유전자 알고리즘을 이용한 유연생산시스템의 작업프로세스 스케쥴링분석

김정원°, 이종근
창원대학교 전자계산학과 정보통신연구실
경남 창원시 사림동 9, 641-773
Tel, Fax : +82-551-79-7420
e-mail: jklee@sarim.changwon.ac.kr

WIP ANALYSIS OF FLEXIBLE MANUFACTURING SYSTEM BY GENETIC ALGORITHMS

Jung-Won Kim, Jong-Kun Lee
LICS / Department of Computer Science Changwon National University,
Changwon, Kyungnam, 641-773, South Korea

Abstract

In this paper, we suggests a WIP(work in process) of FMS analysis methods based on the Genetic algorithm. We conjoined both the assignment and the scheduling problem in order to create a new representation scheme for a chromosome and a mutation operators.

1. INTRODUCTION

To make an efficiency and optionally schedule iterative is one of the critical issues in FMS(Flexible management system). The general problem of scheduling in FMS is in the scheduling a set of partially ordered operation onto a FMS. Also, this schedulling problem is complex even for simple formulation and is NP-hard in many cases. Various scheduling methods have been proposed by many researches[13-18,21,22], based on the ever increasing complexibility and flexibility of manufacturing.

This paper presents an efficient method based on genetic algorithms to solve the FMS scheduling problem. Genetic algorithms have recently received much attention as robust stochastic searching algorithms for various optimization problem. This class of methods is based on the principles of natural selection and natural genetics that combine the notion of suvival of the fittest, random and yet structured search, and parallel evaluation of nodes in the search space. In this work, we apply the genetic algorithm to provide a WIP analysis for minimizing a makespan of the job, and propose a new method to encode the solution of the WIP scheduling, which used the chromosous represented by a set of nodes in parallel.

This paper is organized as follows. In section 2, we introduce a illustartion model for FMS and some related definitions. The genetic algorithm is also given. The representation of the search nodes and a method for generating initial proposition are pesented in section 3.

Section 4, a detailed description of the genetic process is employed to solve the model. Finally in section 5, a conclusion is given.

2. MODEL AND DEFINITIONS

A set of partially ordered scheduling problem in FMS can be represented by a Petri nets TPN=(P,T,F), where P $\cap$T= and F:(P x T) $\cup$ (T x P) -> {0,1}. The elements of P are called places, the elements of T are called transitions and F is the flow relation $\Rightarrow$ exists between the transitions. If Ti $\Rightarrow$ Tj, then job Ti must be completed before Tj can be initiated.

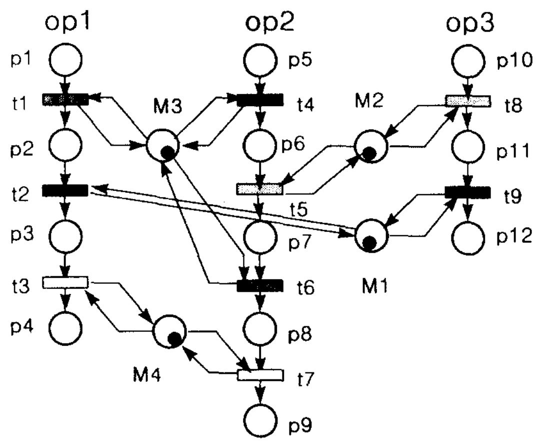
As a starting point of a systematic construction of the model, we take the operating sequences which specify the precedence constraints of the operations for each type of product[1]. Then, by adding shared resource, the mutual exclusion constraints between the operations was carried out by the same machine are modeled (Fig. 1).

We assume the following[3]:

1) no machine may process more than one job at a time,
2) no job may be processed by more than one machineat a time,
3) the sequence of machines where each job is completely specified has a linear precedence structure,
4) process times are known.

We consider a system with 4 machines M1,M2,M3 and M4 and three jobs, and the operation sequences of these jobs are presented as follows:

Job1: OP11M3,OP12M1,OP13M4
Job2 : OP21M3, OP22M2, OP23M3,OP24M4
Job3 : OP31M2,OP32M1.



(Fig. 1) production model

<Tab. 1> Table of transitions in (Fig.1)

transition	time	works
t1	13	OP1
t2	10	
t3	11	
t4	20	OP2
t5	7	
t6	6	
t7	14	
t8	4	OP3
t9	7	

3. THE GENETIC ALGORITHM

The theory of GA(Genetic Algorithm) has been proposed by many researchers [2-5]. Here, we briefly recall its main definitions.

3.1 Fundmentals of Genetic algorithm

A GA is based on the evaluation system and herediatry. In the evaluation process, the population undergoes sexual recombination (crossover); mutation and natural selection that select the best individual for the next generation, and so on.

A GA consists of the following steps(4):

1) Initialization: an initial population of the search nodes is randomly generated.
2) Evaluation of the fitness function: the fitness value of each node is calculated according to the fitness function.

