International Journal of Scientific & Engineering Research, Volume 4, Issue 9, September 2013
ISSN 2229-5518

Present a New Hybrid Algorithm Scheduling Flexible Manufacturing System Consideration Cost Maintenance

Abbas Bagherian Farahabadi, Ali Asghar Rahmani Hosseinabadi

Abstract—Flexible Manufacturing Systems (FMS), are generally included a group of machines that will do various tasks on different parts, an operation can be processed on different machines and all things are controlled by a computer system. This flexibility may increase the intensity of the solution space and schedule FMS will be complicated. Also, since in the real world, the production does not always correspond to a predetermined schedule and may be unexpected events occur that cause schedule outages are expected, concepts such as dynamic scheduling of manufacturing systems and considering the uncertainty in question has special importance. Scheduling problem of FMS is a difficult optimization problems and the timing of these systems is to reduce the time to do things. In recent years, many algorithms have been proposed to solve this problem, this algorithms will be based on evolutionary techniques – In this paper a new algorithm for "Petri net" gravitational force when combined with the name (GELS–PT) for scheduling FMS is presented that the issue will be kept in mind. Experimental results have shown that the proposed algorithm is about "51.91" percent of the algorithm "GADG" and "60.72" percent of the algorithm "PN–GA" has been improved.

Index Terms— FMS, GELS Algorithm, Petri Net, Scheduling

1 Introduction

FMS is one of problems of NP-hard optimization. The aim of scheduling of these systems is to maximize the system efficiency by finding an optimal planning for a better cooperation among various processes. Recent years have witnessed a wide range of research on modeling, scheduling, and evaluation of these systems. The main motivation lies in the fact that such systems require higher flexibility with respect to market change so that they can raise their competency. Many algorithms have been proposed to solve the above-mentioned problem. For example, Takahashi et al. [1] employed Tabu Search for scheduling of FMS. In [2] presented an algorithm for scheduling using Ant Colony optimization. In [3] proposed a genetic algorithm combined with Petri Net,i.e. PN-GA for scheduling FMS. In this algorithm, first manufacturing system is modeled with Petri Net, and then an optimized scheduling is determined using a Genetic Algorithm.

In [4] developed ETPN-GA, a GA based on Timed Petri Net for scheduling manufacturing systems which are suitable for reconfiguration aiming to minimize the production time and costs of reconfiguration. In [5] presented an innovative combinational algorithm to solve scheduling of job shop with parallel machines in order to minimize the production time. In this algorithm the problem is solved through combination of GA and ACO These algorithms do not consider maintenance. Chan et al. [6] proposed a GA to solve scheduling in distributed FMS considering maintenance. The major idea in their paper is presentation a GA with dominant genes. In [7] have used Memetic Algorithm to solve scheduling of distributed FMS which its advantages is considering maintenance factor and trade off between time and cost.

As mentioned earlier, the above mentioned algorithms do not consider maintenance and it decreases the system's reliability [8]. In this paper، an algorithm is linked with the Petri Net gravitational force when called "GELS–PT" for scheduling FMS is presented, taking advantage of this algorithm, the problem will be maintained. The second section describes the problem first and then the third part of the basic concepts are described in section 3 the proposed algorithm is and in section 4 and simulation results comparing the proposed algorithm with other existing algorithms shown in section 5 the conclusions have been made.

2 Problem Description

The aim of scheduling FMS is to present an appropriate scheduling for N jobs in M machines, as each job includes a number of operations that should be executed in order. Each operation requires one machine for execution. After some time each machine needs a time for maintenance and its duration depends on machine service's time (machine age) [8]. While maintenance the machine will be unavailable,

- *Ali Asghar Rahmani Hosseinabadi, Department of Computer, Islamic Azad University, Behshahr Branch, Iran, +989380297581,E-mail: A.R.Hosseinabadi@iaubs.ac.ir*
- *Abbas Bagherian Farahabadi, Department of Computer, Islamic Azad University, Behshahr Branch, Iran, +989111553700,E-mail: Abbas_Bagherianfarah@yahoo.com*

but after completing maintenance it will be available again. after each maintenance, the machine's age will be zero. If the maximum age of a machine can be A, and in case the machine age will be A, after completion of current execution, the machine turns to maintenance state. Figure 1 show the required time for machine maintenance with respect to machine age.

