Production Scheduling for Flexible Flowshop using Genetic Algorithms

Boonchuay Srithammasak[1] *and Weerapan Sae-Dan*[2]
[1,2]*Major of Computer Engineering, Faculty of Engineering, Ramkhamhaeng University*
Emails: boonchuay@ru.ac.th[1], *s.weerapan@ru.ac.th*[2]

ABSTRACT This paper was to study the production scheduling for the industry enabling to change the product form according to customer's order. From a variety of products lengthened the process steps and time. The paper studied how to schedule the production table for increasing the efficiency of flexible flow system scheduling. The two state units were experimented. Each unit consisted of unrelated parallel machines. Then, based on the customer's needs, the flexible flow system production tables were scheduled with the genetic method. The experiment with data characterized by a normal distribution as the number of 20, 50, 80 and 107 jobs was implemented. The results showed that the production scheduling with the genetic method enabled to reduce the production time from 14 days into 12.4 days, or 11.42 percent, and the time analysis of the program processing had the significant relation to the number of the jobs, and the number of the search model.

KEY WORDS -- *genetic algorithms; production process; unrelated parallel machine; flexible flowshop*

1. Introduction

An industry changes production form according to the customer's order. Unrelated parallel machines in the industry are capable of different production. The products are various. The production scheduling is complicated and complex. Moreover, the production is delay and loses the overtime cost. The arrangement of the production scheduling in the industry is flexible flow shop. The schedule comprises two units of the state. Each consists of the unrelated parallel machines. The first unit contains 12 unrelated parallel machines as the last does 17 ones.

2. Related Research

The production scheduling of the parallel machine system [1] manages job sequence on parallel machines line as follows: 2 and more machines units, which are the same unit and equal performance, will spend production time equally when there are several jobs entering the system and select every production unit, arrange the sequence of jobs on each unit. Each job whether it be to work on any given production unit will equally spend the time.

(1) Flexible flow machine combines between flow shop machine and parallel machine production. The hybrid flow shop or the multiple processor flow shop system is so called. The system comprises multiple stages; s=1, 2, 3, …, S. Each stage consists of the parallel processors, as depicted on Fig. 1

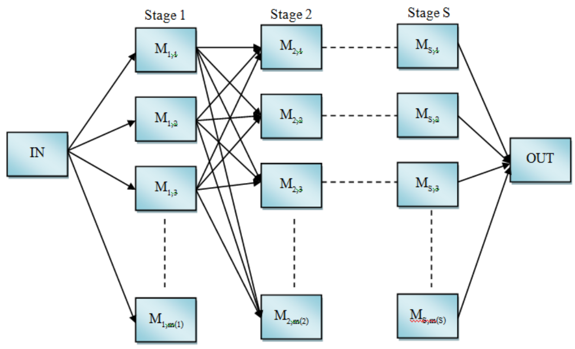

Figure 1 Job structure in the multiple processor flow shop system

From the Fig. 1, there exist many of the production stages. The parallel processors or machines are identical, uniform, or unrelated.

(2) The production scheduling with genetic algorithms [2] simulates or create multi-format production scheduling solutions calculation. The far better solution (Fitness) is selected. The solutions depend upon the objective of the production scheduling, and then pass through the crossover and mutation processes. The solution is calculated and compared with the fitness solution. The best solution is used in the selection procedure for scheduling the next round. The production scheduling, whose solutions are not required, is destroyed. Finally, the result of the best production scheduling is remained as the final solution. From the study on the provision of jobs for the workers [3], there will exists different noise exposure and different working time in each day. This paper has already led to the genetic algorithm including the aim to study and compare the results received from the different crossover process, the normal one (Regular Sampling Space) and the extension one (Enlarging Crossover Space.) The normal process or the regular sampling space is the general method of crossover when chromosome from the crossover substitutes the one from the parents while in the enlarging crossover space, the parents' chromosome is still kept. The paper concludes that the regular sampling space approach is more appropriate in the solution of the work assignment; moreover, such an approach requires the less version of population for getting the right answer.

2.1 Research Methodology

The procedure of the genetic algorithm applies for the solution of organizing multiple tasks, multiple machines. The genetic procedure comprises the following stages:

(1) Initial Population
To generate a certain amount of preliminary answers is used in the process of genetic programming. The preliminary answer is the population (popsize) used the random machine selection method. All preliminary answers from the random popsize must be unique in order to prevent the solutions from the genetic programming not to be the local optimal. Besides, the difference of the preliminary answer allows the less popsize.