3) Genetic operation: new search nodes are generated randomly by examining the fitness value
of the search nodes and applying the genetic operators to the search nodes.
4) Repeat step 2 and 3 until the algorithm converges.

The construction of GA for any problem can be separated into four distinct and yet related operations[4].

1) the choice of the representation of the strings,
2) the design of the genetic operators,
3) the determination of the fitness function,
4) the determination of the probabilities controlling the genetic operators.

3.2 Chromosome syntax(Encoding)

The application of a genetic algorithm to the problem of generating FMS scheduling requires an encoding of the solution in the problem domain. For solving this complex scheduling problem, the chromosome syntax may be changed to a set of machine put in parallel where each machine is a vector which contacting the assignment operation for this machines[1]. The ordering of machine and the job modeling of this work are represented in <Tab.2> and <Tab.3> respectively.

<Tab.2> Ordering of Machine

Job no.	Ordeing of machine
Job1	M3M1M4
Job2	M3M2M3M4
Job3	M2M1

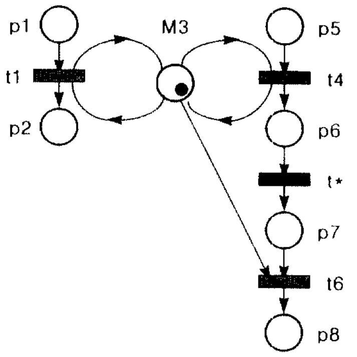
<Tab.3> Ordering of Jobs

Machine no.	Ordeing of Jobs
M1	Op11,Op31
M2	Op22,OP31
M3	Op11,OP21,OP23
M4	Op13,OP24

The machine operation in the job of models defined in section 2 are summarized as follows:

1) Machine 3(M3):

Machine 3 involved three works in two process like as (Fig.2).

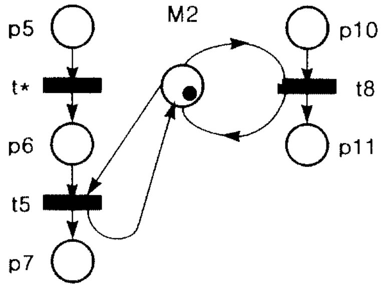


(Fig.2) Operation of M3

2) Machine 2(M2):

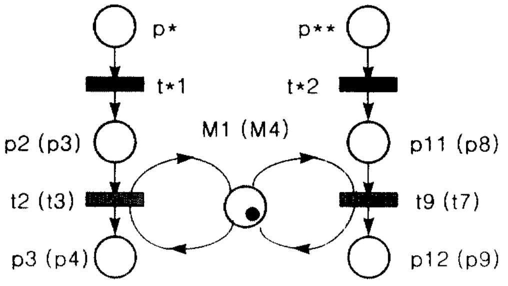
Machine 2 involved two works(Op2 and Op3) in two process like as (Fig.3).

(Fig. 3) Operation of M2



3) Machine 1 and 4(M1,M4):

Machine 1(4) involved two works in two process like as (Fig.4) .



(Fig.4) Operation of M1(M4)

We defined the operation are represented by three
terms like as in [1]: the first is number of operation, second is the job number, and the third is the starting time of the operation. One

chromosome has a parallel representational scheme like as follows:

M1	(2,1,13), (2,3,4)
M2	(1,3,0)(2,2,33)
M3	(1,1,0) (1,2,13) (3,2,40)
M4	(3,1,23) (4,2,46)

3.3 Crossover

The role of the crossover is to generate a better solution by exchanging information on contained in the current good ones. But in our model, this crossover operation made new information more complex than before, because each machine has it's own work and can't exchange its operation with another. So we ignored this operation in this work.

3.4 Mutation

Although the crossover tries to improve the fitness in its offsprings, the chances of offsprings converged in a local solution exists. During the repair of this situation, there is an escape mechanism called mutation, which changes genes in the same list randomly with little probability. Mutation provides and maintains diversity in a population. It typically works with a single chromosome, and always creates another chromosome.

3.5 Fitness function

During each iteration, the individuals in the current population are evaluated, using some measure of fitness. There is a number of characteristics of the evaluation function that enhance or hinder an evaluation program's performance. In the most optimization applications, fitness is calculated based on the natural objective function. In our case, the fitness function is the minimizing a WIP. For each chromosome we have M machines, we calculate the time to execute all the assigned operations and we take the-maximum time of the m machines. This time is calculated as follows[1]:

1) We take the starting time of the last operation in the performed machine(Tsl)

2) We add the Tsl to the processing time of this operation, we obtain (Tsl + Pijk).

We repeat these two steps for all machines in the chromosome, and the WIP of this chromosome is calculated as follows[1]:

$$\text{Max } [(Tsl + Pij\ k)_{1},(Tsl + pili\ k)_{2}\ (Tsl+pij\ k)_{m}\ ]$$

3.6 Initial population[1]

The choice of the generation lays an important part in the research of the good solution. For this purpose, we diversified the first generation, and used a combination of the priority rules.

