Journal of Manufacturing Engineering, June 2015, Vol. 10, Issue. 2, pp 069-073

A GENETIC ALGORITHM FOR JOB SHOP SCHEDULING PROBLEM IN AGILE MANUFACTURING SYSTEM

*Manoj Kumar

International Engineering Services, H.No.:87A, RZI-Block, West Sagarpur, New Delhi – 110046, India

ABSTRACT

The new challenges of Agile Manufacturing system led to study of computational cooperative problem solving models. The goal is to develop appropriate computational approaches to support adaptive, cost-effective responsiveness. In particular, the challenging problem of job shop scheduling, this has been one of the primary foci of production scheduling research. In this paper, we propose genetic algorithm to overcome the impact of agile environment such as changing customers' preferences, machine breakdowns, deadlocks, etc. by inserting the slack that can absorb these disruptions without affecting the other scheduled activities. The proposed algorithm also focuses on the impact of agility in the job shop environment in such highly complex scenarios. The algorithm inherits the delicacies of Genetic Algorithm (GA) converges towards optimality in less computational time. The proposed model encompasses the objectives of minimizing the delay time and flow time using the genetic algorithm.

Keywords: *Agile manufacturing system; Job shop scheduling problem, Genetic Algorithm (GA).*

1. Introduction

Present scenario being a highly competitive one, urges the manufacturers to strive hard for achieving the timely and cost effective production that can facilitate them to respond to the exponentially increasing demands of the customers. Agile manufacturing refers to the ability of a company to modify its production according to the sudden changes in the customers' demands. The effectiveness of the production schedule in the dynamic environment depends on its ability to cope up the various stochastic disturbances such as changes in machine schedule, breakdown of machines, deadlocks etc. in the system. It is very difficult task to predict the actual states of agile manufacturing where many uncertainties related to change in customers' preferences, arrival of parts, machine breakdowns, tool breakages, deadlocks, etc. exist and this is the primary reason, why the implementation of the on-line scheduling is practically infeasible. In this regard, authors have primarily focused to develop such extrapolative schedules, which efficiently take care of the disruptions on the shop floor and retain the high performance value of the system. Main motive behind these schedules is to assign the shop resources to the different jobs effectively for optimizing the performance measures of agile manufacturing. The uncertainties in agile manufacturing environments have been broadly classified in the three categories such as, sudden change in customers' demand, complete unknowns, suspicious about the future, and known uncertainties. The Genetic Algorithm (GA) based solution methodology is employed to obtain optimal or near optimal performance measure for the system i.e. minimum make span, average flow time and delay time for the schedules in an agile manufacturing. Intensive computational experiments have been performed for different scenarios of the problem in agile manufacturing.

The paper attempted to study of the impact of agility measures under dynamic and changing conditions in the job shop scheduling problem. In order to satisfy the need of agile manufacturing systems, job shop scheduling problem focusing with stochastic process time has become one of the newest issues that have been increasing recently. Section 2 of the paper describes literature review. A complete modeling of the problem that takes into account the uncertainties is detailed in section 3. Genetic algorithm and their application over the underlying problem are discussed in section 4. Computational experiments are presented in the section 5. The paper is concluded in section 6.

2. Literature review

In general, there are types of problems solved in the literature that are related to the scheduling problems discussed here. One type of related problems

