doi:10.5937/jaes16-14953 Paper number: 16(2018)1, 495, 36 - 42

SCHEDULING IN PARALLEL MACHINES ENVIRONMENT USING GENETIC ALGORITHM

*Dr. Raghavendra B.V**
JSS Academy of Technical Education, Bangalore, India

Effective scheduling of the production process improves the operational efficiency. The objective of the scheduling is to meet the due date, maximum utilization of resources, reducing work in process inventory and improving manufacturing lead time etc,. Scheduling in the multi objective criteria is the crucial task but necessary to achieve the better operational efficiency in the competitive environment. When the complexity of the problems increases, it is the challenging process to obtain the optimum solution using mathematical or heuristic process alone. However application of genetic algorithm in scheduling process make easy to obtain the better and quick solution in the real environment. This paper uses the genetic algorithm for scheduling of jobs in the parallel machines production process. The algorithm is coded in MATLAB, and the objective functions are, minimum penalty cost, minimum machine idleness cost and combination of minimum penalty and machine idleness cost for comparison and discussion. The algorithm is tested for convergence, consistency and computational time.

Key words: Scheduling, Algorithms, Costs, Machine

INTRODUCTION

Scheduling strategy plays a major role in optimization of the production process. The objective of scheduling is to adopt the changes in the production process and plan the sequence of work to systematically distribute the workload on the machines. This helps the decision makers in deciding and analyzing the effectiveness of the process. The objective of the scheduling is to improve manufacturing lead time, reducing in-process inventory, penalty cost to meet the due date and balance the machine work load.Today, need is to accommodate the customer need and demands with immediate effects. An efficient and quick tool is required to schedule the work to be processed. Researchers have been working to obtain better solution in scheduling and continuing the research to develop the methodology and algorithms to optimize the objective functions. Researchers have taken the aid of various traditional mathematical, heuristic and meta-heuristic tools such as genetic algorithm (GA), ant-colon optimization (ACO), particle swarm optimization (PSO), artificial intelligence, Pareto diagram etc.

The basic and important need of scheduling is to define objective function to obtain the optimum schedule. Some of the most important objectives, which have been considered by the researchers in the context of batch production environment, are

- Maximizing system utilization
- Work load balancing on machine
- Minimizing manufacturing lead time
- Reducing the penalty cost or to meet due dates
- Minimizing work-in-process jobs
- Maximizing the rate of production
- Minimizing setup and tool changes times

Liu, Yang, Cheng, Xing, Lu, Zhao et al., (2012) [06] stated that flow job shop problem is an extension to traditional job-shop scheduling problem and usually has multiple optimization objectives. In multi-objective optimization, objectives are conflicting to each other. A trade-off among the objectives are required in multi-objective optimization, i.e., an improvement in the one objective solution is only achieved by making concessions in the solution of the objective. Hence obtaining the simultaneous optimum solution for the entire objective is not possible but there only exists a “compromise solution’ among the objective considered. Zheng (2010) [16] stated that a multi-objective optimization result is often not a single optimal solution, but a set of Pareto optimal solutions. The solution of mutli-objective optimization problems can be defined as if vector X is a solution of the stated equation, there exists on feasible vector X which would decrease some objective function without causing a simultaneous increases in at least one objective function. The solution following the above definition is also called a Pareto optimum or non-dominated solutions.

Xing, Chen & Yang, (2009) [13], Zhang, Shao, Li & Gao, (2009) [15], Xu, Ying & Wang, (2010) [14] have worked on single objective transformation, random weighting and optimization method based on Pareto. Li, Pan & Wang, (2010) [045, Ghasem & Mehdi, (2011) [02] stated that a Pareto optimal method can obtain a set of Pareto optimal solutions in an optimizing process, which is consistent with an actual scheduling problem.

Researchers have formulated the problem as 0-1 mixed integer programs, goal programming models, branch and bound approach, Petri net model and a reactive fast graph search algorithm etc. Many mathematical and heuristic approaches are applied to fulfill the multi-objective criteria. Swamkar and Tiwari (2004) [11] addressed

* *JSS Academy of Technical Education, Bangalore, India, Uttarahalli-Kengeri Road, Bengaluru – 560 060, bv_raaghu@yahoo.co.in*

machine-loading problem for minimizing system unbalance and maximizing the throughput.