4. COMPUTATION OF SOIUTION

This section presents a detailed description of the genetic process employed to solve the modeling problem.

The operating sequence of these jobs are

Jobl:01, 02, 1 03, 1,
Job2:01, 202, 203, 2, 04,2
Job3:01, 302, 3.

1) Chromosome

The basic chromosome of this model is shown in (Tab.4).

(Tab. 4) The basic chromosome of (Fig.1)

M1	(2,1,13), (2,3,4)
M2	(1,3,0)(2,2,33)
M3	(1,1,0) (1,2,13) (3,2,40)
M4	(3,1,23) (4,2,46)

2) Mutation

We consider this as follows:

(Tab. 5) Chromosome 1

M1	(2,1,13), (2,3,4)
M2	(1,3,0)(2,2,33)
M3	(1,1,0) (1,2,13) (3,2,40)
M4	(3,1,23) (4,2,46)

We suppose that the first of operation of the jobl is selected at random, which was previously assigned to machine M2(because M2 involved to first operation job3). To respect the precedence constraints we placed it before operation job3. We have recalculated the starting time for this chromosome, and we obtained the following new chromosome:

(Tab. 6) chromosome 2

M1	(2,1,13), (2,3,4)
M2	(1,3,0)(2,2,33)
M3	(1,1,0) (1,2,13) (3,2,40)
M4	(3,1,23) (4,2,46)

3) Fitness function

In our model, we can consider two chromosomes for fitness function like as (table 5 and 6). We selected one chromosome minimizing the makespan by the fitness function.

Take the starting time of the last operation and add its own processing time:

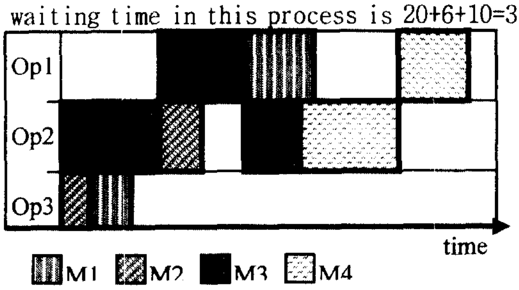
chromosomel: (Tsl + Pijl) = 13 + 10 = 23
(Tsl + Pij2) = 20 + 8 = 28
(Tsl + Pij3) = 31 + 6 = 37
(Tsl + Pij4) = 43 + 11 = 54

The makespan of chromosome 1 is max (23,28,37,54) = 54.

Chromosome2: (Tsl + Pijl) = 13 + 10 = 23
(Tsl + Pij2) = 33 + 7 = 40
(Tsl + Pij3) = 40 + 6 = 46
(Tsl + Pij4) = 46 + 14 = 60

Also, the makespan of this chromosome2 is 60.

We selected one schedule process based on chromosome2. Finally, we made a process schedule table based on the machine process works(Fig. 5). Total waiting time in this process is 20+6+10=36.



(Fig. 5) Schedule table based on the operation procsss

5. CONCLUSION

The application of genetic algorithms to a WIP scheduling problem with real-word constraints has been defined. A parallel representational schemes[1] for the genetic operations has been proposed using the original item of coding the assignment and scheduling problem. We suggest that this result is equal to our previous work which analyzed the scheduling problem using the Time Petri nets unfolding methods[7]. In this work, we showed how GA is used to analyze the WIP analysis. We are working on the simulation of our method for a more complex data structure continuously.

REFERENCES

[1] K.Mesghouim.5.Hamachi,P.Borne,Production Job-shop Scheduling using Genetic Algorithm, in proceeding IEEE-SMC'96,pp.1519-1524.

[2] K.Mesghouim,S. Hamachi,P. Borne,Evaluation Program for Job-Shop Scheduling,in Proceeding IEEE- SMC'97, pp .720-725.

[3] BM.Gen ,Y.Tsujimura,5. Kubota.Solving Job-Shop scheduling Problem by Genetic Algorithm,in proceeding IEEE-SMC'94,pp.1577-1582.

[4] E-S.H,Hou,N.Ansari,H.Ren,A Genetic Algorithm for Multiprocessor Scheduling, IEEE tr. Of Parallel and Dist. sys.,vol.5,N.2,1994,pp.113-120.

[5] Z. Michealewicz, Genetic Algorithm + Data Structure = Evaluation Progress, Spring-Verlag,1992.

[6] J.J.Park, C.H.Park, Application of Genetic Algorithm to Job-Shop Scheduling Problem with Active Schedule Constructive Crossover, in proceeding IEEE- SMC'95,pp.530-535.

[7] J.K.Lee, WIP analysis of the FMS using TPN unfolding, will appear in proceeding IEEE-SMC'98