(2) Evaluation
Before entering the selection process of the genetic algorithm, it is necessary to evaluate how each of preliminary answer is appropriate. This appropriation is from the fitness of each binary. The fitness means the object functions which make the least time of the work flow in the system:

$$ES_{i,m} > \sum_{i=1}^{I} FT_{i-1,m} \qquad (1)$$

$$given\ WT_{i,m} = ES_{i,m} - \sum_{i=1}^{I} FT_{i-1,m} \qquad (2)$$

$$FT_{i,m} - \max[(WT_{i,m} + PT_{i,m}) + \sum_{i=1}^{I} FT_{i-1,m}] \qquad (3)$$

$$if\ ES_{i,m} \leq \sum_{i=1}^{I} FT_{i-1,m} \qquad (4)$$

$$given\ FT_{i,m} = \max[PT_{i,m} + \sum_{i=1}^{I} FT_{i-1,m}] \qquad (5)$$

define

FT	=	*time of job flow*
ES	=	*least time for job starting*
WT	=	*waiting time of job*
i	=	*all job in the production schedule i=1,2,3,..,I*
m	=	*the number of machine*

After that, the minimal value of the fitness is kept in the elite solution before entering the genetic programming

(3) Selection
The selection of the answer is to bring a group of preliminary answers into the selection process through qualifying by the Fitness of answer each one. By creating a roulette wheel, a circle with an area of 1 unit area is divided into sections as the number of answers for each model (equal Popsize). The area of each section has the same size as the probability of being selected each answer. How to make is as follows:
3.1 Find the total of the Fitness (F) with the size of Popsize as of Equation 6.

$$F = \sum_{i=1}^{Popsize} X_i \qquad (6)$$

given X_i = The Fitness of the answer i

3.2 Find the probability of each answer selection (P_i) as of Equation 7.

$$P_i = \frac{f(x_i)}{F} \qquad (7)$$

when i=1,2,3,…Popsize

3.3 Find the cumulative probability of each answer selection (q_i) as of Equation 8.

$$q_i = \sum_{j=1}^{i} P_i \qquad (8)$$

Then spin the roulette wheel and follow the steps below.

1) Generate a random number, r, whose range is between 0 and 1. The first value is r_1.
2) If $r_1 < q_1$, then choose the first answer.
If $q_{i-1} < r_1 < q_i$, then choose the answer i to be the first answer.
3) Generate another r. The second value is r_2.
4) if $r_2 < q_2$, then choose the second answer.
If $q_{i-1} < r_2 < q_i$, then choose the answer i to be the second answer.
5) Match the Fitness of both answers into the Mating Pool
6) Repeat step 1 – 5 until the answer in the Mating Pool equal to Popsize.

(4) Crossover
The crossover method is to find a crossover point to one location and then bring the end of the string attached to the header of one answer. This crossover method starts from setting a crossover point, Pc. The length of binary position is represented by "|".

(5) Mutation
The swap of the value position within a single answer is to swap the position of value in the string. This will get a chance of a new answer which is highly not possible to be an answer. The mutation, depending on the probability of the mutation (Pm), will get involved. The methodology is to randomize one position and then make the mutation. The mutation value is between [0, 1].

(6) GAs-Loop
If the experimental generation is less than the maximum generation, then repeat (1) to (5) If not, then go to (7)

(7) Stop
The experiment stops searching the solution if the generation is the full amount specified or the minimal Fitness value of each generation has changed very little.

3. Experimental Results

Three main factors affect to production scheduling with genetic algorithm. They are the preliminary population (Popsize), Crossover Probability (Pc), and Mutation Probability (Pm).

(1) Factor Test
Before using the genetic method for manufacturing scheduling, all three factors are tested for appropriate values.

Table 1 Factor and Its level for parameter test

Factor	Low	High
Generations	250	500
Crossover Probability	0.8	0.95
Mutation Probability	0.01	0.02

Data is analyzed using ANOVA. The results show as Fig. 2.

