International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 9 (2015) pp. 22843-22851

http://www.ripublication.com

Application of Metaheuristic Approach In Parallel Flow Line Scheduling

Rajendran.S [#1], Balasubramanian K [*2], Rajeswari N [#3]
[1]Research Scholar, Dr MGR Educational Research Institute University, Chennai, India
[1] rajendiran.s@gmail.com
[2] Veltech Engineering College, Chennai, India
[3] Veltech Multitech Dr.Rangarajan Dr.Sakunthala Engineering College, Chennai, India

Abstract

A metaheuristics approach is chosen in cases where the exact methods are not enough to provide a solution. It is an iterative process to efficiently produce solutions which are closer to the global optimum.. This paper addresses the application of generic algorithm for a parallel machine flow line scheduling problem using the algorithm proposed for minimizing the makespan. Sequencing of operations to attain a feasible minimum makespan is one of the important requirements to achieve effective production from process planning. Since the problem is NP-hard, it is chosen to adopt genetic algorithm as it is one of the proven methods to search for a feasible optimal solution to the objective function. This paper addresses the methodology to obtain a near optimal sequence of jobs for an allocation of constrained resources with the objective of minimization of overall completion time or makespan. The methodology is based on creating a group of random solutions and uses the genetic operators of cross over and mutation to improve the solutions till an acceptable fitness level is reached. As the implementation of the local search is time consuming, a tool based on visual basic was developed to do the search faster. The tool had the options to search exhaustively or using the genetic algorithm methods. The computational experiments deployed indicate that it is feasible with the proposed methodology and procedures to arrive at better solutions faster than the conventional methods.

Keyword: Scheduling -Parallel machine – Flow line scheduling– Genetic algorithm.

Introduction

Production scheduling as defined in standard text books is the allocation of available production resources over time to perform a collection of tasks [13].

It is known that scheduling is a complex activity and *NP- hard* for most of the problems which could be even static in nature. The concept for the chosen problem is motivated by the corollary of Just in time concepts by which we seek to achieve minimum processing time for the jobs.

Also in the batch/mass production environment the situation exist in terms of machine setups which are procured and installed over different periods of time. In such cases, it is obvious that the new lines are effective in terms of achieving minimum time for processing but in the meantime, it is also profitable that the relatively older lines which are in good working condition are effectively utilized in the interest of increasing return on investments. The typical challenge would be to decide which lines are to be used and which ones may not be effective to get profitability. The flow line scheduling is used as an option to enable decision makers in knowing a near optimal solution.

The flow line scheduling is a combination of parallel machine and flow shop scheduling. Flow shop is characterized by a different set of machines arranged in a line sequence and parallel machine setup is characterized by the same set up of machines arranged in parallel with more than one processing line for the jobs to be processed. Hence the parallel flow line is characterized by a number of parallel lines, each consisting of a number of machines. The corresponding machines in any two given parallel lines are considered as identical and have equal number of machines on the lines. Such a set of flow shop machines arranged in parallel setup is defined as a parallel flow line setup.

The essential consideration of conditions in a parallel flow line is that the jobs are not moved from one line to other during phase of processing. Even though the processing time of a job is considered different for each machine, jobs are assigned to one particular line and the sequences of machining are completed in that line only.

The decision to utilize a specific algorithm for resolving this situation is not an easy one as there are multiple options to consider. Based on the literature surveys, in terms of the characteristic feature of achieving a near global optimization in a shorter time, higher speed and amenability to utilize the algorithms in a coded form using digital computers to study the time taken to complete complex problems, we have chosen genetic algorithm as a choice to study the problem chosen. The objective of minimization of makespan is presented in this paper.

The makespan (Tmax), is defined as max(T1, . Tn), is the time difference between the start and finish of a set of tasks or sequence of jobs. It implies the completion time of the last job to be completed by the chosen line. A minimum makespan indicates an effective utilization of the machine setup and hence the objective is always to minimize the makespan. This makespan considered in this paper includes the processing times and other setup and change over times.

Structure of The Problem

This paper has the following assumptions:

1. All jobs are available at time t =0.
2. The jobs are ready to be submitted to the manufacturing sequence in any random fashion.
3. The required times including loading the job, setting up the machine, processing time, transfer times, any other idling time and unloading times are previously known for each job and included in the processing time.
4. There is no correlation between operation time and setup times on different machines.
5. Job once chosen in a line will not be allowed to transfer between lines.
6. It is assumed that the breakdown of the machine is not existent..