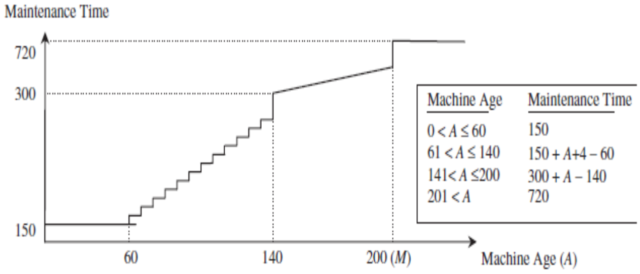

Fig.1. Maintenance time related to machine age [9]

Table 1 shows characteristics of a sample FMS with three machines and two jobs. In this system there are varying numbers of jobs operations. For example, it shows that second operation of first job in second machine needs four seconds, and that by the third machine requires seven seconds.

TABLE 1

CHARACTERISTICS OF A FLEXIBLE MANUFACTURING SYSTEM

Job 1 Operation	Machine	Process Time	Job2 Operation	Machine	Process Time
O_1	1	4	O_1	1	12
O_1	2	3	O_1	2	6
O_2	2	4	O_2	2	4
O_2	3	7	O_3	3	6
O_3	2	6	O_4	2	7
O_3	3	4	O_4	3	3
O_4	3	6	O_5	3	6

3 Basic concepts

In this section، a random search algorithm to mimic the gravitational force and the time Petri Net is described.

A. Gravitational Emulation Local Search (GELS)

In 1995, Voudouris and Tesang [10] offered the algorithm GLS for searching in a search space and solving the example NP-complete for the first time, and in 2004, Barry and Webster [11] offered the algorithm as a powerful algorithm and it was called GELS. This algorithm introduced randomization concept along with two of four primary parameters i.e. velocity and gravity in physics through swapping in terms of groups by using random number in the existing local search algorithm GELS in order to avoid local minima. GELS takes as its basis the natural principles of gravitational attraction. Gravity works in nature to cause objects to be pulled towards each other. The more massive the object, the more gravitational "tug" it exerts on other objects. Also, the closer two objects are to each other, the stronger the gravitational forces are between them. This means that a given object will be more strongly attracted to a larger, more massive object than to another object of lesser mass at a given distance, and it will also be more strongly attracted to an object close by than to another, more distant object having the same mass. In GELS the formula of newton gravitational force theorem between two objects, are involved in:

$$F = \frac{Gm_1m_2}{r^2} \qquad (1)$$

In which m_1 and m_2 are the mass of the first and second object respectively. G is equal to a constant gravitational force which is 6.672; R is the radius of the distance between two objects.

GELS also emulate these processes of nature for searching in a search space. The idea is to imagine the search space as being the universe and object in this universe are the possible solution for the search. Each of these solutions has a "mass" that is represented by its objective function value. The better the solution's objective function value, the higher its mass is. Locations within the search space that do not contain valid solutions are assigned a zero mass [12,14].

In this method, the possible solution in the search space has divided into some sets based on criteria that depend on the kind of problem and each of these sets is called a dimension of the problem solution and for each dimension of the problem solution a value entitled initial velocity has been intended, that it will be explained in continuation.

GELS computes the gravitational force between the solution or the objects in the search space by two methods. The first method which is a solution from the local neighborhood space is selected as a current solution and the gravitational force between these two solutions can be computed. The second method applies the formula to all solutions within the neighborhood and tracks the gravitational force between each of them and the current solution individually all solutions. In the movement through the search space, GELS acts in two modes, too. The first mode, allows movement only to solutions within the current local search neighborhood. Each of these movement modes can be used with each of the computation gravity forces and as a result those four models GELS are made.

GELS maintains a vector, the size of which is determined by the number of dimensions in a solution. This vector's values represent the relative "velocity" in each dimension. The algorithm begins by initializing the current solution, velocity vector and direction of movement. For each dimension, in the velocity vector, random integer between one and the maximum velocity is chosen, and this becomes the value of the element at that dimension. The initial solution can be made as current solution either with user or randomized. For each dimension in the initial velocity vector, concerning the initial velocity vector of the solution dimensions, a direction is selected for the movement that the direction is equal to the solution dimension which has the largest value in the initial velocity vector.