Deb, K. and Miettinen K (2008) [01] used interactive and evolutionary approaches for multi-objective optimization, Tsung-Che Chiang, et al (2010) [12] proposed memetic algorithm for minimizing total weighted tardiness, simulated annealing algorithm for minimizing make-span used by Purushothaman, Damodaran and Mario C V´elez-Gallego (2012) [09]. A hybrid genetic algorithms was proposed by Joo, C. M. and Kim, B. S (2015) [03], Stefan Lausch and Lars M¨onch (2016) [10] used meta-heuristic approach and Jos´e Elias C Arroyo and Joseph Y-T Leung (2017) [04] worked on iterated greedy algorithm for scheduling on parallel batch machines.

M. Laumanns, L. Thiele, K. Deb and E. Zitzler, (2002) [07] stated that evolutionary algorithms suffer from the large size problem of the Pareto set. M. S. Osman, M. A. Abo-Sinna and A. A. Mousa (2006) [08] proposed methods that reduce the Pareto set to a manageable size. However, the goal is not only to prune a given set, but rather to generate a representative subset, which maintains the characteristics of the generated set. Also evolutionary algorithms such as, genetic algorithms (GAs) can be used as a global optimization tool for continuous and discrete functions problems.

Above approaches have limitation in its application as complexity increases due to increase in number of jobs, setups and machines. But with the technological development in computing speed and evolutionary algorithms such as Genetic Algorithm (GA), Ant-colony algorithm, Particle Swarn Optimizatin (PSO), etc. would help to obtain the better solution for complex problems.Scheduling problem using genetic algorithm was presented by researchers have used the multi-objective function and the solution obtained from different approaches were compared and analyzed the performance of minimizing total penalty cost and minimizing total machine idleness.

In this paper a heuristic method is used to minimize the wok-in-process inventory and to handle complexity of the process, genetic algorithm is used to minimize the combined objective of penalty and machine idleness cost.

PROBLEM FORMULATION AND PROPOSED METHODOLOGY

The study have been done for identical parallel machine scheduling, with the objective of minimizing work-in-process material, penalty (due date) and machine idleness cost in a batch production environment. A heuristic procedure is developed (Figure 1) for minimization of the work-in-process job and machine idleness followed by genetic algorithm to determine the best sequence of the job to meet the objective function (Fitness function).

Technological constraint of the machine to process all the operation on job decides the number of times jobs should be loaded for machining on the machine. Each time loading of the jobs on machine, needs set up (Si) of machine for fixture and tools. Feasible operations are grouped to process on machine in each set-up (loading job on the machine) considering the technological features available on the machine. The number of such feasible groups of operations will be processed in different set-up sequentially considering the precedence setup is the constraint. Hence the operation grouped for successive set-up (Si+1), should be done only after completing the operations grouped in the precedence set-up (Si).

The jobs processing time in each set-up is determined and processed on any of the parallel machines. An efficient job sequence and machine loading need to be determined to meet the effective scheduling.

The algorithm is suitably designed for parallel machines having “m” number of machines, “Si” number of setup, “J” number of part types and batch quantity is “bj”. The algorithm is experimented for six parallel machines, three number of setup, ten part types and batch quantity of ‘10’ for each part types/Jobs.

