International Journal of the Physical Sciences Vol. 6(14), pp. 3557–3563, 18 July, 2011
Available online at http://www.academicjournals.org/IJPS
DOI: 10.5897/IJPS11.463
ISSN 1992 - 1950 ©2011 Academic Journals

Full Length Research Paper

Distributed flexible manufacturing system (FMS) scheduling using Memetic algorithm, particle swarm optimization and timed Petri net

Majid Aboutalebi[1], HosseinShirgahi[2]* and Homayon Motameni[1]

[1]Islamic Azad University, Sari Branch, Sari, Iran.
[2]Young Researchers Club, Jouybar Branch, Islamic Azad University, Jouybar, Iran.

Accepted 16 May, 2011

Allocating jobs to the best factories and scheduling them are two important problems in distributed flexible manufacturing systems that are among NP-hard problems. Many different intelligent algorithms have been proposed for these problems. In this paper, two new algorithms were proposed for distributed flexible manufacturing system (DFMS-MPN and DFMS-PPN algorithms), in which one of them was based on Memetic algorithm and the other was based on the particle swarm optimization method. In the proposed method, the distributed flexible manufacturing system was modeled by Timed Petri net and then a scheduled task was programmed by Memetic algorithm and particle swarm optimization method. The experimental results showed that the proposed method has reasonable performance in comparison with other algorithms.

Key words: Distributed flexible manufacturing system, scheduling, Memetic algorithm, particle swarm optimization, timed Petri net.

INTRODUCTION

Recently, many researches have been done for the distributed flexible manufacturing system (DFSM). Usually, DFSM include two main subjects. Firstly, choosing the best factories for allocation of jobs and secondly, scheduling tasks in the selected factory. The aim of DFSM is to reduce scheduling time. The mentioned problem is among the NP-hard problems and usually, it cannot be solved by classical algorithms. Consequently, in solving these problems, intelligent methods were used.

Up to now, many different methods have been developed. Memetic algorithm (Yadollahi and Rahmani, 2009), ant colony optimization (Caballero and Mej.a 2004; Kumar et al., 2003), Petri net and genetic algorithm are some of these methods, but one of the best methods is genetic algorithm. The advantages of using Petri net are the simulation and performance evaluation ability of the method (Chan et al., 2006; Mej.a and Acevedo, 2006; Michalewicz, 1996; Yadollahi et al., 2008), in that none of the proposed methods in this subject has the capacity of modeling DFMS (Anderson and Franks, 2001; Ghedjati, 1999; Jia et al., 2002; Sheikhan and Shabani, 2009).

In this paper, DFMS-MPN algorithm based on Memetic algorithm is proposed for scheduling time optimization in DFSM. The proposed algorithm comprised two stages: In the first stage, the system was modeled by Timed Petri Net, after which it was modeled with Memetic algorithm for scheduling optimization, and the simulation of results was done with Petri net method. In the second stage, a new algorithm based on particle swarm optimization was proposed which is called DFMS-PPN.

A distributed flexible manufacturing system has different groups of jobs. Each job needs an allocated machine. A distributed flexible manufacturing system should allocate jobs to the best benefitted factory. The following are some limitations in DFMS:

1. All operations of a job should be done in one factory.
2. A new operation will be start after ending the previous one.
3. During running of an operation, all interruptions should

*Corresponding author. E-mail: hossein.shirgahi@gmail.com.

Table 1. Illustration of an example of distributed flexible manufacturing system.

Job/Operations	Machines	Executing time
1/1	1,2	7,4
2/1	2	3
3/1	2,3	4,6
1/2	1	2
2/2	2,3	4,6
3/2	2	2
4/2	2,3	2,5
1/3	1,2	3,4
2/3	2,3	4,6
1/4	2	3
2/4	3	5
3/4	2,3	3,7