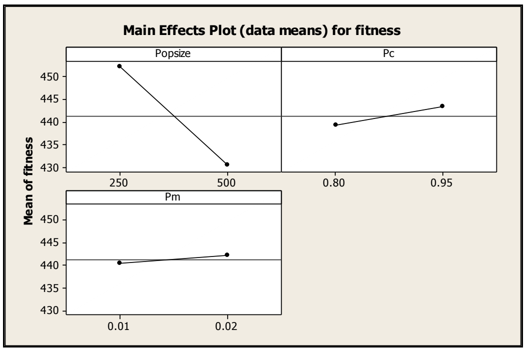

Figure 2 Mean of each main factor

The analysis shows that factors, that impact on the response (Flow time), affect mainly the probability of crossover (Pc), probability of mutation (Pm) and population (Popsize) contribute significantly to the 95 percent confidence level. Therefore, the appropriate test (Response Optimization) for manufacturing scheduling using the genetic algorithm follows: Initial population of 500 people, the crossover probability at 0.8, and mutation probability at 0.01.

(2) Relationships of manufacturing scheduling program runtime. The relationships between work flow in the system and the generation search including the number of work for the answer search with the genetic method are experimented. Data set of 20, 50, 80 and 107 jobs is randomized. Three samplings are tested. The generation search, 100, 500, 1000, 1500, 2000, 3000, 5000, and 10000 generation, is used for correlation analysis of the experimental results with the following three steps: 1. Verify the Residual plot, 2. Analyze coefficient correlation, and 3. Analyze the regression equation. The results show the analysis of the P-Value is equal to 0.000. It enables to conclude that the time to process the applications are related to the number of models in search and job number. It also show that the correlation equation for the time to run the model with the number in the search and the number of jobs is from equation 9.

R.T= - 20.9 + 0.0221 Generation + 0.885 Job (9)
with R.T = run time

The relationships between program run time and the generation search including the number of work depicts in Fig. 3.

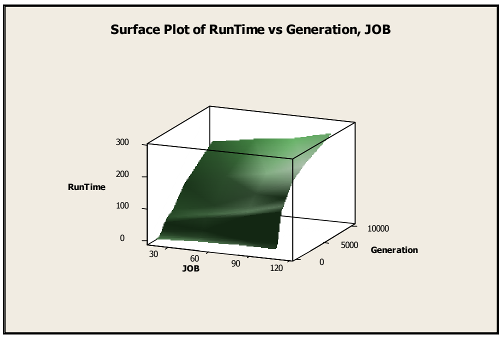

Figure 3 The relationships of program run time and the generation search including the number of work

(3) Comparison on times of jobs flow in the system. The comparison on new manufacturing scheduling to the traditional one is that for every 20, 50, 80, and 107 jobs with 500 generation of search (from experiment), the results show in the Table 2.

Table 2 Comparison on the time of work flow in the system

No. Jobs	Δ (stage1) %	Δ (stage2) %	Δ (total) %
20 jobs	11.25	22.71	18.89
50 jobs	15.52	30.58	23.44
80 jobs	15.50	9.72	12.35
107 jobs	14.86	36.00	15.48

4. Conclusion

From the application of the solution search by genetic methods with dividing the production process having the goal of the operation, to reduce the flow of the system to a minimum, it is found that the lead time of 9 days can be reduced to 7.4 days and result in the lead time in the industry, from 14 days to 12.4 days, leading to a faster response to customers.

References

[1] Bor-Wen Cheng, Chun-Lang Chang., *"A Study on flow shop scheduling problemcombining Taguchi Experiment design and genetic algorithms."* Expert System withApplication, 2006.

[2] Chae Y. Lee, *"Genetic Algorithm for single Machine Job Scheduling with Common Due Dateand Symetric Penalties"*, Journal of the Operation Research Society of Japan, Vol37, No.2June 1994.

[3] Fink, A., and Vob, S.,*"Soloving the continuous flowshop scheduling problem bymetaheuristics"*, European Journal of Operation Research 151, 400-414, 2003.

[4] Khammuang, K., *"A Genetic algorithm for the cell formation problem with routingflexibility"*, Master thesis, Asian Institute of Technology, Thailand, 2000.

[5] Kumar, S. and Harms, R.,*"Improving business process for increased operationalefficiency: a case study"*, Journal of Manufacturing Technology Management, 15(7), 667-674, 2004.

[6] Masato Watanabe, Kenichi Ida and Mitsuo Gen., *"A genetic algorithms with modifiedcrossover operator and search area adaptation for the job-shop scheduling problem."*
Computers & Industrial Engineering. 48(2005): 743-752.

[7] Melanie Mitchell., *"An Introduction to Genetic Algorithms*", London: MIT Press, 1998.