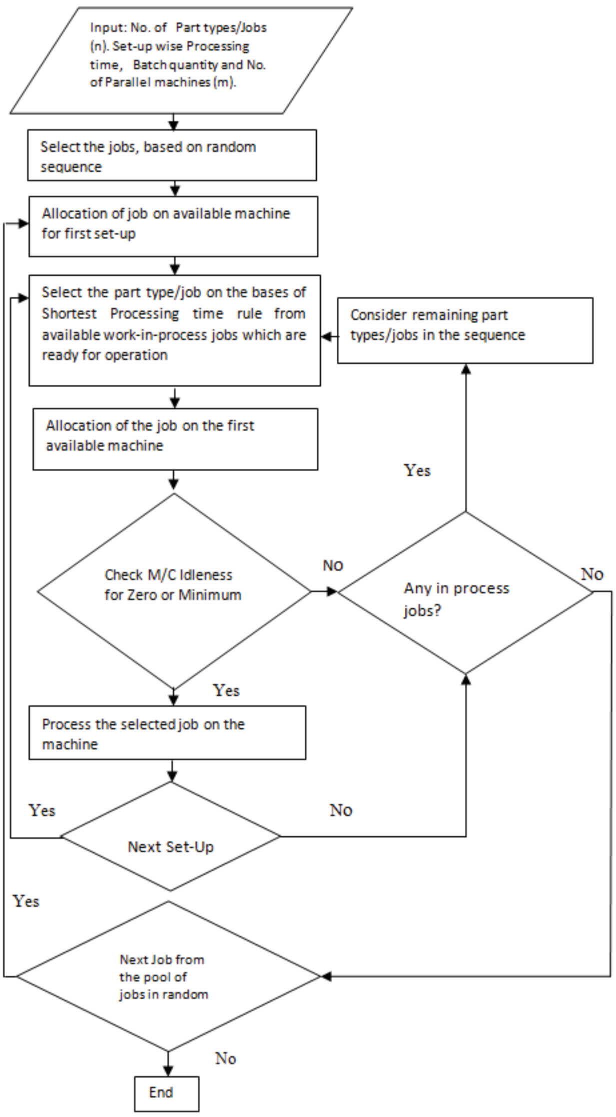

Figure1: Flow Chart for Heuristic Method

ASSUMPTION

- Parallel processing of selected part types/jobs.
- Enough tool slots are available on machine to carry out the machining operation.
- Beginning of the process all the jobs and machines are available.
- Relaxed transportation time, availability of the resources such as material handling system, pallets, fixtures and set-up time of machine.
- Part types/jobs are processing in a batch of quantity "bj".
- Machining of the job on set-up (Si) should be completed before starting of next set-up (Si+1) on the same job.
- Number of hours available per day on each machine is 480 minutes. (One shift 8 hours per shift)

PROPOSED JOB SEQUENCE RULE

- Random sequence of job (generated by random generator in genetic algorithm) is used for loading the jobs on machine in I set-up (Si, i=1).
- For remaining set-up (Si, i=2,3…), job sequence is based on shortest processing time (sum of all the operations time in the set-up) of immediate competing work-in-process jobs (WIP).

PROCEDURE (REFER FIGURE 1)

- Beginning of the scheduling process all the machines are available for machining and job to be loaded based on random sequence of job (generated by random generator in genetic algorithm).
- After completion of first set-up operation of selected jobs, load the machine by work in process jobs based on shortest processing time rule (sum of all the operations time in the set-up). If machine is idle for selected job, then select the next (competing) work in process job.
- If all the work in process jobs is completed, assign the remaining jobs on the machine from randomly generated job sequence (generated by genetic algorithm in the first step).
- Complete the machining of all the jobs by repeating above steps.

MODEL DESCRIPTION

j= Part type/Job index: j= 1,2,...,n (Number of Part types);

i= Machine index; i=1,2,...,m (Number of machines);

tjs= Processing time of operations on "Si" setup for job j;

bj= Batch size of Part type j;

dbj= Due date (deadline) of batch b of part type j;

Cbj =Completion time of batch b of part type j;

pbj= Penalty cost of batch b of part type j. (Rs/unit/day)

dm=Idleness of machine m (hrs);

lbj= Lateness of batch b of part type j=Cbj-dbj; j=1,2,...,n. { if (Cbj>dbj) other wise 0};

dmc= Idleness cost of machine m (Rs/hrs)

$$\text{Ctpc} = \text{Total Penalty cost} = \sum_{j=1}^{n} \text{pbj X lbj}$$

Ctmc=Total machine idleness cost= $\sum_{i=1}^{m} dm\ X\ dmc$

Total Cost= Ctpc+Ctmc

Objective function: Minimize Ctpc, Ctmc and Ctpc+Ctmc

Designing and representation (encoding) is important in the development of genetic algorithm. In this work, sequence-oriented representation scheme is used. Initially populations are randomly generated by the genetic algorithm. The part types/jobs are the gens in the chromosomes.

In this paper objective function is used to evaluate each chromosome (Job Sequence). The chromosome which has the least cost is the best solution to meet the stated objective. The best generated solution of the job sequence is used to assign the work on the machine.

Sum of processing time of all the part types/jobs processing on a machine is the total workload on the respective machine. Machine idleness is the sum of waiting time that the machine to process/machining of the jobs till last jobs leaves in the system.