The notations used in the problem are as follows.:

j refers the job identifier { j = 1,2,..,J}

l refers the line identified {l=1,2,…,L},

m refers the machine identifier {m =1,2,..M}

T_{Qjl} – refers the flow time of job in the line l as processed by the machines

q – batch quantity for job j

FL | $p_{jl} = p_l$ | $\sum T_l$ denotes a flow line environment with m machines in series; the processing times of job j on all m machines are identical and equal to p_l (proportionate). The objective is to finalize the order in which the n jobs go through the line so that the sum of the weighted / completion times is minimized [12].

Using the chosen objective function minimizing the makespan is ensured so as to enable us to decide which job needs to be assigned to which line in order to reduce the overall flow time.

Literature Survey

The literature survey referred from authors of similar related research work and from basic text books on scheduling. Michael L. Pinedo [12] has discussed the fundamentals of scheduling models. Baker KR et. al. [13] has discussed the basic principles of scheduling with practical aspects of coverage.

Campbell et.al. [1] Showed a simple algorithm to solve flow shop type problem which can be extended to large problems and even without the use of computer. Gupta et.al. [2] Have discussed on a problem that gives the smallest makespan among the set of all schedules with optimal total flow time using an improvement algorithm. Al-Salem et.al. [2] Had proposed a flow line scheduling algorithm for minimizing the makespan on a 2 stage parallel flow shop and compared with SKV algorithm. Bilge et.al [4] discussed for parallel machine problem, a generalized total tardiness problem using tabu search approach and compared with alternative approaches. Haq A et.al. [5] Have presented a scheduling problem for multiple parallel flow line machines with sequence dependent setup times with the objective of minimizing makespan using genetic algorithm and compared the randomly generated problems with tabu search method. Rajeswari N et.al. [6] Presented the problem of scheduling jobs on parallel flow line with no common due dates using genetic algorithm and tabu search methods.

Berrichi. A et. al [7] has proposed an evolutionary genetic algorithm for a flow line scheduling problem to find an approximation of the Pareto optimal front. Geiger et.al. [8] Presented a paper on the method to predict the relative objective weighting scheme necessary to cause arbitrary members of a Pareto solution set to become optimal. Liu et. al. [9] presented a kind of genetic algorithm based on machine code for minimizing the makespan in identical machine scheduling problem. Dipak et. al. [10] has proposed a heuristic for the no-wait flow shop scheduling problem. Luangpaiboon, et. al. [11] has compared the various methods of flow shop scheduling problems with constraints.

Genetic Algorithm

Genetic algorithms are problem solving methods based on search principles of natural selection and genetics. This was formulated in 70's by John Holland. This method is proven to be useful for resolving combinatorial optimization problems to find faster near optimal solutions in a global search space chosen.

When we solve any problem, we are always on the lookout for a solution which is the best among others. The space of all such feasible solutions forms the search space. One point among the search space forms the best solution. On NP problems (i.e Non deterministic polynomials) it is difficult to check the solution, and in such cases genetic algorithm is one of the accepted form of a solution as on today.

This works on the principle that a process with a set of attributes which follows the natural evolutionary sequence will eventually lead to a good solution by survival of fittest attributes. A genetic algorithm, when applied to scheduling, represents the sequences as members of a population. Each individual member is characterized by a fitness value. The fitness value of an individual is indicated by the associated part of the objective function value. The strings which are candidate solutions to the search problem are referred as 'chromosomes', the alphabets or strings used are referred to as 'genes' and the values of genes are called termed as 'alleles'. The procedure works iteratively, and each iteration is referred to as a generation. The population of one generation consists of survivors from the previous generation plus the new schedules, i.e., the offspring of the previous generation. The size of the population normally remains constant from one generation to the next offspring. The off springs are generated through a method of reproduction by crossover and mutation of individuals that were part of the parents.

Crossover is deployed by exchange of variable parameters between locations whereas mutation is achieved by introducing one or more variables by a random change. The crossover was deployed in single or two points which is known as single point cross over or multi point crossover as the case may be.

The population size, which is usually a user-input parameter, is one of the important factors affecting the result performance of genetic algorithms. A smaller population size might lead to substandard solutions. Whereas a large population size might lead to excessive computational time.

The sequences of operations in a GA are

1)Initialization, 2)Selection,3) Recombination 4) Mutation & 5) Replacement and repeating these steps as needed till the conditions of the objective function are met.