**Corresponding Author - E- mail: kumarm1968@rediffmail.com*

are those of scheduling products represented by simple and complex digraphs in a two-stage manufacturing system first solved as an aggregate scheduling problem in Kusiak (1989). Here, the aggregate scheduling problem is modeled as the two machine flow shop scheduling problems. However, the aggregate scheduling problems solved in Kusiak (1989) assume that only one processing unit is available at both machining and assembly stages and this assumption does not reflect the real situation in implementing product differentiation strategies in an agile manufacturing environment. Another type of related scheduling problems solved in the literature are the flow shop problems with parallel machines (FSPM). FSPM is also a basic model for flexible flow line scheduling problems solved in the literature. Hunsucker and Shah (1994) reviewed industrial applications of scheduling in chemical engineering, computer systems, tele networks, etc. Brah and Hunsucker (1991) developed a branch-and-bound algorithm to solve the make span FMPM scheduling problem. Mathematical models of FSPM have been discussed in Brah *et al.* (1991). Some special cases of FSPM have also been studied in the literature. Gupta (1988) developed a heuristic algorithm for a two-stage problem with one machine at the second stage. Sriskandarajah and Sethi (1989) developed heuristics for a two-stage case and established the worst case bounds. Two heuristic algorithms generating high-quality solutions for a two-stage case with one machine at stage one and several machines at stage two have been developed by Gupta and Tunc (1991). Chen (1995) developed heuristics to solve the special cases for systems that have only two centres or systems where only one of the centres has parallel machines. Extensions of FSPM that incorporate buffers and transporters between centres have been studied by Wittrock (1988) and Sawik (1993). Flow shop scheduling problems and parallel machine scheduling problems represent another class of related problems. The two-machine flow shop-scheduling problem can be solved by Johnson's algorithm (1954). However, in general, if the number of machines in a flow shop is more than two, then the scheduling problem is known to be NP-complete (e.g. Gonzalez and Sahni 1978). While the majority of research into flow shop scheduling is focused on the serial-type flow shop, a three-machine assembly- type flow shop-scheduling problem was studied by Lee *et al.* (1993). Parallel machine scheduling problem (*P*k*C*max) has been proved by Garey and Johnson (1978) as NP-hard in a strong sense when the number of machines is unlimited. However, the problem is solvable in pseudo-polynomial time when the number of machines is fixed and thus NP-hard only in the ordinary sense. A recent survey on parallel machine scheduling problems was by Cheng and Sin (1990). Blazewicz *et al*. (1991) proposed optimal algorithms. However, most algorithms developed for solving *P*k*C*max are heuristics (e.g. Graham 1969). Except for the scheduling problems solved by Kusiak (1989), none of the related scheduling problems solved in the literature considered product structures represented by the simple and complex digraphs even though they provide the best structural information of the products. Lee and Vairaktarakis (1998) considered a similar system structure in their paper to the problem discussed in this paper and developed heuristics with worst-case error bounds. This structure allows the implementation of the product differentiation concept. A three-machine assembly-type flow shop-scheduling problem was studied by Lee *et al.* (1993).

3. Agile Job Shop Scheduling Problem Model Formulation

The agile job shop scheduling problem deals with the allocation of jobs to different machines in agile environment over time span. It is a decision making process in agile environment with the goal of optimizing more objectives to satisfy the need of agility. Suppose there are n jobs $(J_1, J_2, \ldots, J_n)$, which is to be processed on m machines $(M_1, M_2, \ldots, M_m)$, and these jobs are subject to many constraints. The optimal solution is to find out for given objective functions and constraints. Due to agile environment some objectives and constraints may possess uncertainty of type fuzzy, stochastic, or others in nature. The completion of job J_i consists of a sequence of n_i operations $O_{1i}, O_{2i}, \ldots, O_{n_i i}$, which is called as its machine list. The precedence of the machine list is defined as $O_{ti} \rightarrow O_{t+1,i} (t = 1,2, \ldots, n_i - 1)$. For a given operation O_{ti}, denote by St_{ti} the time units needed to process job j_i on machine $\mu_{ti} \in \{M_1, M_2, \ldots, M_m\}$.

Notation: $\boldsymbol{J} = \{\boldsymbol{J_1}, \boldsymbol{J_2}, \ldots, \boldsymbol{J_n}\}$:set of jobs, where n is the number of jobs.
$M = \{M_1, M_2, \ldots, M_m\}$:set of machines, where m is the number of machines.
k = number of operations to be performed.
S_{jk} = slack of operation *k* with respect to the part type *j*.
SP_{ik} = processing time of job j on machine m with respect to operation *k*, a stochastic variable subject to normal distribution
Y_m = mean repair duration on machine *m*.
η_m = mean rate at which breakdowns occur.
K_T = Part type counter.
$E_{j,k}$ = Starting time of operation *k* for part *j*.
C_{jk} = completion time of *k* for part type *j*.
Dj = Distance between the part types.
Z_w = extrapolative schedule.

V (a, b)= length of the longest path from a to b.
χ = processing speed or capacity.
ϑ = an increasing function of variability.
$\mathfrak{A}$= an increasing function of agility.
$E(\Psi_m)$= expected flow time.
λ_j = part processing time of part j.
μ = ratio of processing to part inter-arrival time.
α_a = COV of the processing time.
β_b = COV of the part inter-arrival time.
φ = part arrival rate.
E_{jk} = starting time of the operation k on part j.
P^s = extrapolative schedule.
Y_n = minimum time processing part j .
$G(X)$= Gaussian probability distribution.
$P(X)$= Poisson's probability distribution.
p_m = priority of the machine.
f_j = mean time between failures.