In general, Genetic algorithm performance is based on cross over and mutation operator. Partially mapped cross over is used considering workload balance is the combinatorial optimization problem. The mutation operator, reciprocal exchange is used. A new string is generate by randomly selects two positions in the string and swaps the part types in these two positions.

Termination is the criterion by which the genetic algorithm decides whether to continue searching or stop the search. In this work maximum number of generation (A termination method that stops the evolution when the user specified maximum number of evolution has been run.) is applied for termination of the evolution. MATLAB is used to code the algorithm and experimented based on objective functions such as, Minimum Penalty cost, Minimum Machine Idleness cost and Combination of both.

Journal of Applied Engineering Science Vol. 16, No. 1, 2018
ISSN 1451-4117

RESULTS AND DISCUSSION

The experiment has been conducted with setting the genetic algorithm parameter as, population size 20, selection of best parents 12, maximum number of generation for which evolution stops is 30 and cross-over fraction is 0.9. Experiment of 110 trials was conducted to study the performance of the designed algorithm and the computational time. The solution is converged at 11 to 15 evolutions in the experiment. However the evolution was continued till to 30 and found that the solution was constant after 15 evolutions. The job sequence for objective function of minimum penalty cost, minimum machine idleness cost and combination of both, is shown and also the result were compared (Figure 5 and Table 2). The convergence of the result is presented for objective functions, Minimum machine idleness cost (Figure 2), penalty cost (Figure 3) and combination of both (Figure 4).