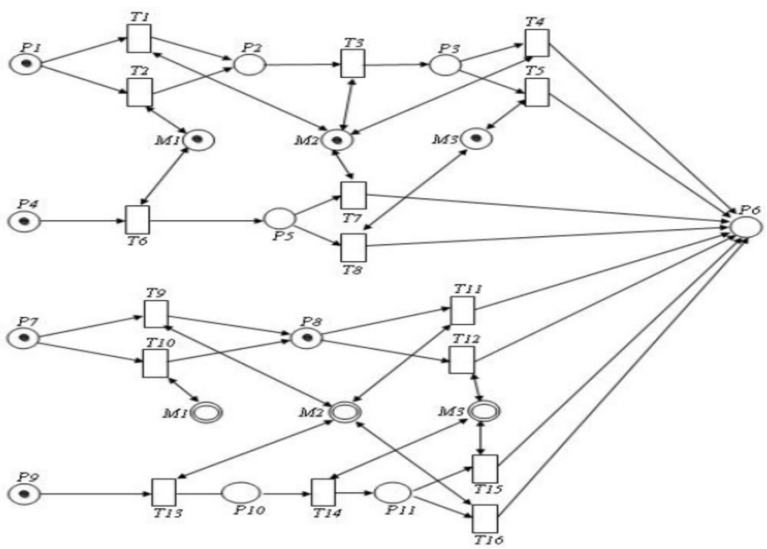

Figure 1. DFMS modeling presented in Table 1 by using Timed Petri net.

be avoided.
4. An operation should be run with one factory separately.
5. Each machine can run just one operation at a particular time.

Table 1 illustrates an example of the distributed flexible manufacturing system. There are two factories with three machines in each factory. For example, operation 1 of job 3 needs three seconds for running in machine 1, but needs four seconds for running in machine 2.

The Petri net is a mathematical and graphical modeling tool that includes Place, Transition, Arc and Token. The Petri net can be used for modeling, scheduling, simulation and systems performance evaluation. In addition to the given merits of using Petri net, the most important advantage of applying Petri net is its formality.

THE FIRST PROPOSED ALGORITHM (DFMS-MPN)

This paper proposed first a new algorithm by name DFMS-MPN based on Timed Petri net and Memetic algorithm for the distributed flexible manufacturing system scheduling, after which another method based on PSO was proposed.

Distributed flexible manufacturing system modeling

In the proposed algorithm, first a DFMS was modeled based on Timed Petri net. Figure 1 shows DFMS modeling which is presented in Table 1 by using Timed Petri net.

Scheduling based on Memetic algorithm

After modeling DFMS based on Timed Petri net, the best scheduling was produced with the modeled system. In the proposed algorithm, a chromosome shows sequence of total operations in factories.

221	112	322	412	223	112	413	323	113	412

Figure 2. The structure of the sample chromosome based on the sample system.

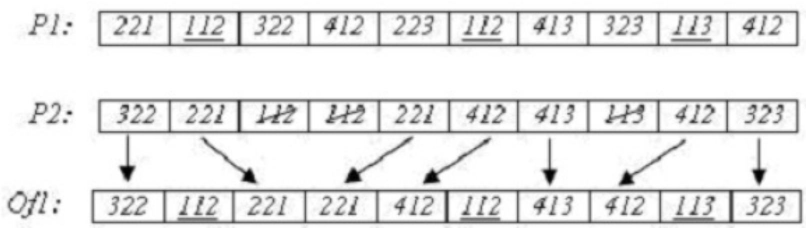

Figure 3. The operations of the crossover operator in job 1 randomly.

P1:	221	112	322	412	223	112	413	323	113	412
Ofl:	221	122	322	412	223	122	413	323	123	412

Figure 4. Operation of mutation in job 1 randomly.

In other words, a gene on chromosome describes sequence of fire transitions in Petri net. Figure 2 illustrates a sample chromosome for the sample system shown in Table 1. In this method, a chromosome includes many genes that have been shown by a number of chromosomes for the purpose of sequence job number, factory and allocated machine and repetition of sequence job number in the chromosome which shows the number of different operations of the job. For example, gene number three is 322. The first sequence job number is 3. Also, 3 is the sequence job number for the first time in the chromosome. It means that operation 1 of job 3 should run in machine 3 of factory 2.

Fitness function

The proposed algorithm uses Petri net for measuring the fitness of chromosomes. In other words, by using the Petri net in the distributed flexible manufacturing system, the sequence of the fire operation was simulated. Also, the length of scheduling was set as the fitness function of the chromosome.

Crossover and mutation

In the proposed algorithm, chromosomes were selected by roulette wheel. The mentioned crossover uses crossover based on job. At first, a job was selected randomly then all related genes of the job in the first parent were transferred to the first child in the same place. After that, all genes of the selected job in the second parent were deleted. Finally, the remaining jobs' genes were inserted into the first child sequentially and the mentioned stages were done for the second child.