Each part type requires an operation on the corresponding machine with an average processing time $1/\lambda_j$. The part inter-arrival and processing times are exponentially distributed with respect to the means $1/\varphi_j$ and $1/\lambda_j$. Symbol α_a^2 and β_b^2 refer, respectively to the coefficients of variance. Higher values of α_a^2 correspond to the higher variability in part type arrivals and can be used to indicate higher part type demand variability and predictability. The values of β_b^2 explain the variability in part processing times that is in the model to represent the variability in the processing capabilities of the machine, or the processing requirements of the part types. The part related variability (ϑ) is due to part variety in the product mix or too frequent changes in design and manufacturing specifications of the part types and is expressed in equation (1).

$$\vartheta = \frac{(1+\beta_b^2)(\alpha_a^2+\mu^2\beta_b^2)}{2(1+\mu^2\beta_b^2)} * P(X) \quad \ldots (1)$$

The coefficient of variance β_b is mathematically expressed as,

$$\beta_b^2 = \lambda^2 \sum_{j=1}^{J} \frac{\varphi_j}{\varphi}\left(\frac{1}{\lambda_j}-\frac{1}{\lambda}\right)^2 \quad \ldots (2)$$

The overall average arrival rate is expressed as,

$$\varphi = \sum_{j=1}^{J} \varphi_j \quad \ldots (3)$$

and average processing time is expressed as,

$$\frac{1}{\lambda} = \sum_{j=1}^{J} \frac{\varphi_j}{\varphi}\frac{1}{\lambda_j} \quad \ldots (4)$$

The effect of agility on the performance can be easily shown to increase in magnitude as variability in either processing or demand increase. That is, the performance improvement due to agility rises in significance as variability increases. The agility plays a major role in determination of the performance measures of the system, thus agility is expressed as an increasing function following the Gaussian probability distribution and shown in Figure 1.

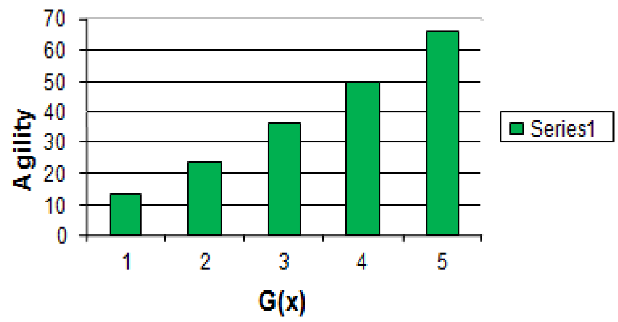

Fig. 1 Agility versus Gaussian distribution function

$$\Re = \left(\frac{m\lambda-\varphi}{\pi m}+\frac{R_z}{m}\right) * G(x) \quad \ldots (5)$$

The priorities for the machines are evaluated as follows:

$$p_m = \frac{f_j}{f} \quad \ldots (6)$$

Objective functions:

$$\text{Min } Z_w = \sum_{jk \in O_1} \max\{E[RD_{jk}] - S_{jk,J}(Z_w), 0\} + Y_n \quad \ldots (7)$$

$$\text{Min } E(\psi_m) = \chi + \frac{\vartheta}{\Re} \quad \ldots (8)$$

$$\text{Min } \aleph_{j,k}^{n} = TJ_{j,k} + \min\{G_{n1}\}; \forall_j, \forall_k, \forall_n \quad \ldots (9)$$

j= 1, 2, 3 …J; k=1, 2, 3…K
Subject to :

$$\sum_{j=1}^{J}\sum_{k=1}^{K} W_{jkm} \le \chi_m \quad \text{… (10)}$$

$$\Re \le 5 \quad \text{… (11)}$$

$$\vartheta \ne 0 \quad \text{… (12)}$$

$$E_{jk} > C_{jk} \quad \text{… (13)}$$

$$\beta_b > m - 1 \quad \text{… (14)}$$

$$\sum_{j=1}^{J}\sum_{k=1}^{K} MT_{jk}(m_{jk}) \le TA_m \quad \text{… (15)}$$

4. Solution Methodology

The proposed genetic algorithm (GA)