The generic structure of genetic algorithm can be considered as follows

```
Procedure GA ()
Begin {
       t -> 0;
initialize pop(t);
evaluate pop(t);
while (Not termination condition) do
begin {
                  t=t+1;
       }
 End;
       }End;
```

Results and Discussion

Multiple sample problems were taken for testing. One such input is reproduced here as a sample and the results from some of the iterations are also shown which is a case of 4 machines and 4 jobs in 4 parallel flow lines. Variable quantity of job were considered as input to arrive at the minimum makespan

		Machine process time			
J1	1	3	4	5	7
	2	2	4	4	5
	3	1	3	3	4
	4	1	2	3	4
J2	1	7	2	1	2
	2	6	6	1	1
	3	5	5	6	2
	4	4	4	5	7
J3	1	3	4	5	6
	2	1	2	3	4
	3	1	1	3	4
	4	7	1	2	2
J4	1	5	6	1	2
	2	5	6	6	1
	3	4	4	5	6
	4	2	4	5	6

Figure 1: Input Sample

The resultant analysis of makespan over the iterations is as shown in the fig.2. The horizontal X-axis shows the number of generations, whereas the vertical Y axis represents the makespan obtained through the generations.

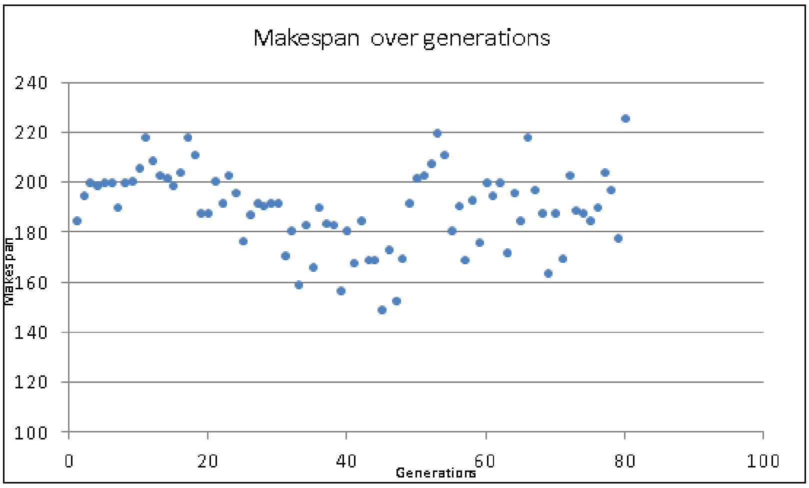

Figure 2: Result – Makespan (Generations vs, Makespan)

Based on the analysis in which the sequence sample size 'N' is considered as fixed value, the result obtained for every sequence is as recorded and it was observed to be the lowest makespan for the sequence chosen. Further generations did not improve the value beyond this minimum which may be inferred as a near optimal minimum makespan for the problem chosen.

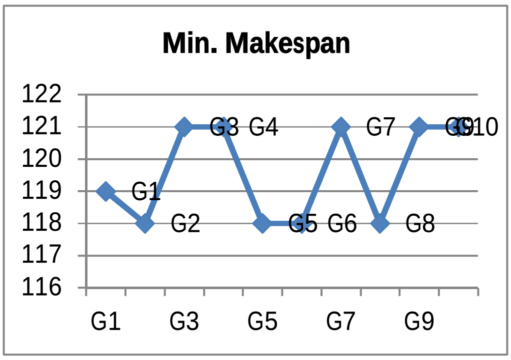

Figure 3: Min Makespan Over Generations

Based on the minimum makespan over the generations is shown in the above fig.3 which shows that the minimum makespan value achieved after repeated generations also does not fall below the first minima reached which turns out to be the global minimum. Another observation is that for the min makespan it is not essential that the flow lines has to share the loading of jobs equally which can be observed from fig.4

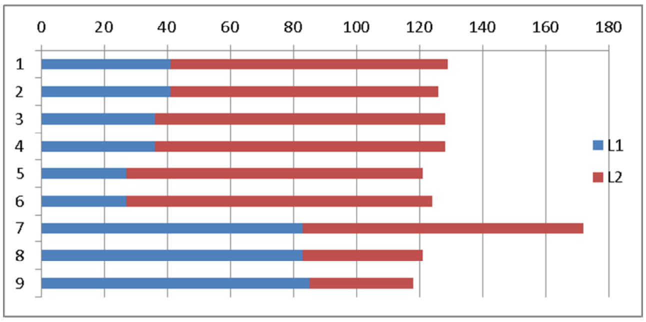

Figure 4: Distribution of jobs over lines

The total CPU time taken for arriving at this computation has been tabulated below in Fig.4. The computer configuration used was of Intel i5 core 2.3 GHz processor with 4GB RAM.