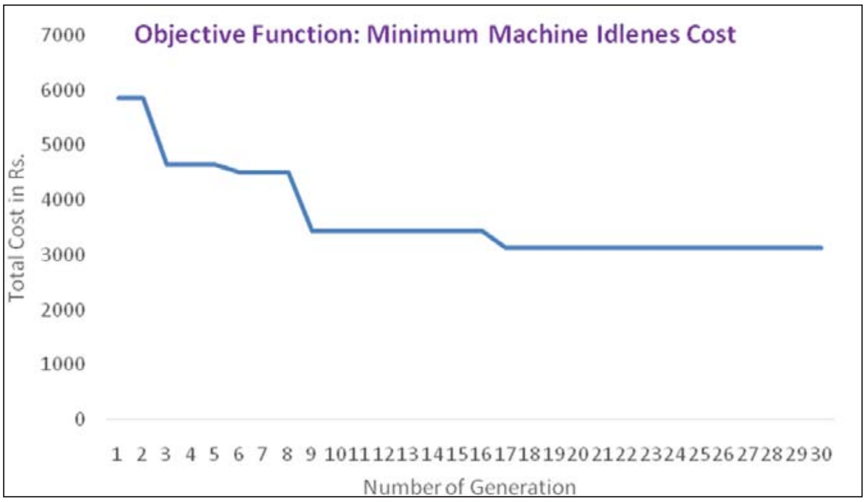

Figure 2: Objective function is minimum machine idleness cost

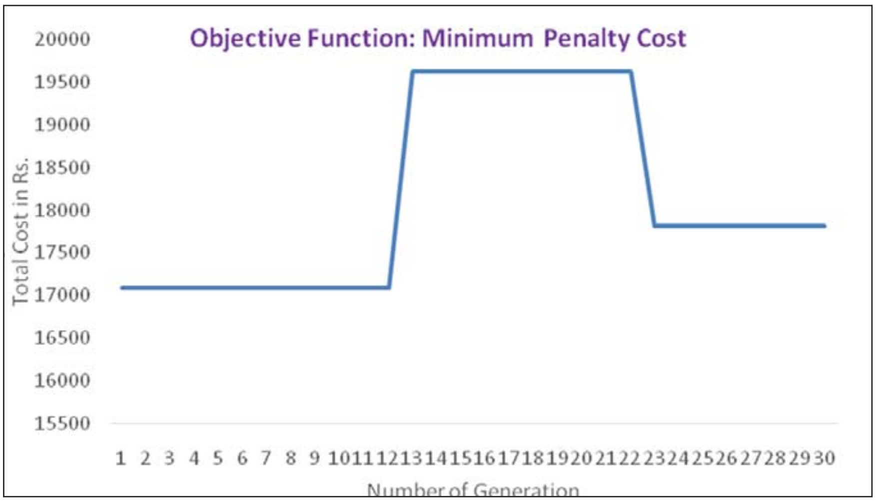

Figure 3: Objective function is minimum penalty cost

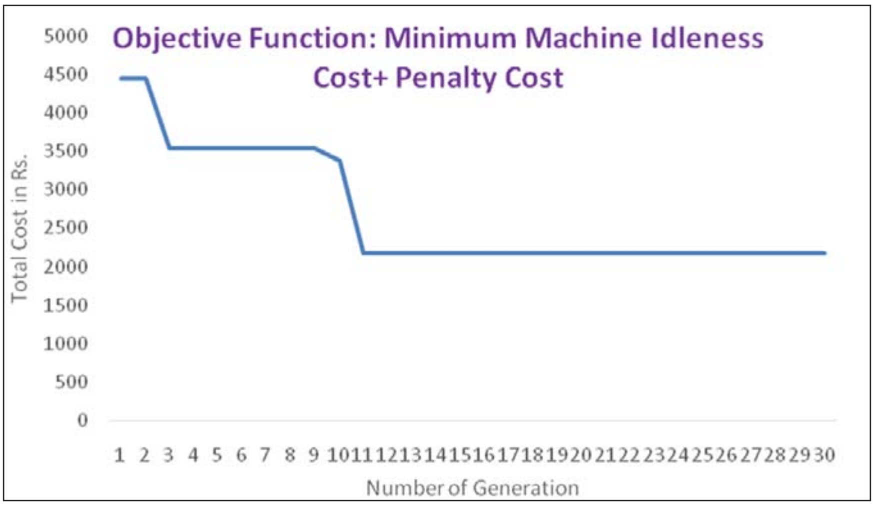

Figure 4: Objective function is minimum penalty and machine idleness cost

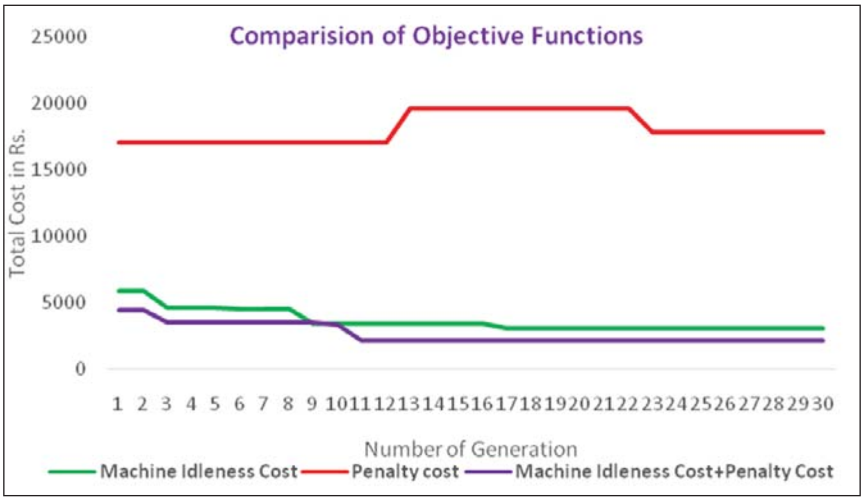

Figure 5: Comparison of Objective functions

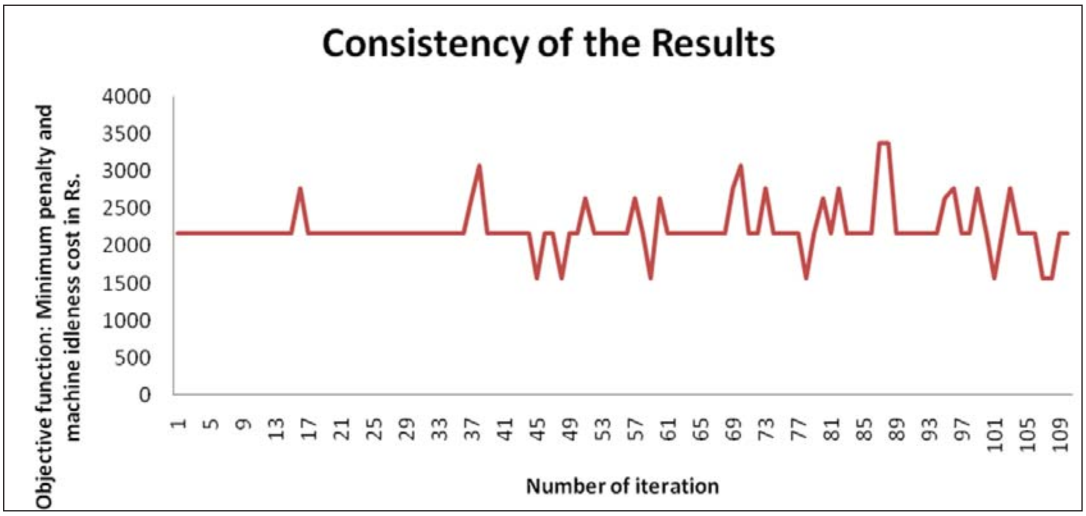

Figure 6: Consistency of the algorithm

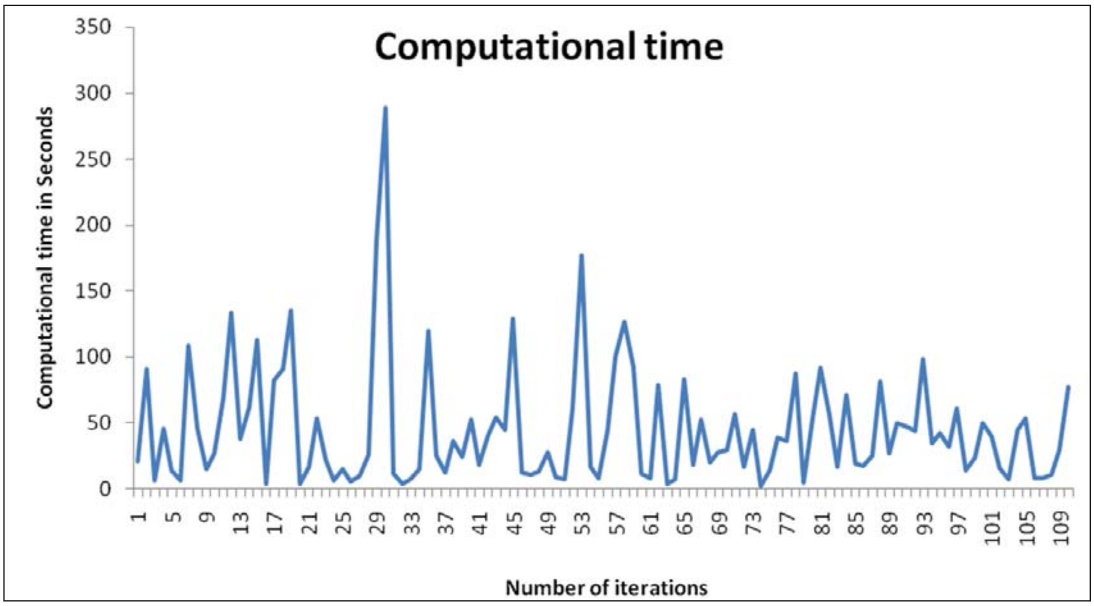

Figure 7 : Computational time of the algorithm

Table 1: Data for Experimentation

Part Type (j)	Set-up(Si)			Batch Quantity (bj)	Due day (dbj)	Penalty cost Rs/Day/batch (pbj)
	I	II	III			
	Processing/Operation time in min.(tjs)					
1	62	44	2	10	2	10
2	53	46	-	10	2	12
3	38	38	20	10	2	12
4	34	31	10	10	2	13
5	32	19	10	10	2	9
6	33	31	9	10	1	11
7	31	30	16	10	1	11
8	75	-	-	10	1	14
9	6.78	-	-	10	1	8
10	17.34	5.15	15	10	1	10

Machine Hour rate: Rs. 600 per machine per hour

Table 2: Genetic Algorithm Generated Job Sequence

Objective Function	GA Generated Sequence of Part type/Job										Objective function (Rs).	Total Cost (Rs).
Minimum Penalty Cost.	6	2	3	9	10	8	7	4	5	1	44	17817
Minimum Machine Idleness Cost.	3	8	10	1	9	6	2	5	4	7	3067	3128
Combination of Minimum Penalty and Machine idleness Cost.	4	3	7	1	8	2	9	10	6	5	2173	2173