Step 1: Assign number of generation n = 1. Assign the values of population size (P), maximum number of generation (G) and T (1).
Step 2: Randomly generate a set of population size chromosomes as initial parent population.
Step 3: Compute the fitness (X1) for each parent.
Step 4: By using crossover and mutation produce children from each parent.
Step 5: Compute fitness function of each child of every family. Select the best one in every family according to having highest fitness value (X2).
Step 6: Compute ΔX = X2 – X1.
Step 7: Get the parent for next generation out of each family, adopting following transition rules: If (ΔX>0 or F (T (n), ΔX)>γ) best child is accepted as parent for new generation. Else the earlier one remains as new parent.
Step 8: Reduce the temperature as per following cooling schedule:

$$T(n) = \frac{3.2 * T(1)}{1 + \log(T^n(1))}$$

Step 9: Perform n = n+1.
Step 10: Select the best one of the final population according to having highest fitness value. This gives the optimal or sub-optimal solution.

5. Results and discussions

The crossover probability is taken to be 0.6 and mutation rate 0.02. The initial temperature was considered to be 600 and final temperature was 20 in the applied algorithm. To show the impact of agility on the flow time, the data sets are prepared with the incorporation of agility under the similar scenario. System performance is obtained for various levels of variability and it is achieved by gradually increasing the variance in the part inter-arrival times and processing times. The effect of agility shows a diminishing rate of return curve for all levels of variability, it also shows that effect of agility is particularly significant when either demand or processing variability is high. With increasing agility after certain level the flow time remains almost unaffected (figure 2). The results of the data sets under such breakdown scenarios, after successive number of iterations reflect the superiority of the incorporated algorithm to converge towards the optimality. The results comparison of the average flow time with respect to the agility measures has been shown in figure (2) and (3). The plot for the time taken versus the agility is shown in figure (4).

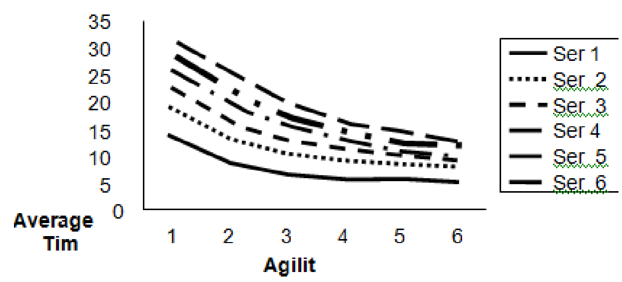

Fig. 2 Flow time versus Agility (α= 6, μ=.06)

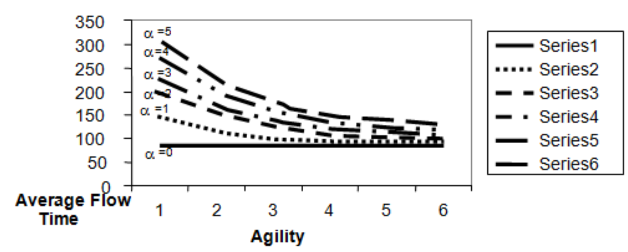

Fig. 3 Flow Time versus Agility (β=6, μ=.06)

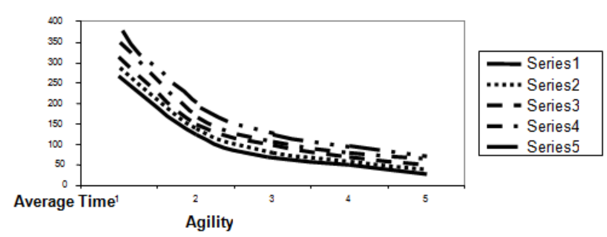

Fig. 4 Average time taken versus agility

The proposed GA approach has been also compared with some standard priority rules and results are much better than those obtained from the priority rules (Figure 5). These comparisons show significant improvement in the results on applying the GA algorithm and the results converge towards the optimality nearly after (40) iterations.