Whereas when multiple trials are run, when the sequence is considered of a higher size, the time taken to arrive at the solution by exhaustive search increased phenomenally as given in the following table.

Table 1: Results for discussion

Sequence	**MS**	**Time using exhaustive search**	**Solution time based on GA method**
4-2-3	118	18 sec	10 min
4-2-5	139	26.4 min	22 min
4-4-5	149	27.4 min	35 min
4-2-7	458	64 min	1h 14 min
4-2-10	867	~6 days	~ 2 days

The user friendly tool and the utilization of digital computer facilitated a near exhaustive search relatively easy. Caution is exercised that though it appears that the exhaustive search may also be a feasible option for very small sized problems, the times taken to run larger problems to arrive at the solutions outweigh the benefits and in case of problem size increase, it is becoming nearly impossible to arrive at a solution. In order to reduce the lead time required to arrive at near optimal solutions, using the proposed GA methodology was inferred to be helpful.

Risk Assessment

In this problem set, the research has the values of make span pre-determined by running the total execution process to validate the results. In cases when a new problem is formulated, we need to rely solely on the GA methodology alone, reaching the near optimal solution could consume a higher time and more complexity than arrived in this demonstrated method.

Conclusion

This paper has provided the method of obtaining a solution of near optimum makespan value attained for the parallel flow line problem using genetic algorithm technique. Based on the fitness criteria evaluation, the minimum is inferred as the global minimum. The test problem was extended to find other minima during the search to ensure that we do not end up getting trapped in local minima. The computer code generated using visual basic helped to have a user friendly interface and a database to store the results for future reference. The same has been tested with multiple standard problems to ensure that the results are reliable for larger problems.

References

[1] Herbert G. Campbell, Richard A. Dudek and Milton l. Smith, A heuristic algorithm for the N-job, M-machine sequencing problem, Management Science, Vol.16, No. 10, 1970. pp. 630 – 637.

[2] Jatinder N D Gupta, Johnny C. Ho and Alex J.Ruiz-Torres, Makespan minimization on identical parallel machines subject to minimum total flow-time, Journal of the Chinese institute of Industrial engineers, vol. 21, No.3, pp. 220-229,2004.

[3] Ameer Al-Salem, A Heuristic to minimize makespan in proportional parallel flow shops, International journal of computing & information sciences, Vol. 2, No.2,August 2004, pp. 98-107.

[4] Ümit Bilge*, Furkan Kiraç, Müjde Kurtulan, Pelin Pekgün , a Tabu search algorithm for parallel machine total tardiness problem Computers and Operations research 2004, Volume 31, Issue 3,2004, pp. 397–414.

[5] Noorul Haq.A, Balasubramanian.K, Search heuristics for multiple parallel flow line scheduling, International Journal of Applied Management and Technology, Vol. 3, No.1, 2005, pp. 109 – 119.

[6] Rajeswari N and Shahabudeen P, Meta heuristics approach to schedule bi-criteria parallel flow line problems, The International Journal of Applied Management and Technology, Vol. 5, Num. 2, 2007. pp. 134-145.

[7] Berrichi.A , Amodeo.L, F.Yalaoui, E.Chatelet, M.Mezghiche (2009), Bi-objective optimization algorithms for joint production and maintenance scheduling: application to the parallel machine problem. Journal of Intell. Manufacturing, Vol. II, (2009) 20:pp.389–400.

[8] Martin Joseph Geiger, Pareto Analysis in Multi objective Optimization Using the colinearity Theorem and Scaling Method, pp. 105-1, MIC 2009: The VIII Metaheuristics International Conference, Hamburg, Germany.

[9] Liu Min*, Wu Cheng, A genetic algorithm for minimizing the makespan in the case of scheduling identical parallel machines, Artificial Intelligence in Engineering, vol. 13 (1999, pp.399–403.

[10] Dipak Laha and Sagar U. Sapkal : An Efficient Heuristic Algorithm for m-Machine No-Wait Flow Shops, vol. I, 2011, pp 767-1580. Proceedings of

the International multi conference of engineers and computer scientists, Hong Kong.

[11] Pongchanun Luangpaiboon, Atipon Boonvirojrit , Comparison of Heuristic Methods for the N-Job and M-Machine FlowShop with Resource Constraints, pp. 246-256. (Conference on operations research - online version, Faculty of engineering, Thammasat University, Bangkok).

[12] Michael. L. Pinedo, Scheduling, Theory, algorithms and systems Ed.3, 2008 Springer Science Business Media, LLC

[13] Baker KR and Dan Trietsch (1974) Introduction to Sequencing and Scheduling. John Wiley & Sons, New York