One of the advantages of the mentioned operator is that the illegal chromosome is not generated, because it has not changed the arrangement of the job. Figure 3 shows the operations of the crossover operator by selecting job 1 on two parent chromosomes as a sample. The mutation operator has been used for making different routes meaning that after selecting a chromosome, one job was selected randomly and the factory number was changed. Figure 4 shows operations of mutation operator on job 1 of the sample parent chromosome.

Tabu Search

The aim of Tabu Search in the proposed algorithm is the allocation of a better machine to the scheduled operation. In other words, in Tabu Search, a better machine was searched by changing the machine number.

THE SECOND PROPOSED METHOD BASED ON PARTICLE SWARM OPTIMIZATION (DFMS-PPN)

Some social manners are portrayed in most kind of animals, while in some other kinds, there are superior members which lead others, for example lions, monkeys, deer and so on. Also there are some animals that live together without any leader, that is, each member can live and move by self organization in their environment (for example, birds, fishes and sheep). These kinds of animals do not have any knowledge about their environment but they can move and live together by exchange of information.

PSO method is a public method of optimization that can be used in problems whose answer is one point or surface in dimension space. In this space, some assumptions are defined and the initial speed is allocated to the particles and connection channels that have been allocated to the particles.

These particles move to the answer space, and then the results are calculated after each time slice with fitness function. After some time, the particles move to higher fitness basis in the same connection