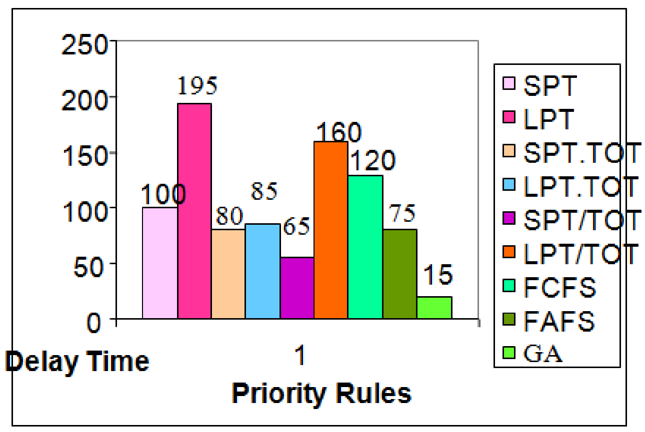

Fig. 5 Comparison of GA with priority rules

6. Conclusion

In this proposed work the performance measurement under agile manufacturing system has been studied. In this research we have incorporated genetic algorithm (GA) to enhance the performance under such unpredictable scenario. To tackle the existing uncertainties such as customers' preference, machine breakdowns, deadlocks etc. in the job shop environment first an extrapolative schedule is generated that is modified when an unexpected event occurs. The adequate slack is inserted in the extrapolated schedule to absorb the undesirable impact of interruptions. The main intention is to optimize the performance measures of the agile manufacturing system. The paper also focuses on the various aspects of the impact of agility over the performance of the system under dynamic conditions. The present work deals with the objectives of minimizing the average delay time and average flow time. The result of the proposed approach reveals the superiority of the algorithm in solving such complex problems. In our view the proposed approach can be extended to cover more practical situations. There is also wide scope for improving the performance of the considered agile manufacturing system. The ability of the GA algorithm to converge towards the optimality in less computational time, and escaping the local optima, lefts its scope of further extension in other complex scenarios. The real time problems are more complex than those considered in this paper. Hence there is need of further study in this area involving more constraints and objective functions. The cooling schedule of the GA algorithm can be further improved to give much better results.

References

1. *Blazewicz J Dror M and Weglarz J (1991), "Mathematical programming formulations for machine scheduling: a survey", European Journal of Operational Research, Vol. 51, 283- 300.*

2. *Brah S A and Hunsucker J L (1991), "Branch and bound algorithm for the flow shop with multiple processors", European Journal of Operational Research, Vol. 51, 88 -99.*

3. *Brah S A Hunsucker J L and Shah J R (1991), "Mathematical modeling of scheduling problems", Journal of Information and Optimization Sciences, Vol. 12, 113 - 137.*

4. *Chen B (1995), "Analysis of classes of heuristics for scheduling a two-stage flow shops with parallel machines at one stage", Journal of Operational Research Society, Vol. 46, 234 - 244.*

5. *Cheng T C E and Sin C C S (1990), "A state-of-the-art review of parallel-machine scheduling research", European Journal of Operational Research, Vol. 47, 271 - 292.*

6. *Goffman E G Garey M R and Johnson D S (1978), "An application of bin-packing to multiprocessor scheduling", SIAM Journal of Computing, Vol. 7, 1 - 17.*

7. *Gonzalez T and Sahni S (1978), "Flowshop and jobshop schedules: complexity and approximation", Operations Research, Vol. 26, 36 - 52.*

8. *Gupta J N D (1988), "Two-stage hybrid flow shop scheduling problem", Journal of Operational Research Society, Vol. 34, 359 - 364.*

9. *Gupta J N D and Tunc E A (1991), "Schedules for a two-stage hybrid flow shop with parallel machines at the second stage", International Journal of Production Research, Vol. 29, 1489 - 1502.*

10. *Hunsucker J L and Shah J R (1994), "Comparative performance analysis of priority rules in a constrained flow shop with multiple processors environment", European Journal of Operational Research, Vol. 72, 102 - 114.*

11. *Johnson S M (1954), "Optimal two and three-stage production schedule with setup times included", Naval Research Logistics Quarterly, Vol.1, 61- 68.*

12. *Kusiak A (1989), "Aggregate scheduling of a flexible machining and assembly system", IEEE Transactions on Robotics and Automation, Vol. 5, 451 - 459.*

13. *Lee C Y Cheng T C E and Lin B M T (1993), "Minimizing the make span in the 3- machine assembly-type flow shop scheduling problem", Management Science, Vol. 39, 616 - 625.*

14. *Lee C Y and Vairaktarakis G L (1998), "Performance comparison of some classes of flexible flow shops and job shops", International Journal of Flexible Manufacturing Systems, Vol. 10, 379 - 405.*

15. *Srikandarajah C and Sethi S P (1989), "Scheduling algorithms for flexible flow shop: worst and average case performance", European Journal of Operational Research, Vol. 43, 143 - 160.*