CONCLUSION

In this study heuristic procedure for minimizing the work in process inventory followed by genetic algorithm based methodology is presented on parallel machine batch production environment. The result obtained in the study using the genetic algorithm is compared for the objective functions, minimum penalty cost (Total cost Rs. 17817), minimum idleness cost (Total cost Rs. 3128) and combination of both and found that the genetic algorithm based combinatorial objective gives better result (Total cost Rs. 2173) than the other two objective functions. The proposed genetic algorithm is tested on Intel ® Core(TM) I 7-6700 CPU @3.4GHZ with 110 iteration for its performance and computational time. The results are consistent and the average computational time to obtain the solution is 45 seconds. The algorithm helps to obtain the convergent and consistent results with the reasonable time for scheduling of the jobs on the machines.

REFERENCES

1. Deb, K. and Miettinen, K. (2008). Multi-objective optimization: Interactive and evolutionary approaches, Springer Science & Business Media, Volume 5252.
2. Ghasem, M., & Mehdi, M. (2011). A Pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search. International Journal of Production Economics, 129(1), 14-22.
3. Joo, C. M. and Kim, B. S., (2015). Hybrid genetic algorithms with dispatching rules for unrelated parallel machine scheduling with setup time and production availability, Computers & Industrial Engineering, Volume. 85, 102-109.
4. Jos´e Elias C Arroyo and Joseph Y-T Leung, (2017). An effective iterated greedy algorithm for scheduling unrelated parallel batch machines with non-identical capacities and unequal ready times. Computers & Industrial Engineering.
5. Li, J.Q., Pan, Q.K., & Wang, Y.T. (2010). Hybrid Pareto-Based Tabu Search Algorithm for Solving the Multi-Objective Flexible Job Shop Scheduling Problem. Computer Integrated Manufacturing Systems, 16(7), 1419-1426.
6. Liu, Yang, Cheng, Xing, Lu, Zhao et al., (2012). Stated that flow job shop problem is an extension to traditional job-shop scheduling problem and usually has multiple optimization objectives.
7. M. Laumanns, L. Thiele, K. Deb and E. Zitzler, (2002). Archiving with Guaranteed Convergence and Diversity in Multi-Objective Optimization, Proceedings of the Genetic and Evolutionary Computation Conference, New Yourk, 9-13, pp. 439-447.
8. M. S. Osman, M. A. Abo-Sinna and A. A. Mousa. IT-CEMOP, (2006): An Iterative Co-Evolutionary Algorithm for Multi-objective Optimization Problem with Nonlinear Constraints. Journal of Applied Mathematics & Computation, Volume. 183, No. 1, pp. 373-389.
9. Purushothaman Damodaran and Mario C V´elez-Gallego, (2012). A simulated annealing algorithm to minimize make-span of parallel batch processing machines with unequal job ready times. Expert Systems with Applications, 39(1):1451–1458.
10. Stefan Lausch and Lars M¨onch, (2016). Meta-heuristic approaches for scheduling jobs on parallel batch processing machines. In Heuristics, Meta-heuristics and Approximate Methods in Planning and Scheduling, 187–207. Springer.
11. Swamkar R, Tiwari MK, (2004). Modeling machine loading problem of FMSs and its solution methodology using a hybrid tabu search and simulated annealing-based heuristic approach. Robot Computer Integrated Manufacturing 20:199–209.
12. Tsung-Che Chiang, Hsueh-Chien Cheng, and Li-Chen Fu, (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.
13. Xing, L.N., Chen, Y.W., & Yang, K.W. (2009). Multi-objective flexible job shop schedule: design and evaluation by simulation modeling. Applied Soft Computing, 9(1), 362-376.
14. Xu, X.L., Ying, S.Y., Wang, W.L. (2010). Fuzzy Flexible Job-Shop Scheduling Method Based on Multi-Agent Immune Algorithm. Control and Decision, 25(2), 171-184.
15. Zhang, G.H., Shao, X.Y., Li, P.G., Gao, L. (2009). An effective hybrid particles warm optimization algorithm for multi-objective flexible job shop scheduling problem. Computers & Industrial Engineering, 56(2), 1309-1318.
16. Zheng, J.H. (2010). Multi-objective Evolutionary Algorithm and Application. Beijing: Science press.

Paper submitted: 07.09.2017.
Paper accepted: 27.11.2017.