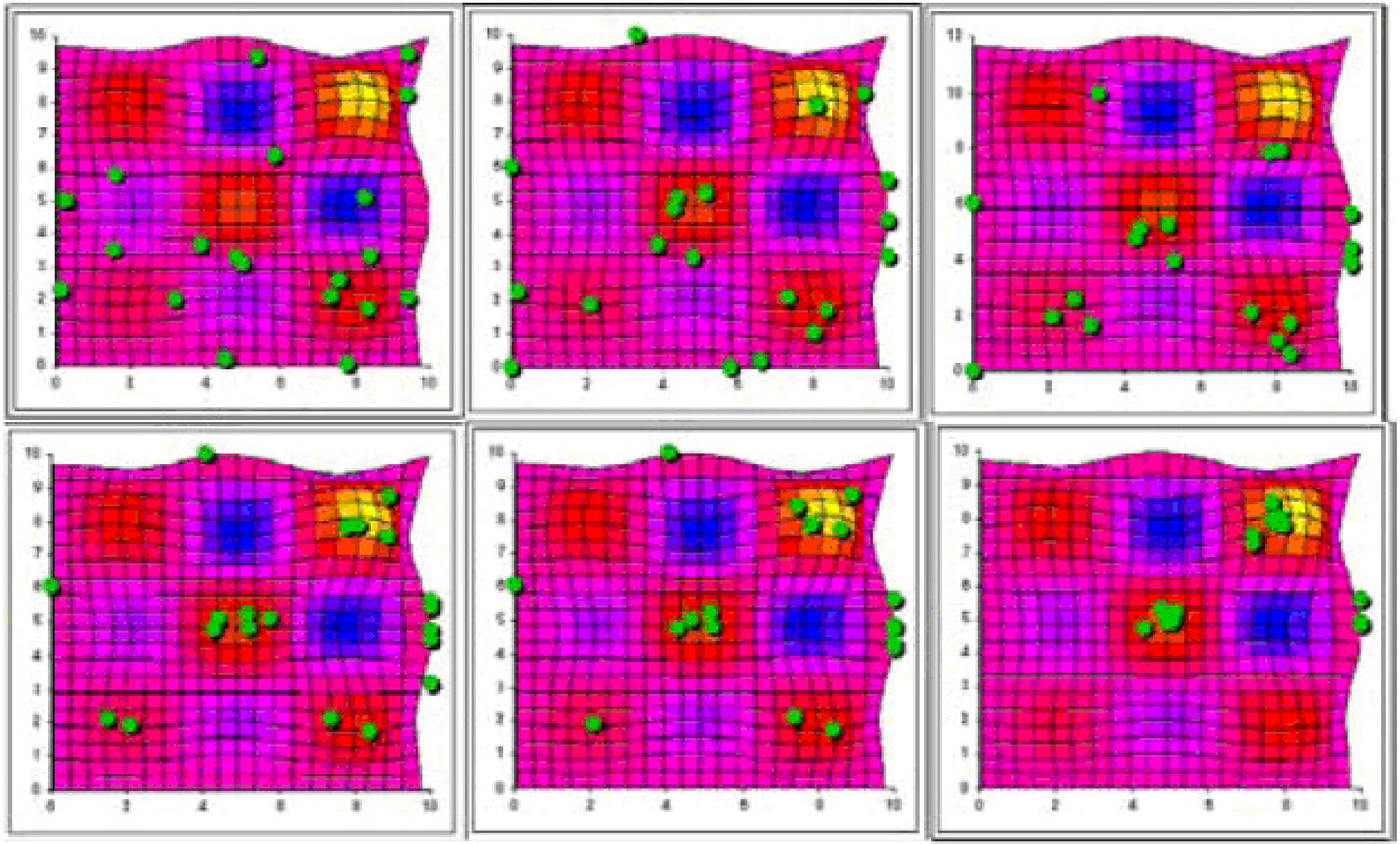

Figure 5. Samples of particles movement in search space

group. The big advantage of this method is flexibility for solving the local optimized problems with many numbers of crowded particles. Figure 5 shows samples of particles movement in search space. The top left of the figure shows the position of the particles at initializing in the two dimension space. Finally, the particles are converged by repetition as

Table 2. SPN-GA, TPN-MA, DFMS-MPN and DFMS-PPN results.

Problem	PN-GA			TPN-MA			DFMS-MPN			DFMS-PPN		
	Min.	Ave.	Std-Dev.	Min.	Ave.	Std-Dev	Min.	Ave.	Std-Dev.	Min.	Ave.	Std-Dev.
Set 1	46	621	510	54	573	566	47	551	537	48	567	532
Set 2	95	1642	1415	87	1601	1425	103	1498	1407	127	1518	1387
Set 3	153	2581	2445	169	2523	2388	147	2397	2089	131	2403	2192
Set 4	274	4062	3921	257	3902	3617	249	3501	3368	236	3596	3416
Set 5	427	5941	5617	412	5324	5201	353	5217	5106	367	5251	5095
Set 6	795	9523	8911	720	8562	8329	679	8388	8059	611	8424	8172

shown in the left part of Figure 5.

Each particle has a position with the particle coordinate in multi dimension search space. By moving the particle with time, the position of the particle changed. $x_i(t)$ defines the position of particle i in time t, while $v_i(t)$ is the speed of the particle in time i. The new position of the particle is calculated with Equation 1.

$$x_i(t+1)=x_i(t)+v_i(t+1)$$
$$x_i(t)\sim U(x_{\min},x_{\max}) \quad (1)$$

The position of the particle was evaluated by fitness function in search space. The particles have the ability to remember their best position during their life span; as such, v_i was named for the best position or experience of a particle (in some algorithms, the name was pbest). Also, the particles were aware of the best positions they meet in the past, and so the position was named $\hat{y}_i$ (in some algorithms, the name was gbest). Speed vector in the optimization shows the community's information and knowledge experience. However, a particle has two components for moving in search space. Cognitive component $v_i(t)-x_i(t)$ is the best solution that a particle obtains, while social component $\hat{y}_i(t)-x_i(t)$ is the best solution that can be recognized by the entire group. There are two main models for standard PSO algorithm and for calculating speed, based on cognitive component and social component. The difference of the two mentioned algorithms (gbest PSO and Lbest PSO) is the allocation of different vicinities. In this paper, standard PSO was used to solve FMS problem, and the results were discussed in the study's experimental results. Thus, the position of particle in this method is the same as the chromosome definition given previously.

EXPERIMENTAL RESULTS

The C#.Net 2008 language programming was used for testing the proposed algorithm and comparing it with TPN- MA and PN-GA algorithms. Six set of data were generated for testing the four mentioned algorithms. However, setX was used in a test to know if X is the number of data test. By increasing X, the system became greater. For increasing the accuracy of algorithms' comparisons, each algorithm was run for 10 times independently. The final result was the average of the results. For the fact that the initial population was 1000, the mutation rate was 0.15 and the reproduction rate was 0.8.

