromosome is a string of numbers that represent the position in a sequence is used. The string also contains the partitioning points which decide the jobs on the machine. For example, any chromosome for a ten job, three machine scheduling problem can be represented as, 3 6 2 1 # 8 4 10 # 5 9 7. Here the partitioning points are 4 3, 3 which allocates the jobs on the machine.

2. Fitness evaluation

The sum of earliness and tardiness penalty cost of the schedules on each machine is calculated and the total penalty cost(Equation (6) for all the machines is taken as the fitness function for the evaluation of the chromosomes.
Selection

A roulette wheel selection is used for forming the mating pool of chromosomes to further take part in the crossover and mutation operation of the GA. In this method the chance of an individual offspring being selected is proportional to its fitness value.

3. Crossover

The generation of new chromosomes takes place by a crossover operator. The crossover function takes two chromosomes and combines them to form new offsprings. For a permutation encoding problem, commonly used crossover operators are order crossover, partially matched crossover (PMX), cyclic and edge recombination. In the present case an order crossover approach with some modification to apply to the parallel machine scheduling problem is used. During crossover, the schedule for one machine is selected randomly from the first parent and copied to the one offspring. Remaining jobs in the same offspring are copied from the other parent. However, the same partitioning structure for jobs on the machine in the parent is taken for the offspring. Similar approach is used for forming the other offspring.

4. Mutation

The mutation operator is used to prevent convergence to a local optimum. In the present work,a swapping rule is used for mutation. Two gene positions in the chromosomes are selected randomly and then the jobs at the gene positions are exchanged.

5. Results and discussion

In this section, we present the numerical experiment based on the procedure mentioned in the previous section

5.1 Numerical Example:

We consider a numerical example of scheduling 9 batches on 3 machines. The data for scheduling problem; processing time, due date, penalty cost. etc. is given in Table 1. The inventory holding cost is taken as 20 % of the unit holding cost per year.

Table 1. Data for scheduling

Batch	Processing Time (hr.)	Due date (hr.)	Penalty Cost(Rs.)	Inventory holding Cost (Rs./unit/Yr.)
1	16	72	240	0.0012
2	29	131	135	0.0010
3	16	72	180	0.0012
4	27	122	248	0.0016
5	15	68	180	0.0010
6	22	99	270	0.0013
7	24	108	248	0.0017
8	21	95	660	0.0016
9	20	90	203	0.0014

The parameters selected for implementation of the GA are, an initial population of 50 chromosomes, the crossover probability of 0.8, mutation probability of 0.02 and number of generation as 100. The algorithm is implemented using a program developed in Matlab R2012a. The results obtained after completing the generation are given in Table 2.

Further, we present a numerical experiment for different job sizes and number of machines. The due dates in the problems are set for two schedule tightness factor; loose (0.3) and tight (0.7). The results of allocations of jobs on machines with total penalty cost using the proposed GA are shown in Table 3. The results are compared with solutions obtained using LEKIN® scheduling software (system developed at the Stern School of Business, NYU, available at http://web-static.stern.nyu.edu/om/software/lekin/). The results indicated the better performance of the proposed algorithm Table 3.

Table 2. Scheduling Results

Machine	Batch Sequence	Total Penalty Cost
M1	4-1-6	1780
M2	7-5	
M3	2-3-8-9	

Table 3. Results of computational Experiment

No. of Machines	**Problem Size**	**GA**		**LEKIN®**	
		Total Cost	**Allocation**	**Total Cost**	**Allocation**
2 Machines Case	9 Job_0.7	29565	[5 4]	22665	[4 5]
	15 Job_0.7	51570	[9 6]	55766	[7 8]
	25 Job_0.3	38218	[13 12]	41878	[12 13]
3 Machines Case	9 Job_0.7	47063	[3 3 3]	36705	[3 3 3]
	15 Job_0.7	78660	[5 5 5]	78630	[5 5 5]
	25 Job_0.3	48312	[8 8 9]	59423	[9 8 9]

6. Conclusion

In this paper, a genetic algorithm approach is presented to solve the identical parallel machine scheduling problem. The objective is to minimize the total penalty cost, which includes earliness as well as tardiness penalty cost. The hybrid approach involving the genetic algorithm combined with backward-forward heuristic is used for obtaining the near optimal solution to the problem. In future, the performance of the genetic algorithm for different crossover, mutation, etc. can be investigated and also compared with other metaherustic approaches. The problem complexities like batch splitting, sequence dependent set ups, etc can be explored. Also, the algorithm can be used for developing the integrated approach between production scheduling and maintenance for parallel machine problem.

References

Balasubramanian H., Fowler J., Keha A., and Pfund M., "Scheduling interfering job sets on parallel machines", *European Journal of Operational Research*, vol. 199, no.1, pp. 55-67. 2009

Balin, S., "Non-identical parallel machine scheduling using genetic algorithm", *Expert Systems with Applications, vol. 38,* 6814–6821. 2011

Bilgesu, A. and Koc,E. A guide for genetic algorithm based on parallel machine scheduling and flexible job-shop scheduling, *Procedia - Social and Behavioral Sciences*, vol. 62 pp. 817 – 823. 2012

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

Chang P.C., Chen S.H., Lin K.L., "Two-phase sub population genetic algorithm for parallel machine-scheduling problem", *Expert Systems with Applications, Vol.* 29, pp. 705–712. 2005

Dang, Q., Diessen, T., Martagan, T. and Adan I., "A matheuristic for parallel machine scheduling with tool replacements", *European Journal of Operational Research,* Available online 6 October 2020

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

Eroglu, D.Y., Ozmutlu H.C., &Ozmutlu S., "Genetic algorithm with local search for the unrelated parallel machine scheduling problem with sequence-dependent setup times", *International Journal of Production Research,* vol. 52, No.19, pp. 5841-5856. 2014

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

Min L., and Cheng W., "A genetic algorithm for minimizing the makespan in the case of scheduling identical parallel machines*", Artificial Intelligence in Engineering*, vol. 13, pp. 399–403. 1999

Min L., and Cheng W., "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,* vo*l.* 22, pp.279–287. 2006

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

Sule, D.R., "Production Planning and Industrial Scheduling", CRC Press, Boca Raton. 2007

Vallada, E., & Ruiz, R., "A genetic algorithm for the unrelated parallel machine scheduling problem with sequence dependent setup times",*European Journal of Operational Research,* vol.211, pp. 612–622. 2011

Zhiming W. and Chunwei Z., "Genetic algorithm approach to job shop scheduling and its use in real-time cases", *International. Journal of Computer Integrated Manufacturing*, vol. 13, No. 5, pp. 422-429. 2000

Biography

Dr. Pravin P. Tambe is an Assistant Professor, at Department Operations Management & Decision Sciences Indian Institute of Management Tiruchirappalli, India. He holds a PhD in Industrial Engineering from Indian Institute of Technology (IIT) Delhi. He has more than 15 years of academic experience. He is Fellow of Indian Institution of Industrial Engineering. His research interest mainly focuses on reliability, maintenance planning, quality control and production scheduling. He has published research papers in international journals by Elsevier, Emerald, Springer, etc. and international conferences organized by POMS, IAENG, IIIE, etc. He is reviewer to many international journals like Computers & Industrial Engineering, Reliability Engineering & System Safety, European Journal of Operational Research, Journal of Manufacturing Technology Management and many international conferences.