Table 2 shows comparison results of PN-GA, TPN-MA, DFMS-MPN and DFMS-PPN. It can be clearly seen that the standard deviation of the proposed algorithm is less than other algorithms. As a result, the proposed algorithm has more stability. Figure 6 shows results of DFMS-MPN, TPN-Ma, DFMS-PPN and PN-GA algorithms based on schedule time. It can be clearly seen that our proposed algorithm has more evaluation by a comparison with other algorithms, because of the use of Tabu Search in DFMS-MPN for local searching and the use of Petri net for simulation and fitness calculation.

Figure 7 shows a comparison of the algorithms based on the time needed for running. We can see that DFMS-MPN need more time in comparison with other algorithms, because DFMS-MPN uses Tabu Search for local searching and Petri net models for simulation and calculating chromosome fitness; but in DFMA-PPN method, the time of running is reasonable.

Conclusion

In this paper, we proposed two new algorithms (DFMS-MPN and DFMS-PPN) for DFMS, in which one of them was based on Timed Petri net and the other was based on Particle Swarm Optimization method. In the study's method, the distributed flexible manufacturing system was modeled by Timed Petri net, after which a scheduled task was generated by Memetic algorithm and particle swarm optimization method. The experimental result showed that the proposed method was a reasonable method when

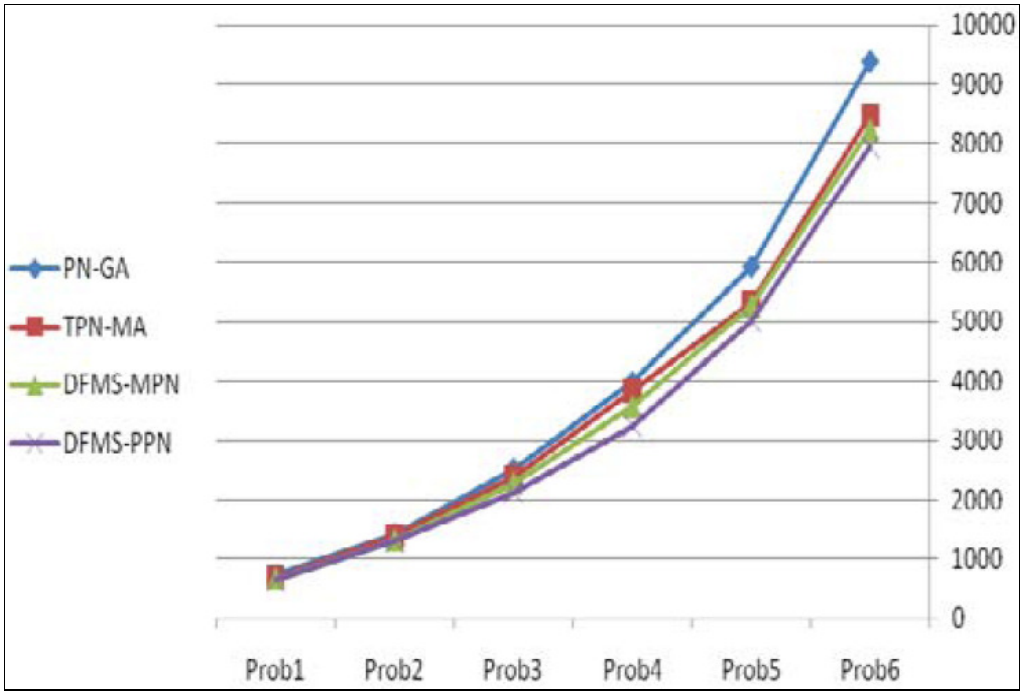

Figure 6. Comparison of DFMS-MPN, TPN-Ma, DFMS-PPN and PN-GA algorithms scheduling time.

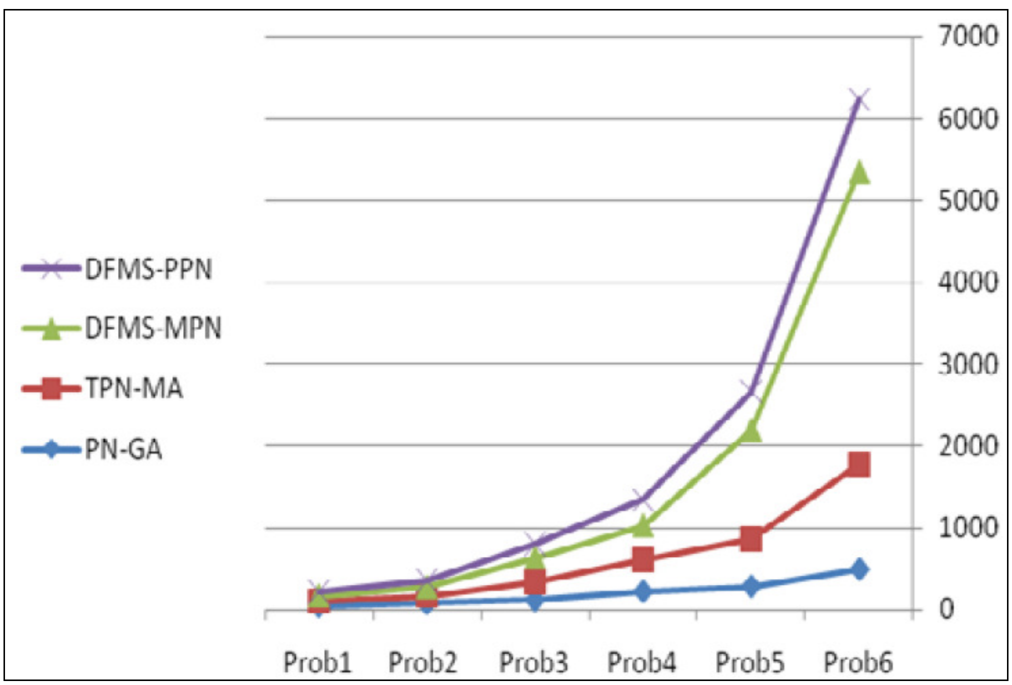

Figure 7. Comparison DFMS-MPN, TPN-Ma, DFMS-PPN and PN-GA algorithms running time.

compared with other algorithms, but one of the algorithms needed more running time because of the usage of Tabu Search for local searching and Petri net for simulation and chromosome fitness calculation.

REFERENCES

Anderson C, Franks NR (2001). Teams in Animal Societies. Behavioral Ecology Computation, 12(5): 534-540.

Caballero J, Mej.a G (2004). A Hybrid Approach that Combines Petri Net

Models And Genetic Algorithms for Scheduling of Flexible Manufacturing Systems, III Congreso Colombiano y I Conferencia Andina Internacional de Investigaci.n de Operaciones, Cartagena, Colombia. Electronic Proceedings, pp. 73-80.
Chan FTS, Chung SH, Chan PLY (2006). Application of genetic algorithms with dominant genes in distributed scheduling problem in FMS, Int. J. Prod. Res., 44(3): 23–43.
Ghedjati F (1999). Genetic algorithms for the job-shop scheduling problem with unrelated paralled constraints: heuristics mixing method machines and precedence, Comput. Ind. Eng., 73: 39–42.
Jia HZ, Fuh JYH, Nee AYC, Zheng YF (2002). Web-based multi-functional scheduling system for a distributed manufacturing environment, Concurrent Eng. Res. Appl., 10(1): 27-39.
Kumar R, Tiwari MK, Shankar R (2003). Scheduling of flexible manufacturing systems: an ant colony optimization approach, IMechE2003 (Part B), pp.1443–1453.
Mej.a G, Acevedo J (2006). Reactive Scheduling in FMS: an Integrated Approach based on Petri Nets, Genetic Algorithms and Simulation, Third International Conference on Production Research – Americas Region(ICPR-AM06).
Michalewicz Z (1996). Genetic algorithms+data structures ¼ evolution programs, NY: Springer, Berlin, Heidelberg, pp. 75-79.
Sheikhan M, Shabani AA (2009). Fast Neural Intrusion Detection System Based on Hidden Weight Optimization Algorithm and Feature Selection, World Appl. Sci. J. (Special Issue for Computer & IT), 7(5): 443-452.
Yadollahi M, Rahmani AM (2009). Solving distributed flexible manufacturing systems scheduling problems subject to maintenance: Memetic algorithms approach, The 9th IEEE Int. Conf. Computer Info. Technol. Xiamen, China. pp. 36-41.
Yadollahi M, Rahmani AM, Zamanifar K (2008). TPN-MA, A novel algorithm for scheduling flexible manufacturing systems with memtic algorithm and time petri net, 2nd joint congress on fuzzy and intelligent systems, Iran. pp. 132-137.