Algorithm includes a pointer object that can move through the search space and a mass which is intended for the pointer object is stable in the whole computations and this object refers to a solution with the largest mass. The algorithm will terminate when one of following two conditions occurs: either all of the elements in the initial velocity vector have gone to zero, or the maximum allowable number of iterations has been completed. In each algorithm iteration as the first method a candidate solution will be selected from the local neighborhood space of the current solution based on the direction of the current movement and the gravity force between the current solution and the candidate solution is computed and then the velocity vector concerning this force will be updated. For the next frequency, the velocity vector is checked and concerning it the movement direction can be chosen. Each iteration algorithm by the second method is completely as similar as the first method, but there is a little difference, as the gravity force and the initial velocity of the update action is computed for each of the candidate solutions instead of the gravity force computation and the update action of the velocity vector for only an obtained candidate solution from the current direction [15]. Newton's formula is used with the alteration that the two masses in the numerator of the equation are replaced by the value of the difference between the objective function value of the candidate solution and that of the current solution. The value of the gravitational force between the two solutions then becomes:

$$f = \frac{G(CU - CA)}{R^2} \qquad (2)$$

That, in which, *CU* and *CA* are the value of the current solution and the candidate solution. This formula is designed to be a positive value if the objective function value of the current solution is larger than that of the candidate solution and negative if the candidate value is larger. Then the value of this force, positive or negative, can be added to the velocity vector in direction of the current movement. If doing so makes the value exceed the maximum velocity parameter setting, it is set to the maximum. If the update makes the value to go negative, it is set to zero.

Some parameters which can be accessible are as follows:

- **Maximum Velocity:** the maximum value that any elements within the velocity vector can have been used to prevent velocities that become too large for being usable.
- **Radius:** it is a radius that can be used in the formula of the gravitational force computation.
- **Iteration:** the number of iterations of the algorithm that will be allowed to complete before it is automatically terminated [14].

B. Time Petri net

Petri net graphical and mathematical modeling as a tool that contains four elements of place, transition, are and is a token. The petri net can be modeled; scheduling‘ simulation and performance evaluation systems can be used. Petri net modeling a system in addition to the above advantages, we will need from the problem by mathematical formulas. Petri net is a kind of Petri net in which a transfer after the time specified, the token is fired.

4 The Proposed Algorithm

In this issue should be on appropriate timing for *N* work in *M* car done so that should be implemented in order to perform any operation that requires a car. And objectivity of each car after that time will hold up to the time dependent maintenance car service (car old) car is. To solve this problem, even with a small number and the machine will be trouble. The gravitational force algorithm when combined with a Petri net is used for scheduling FMS.

This algorithm is a two phase system with time Petri net models, and secondly by using the gravitational force of a scheduling algorithm optimized for the system model, will be produced. Petri net simulation model is then used to determine the extent of inappropriate timing.

A. Modeling with time Petri net

The first step, the Petri net modeling of FMS is the time. This work will need to overcome the problem of mathematical expression and the possibility to simulate and evaluate the system performance is expressed in Table 1 and figure 2 modeling prototype system by Petri net time will show.

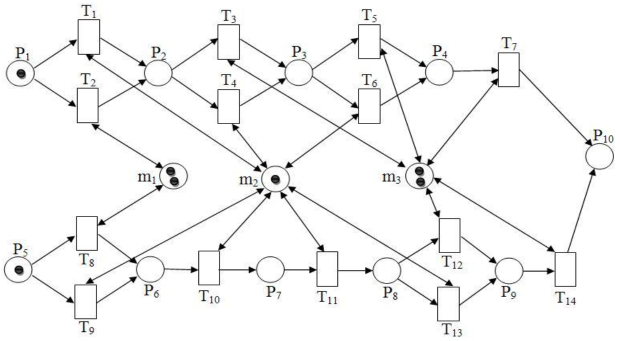

Fig.2. Modeling a sample FMS with Timed Petri Net

B. Scheduling algorithm with the gravitational force

In the second step, using a random search algorithm is a scheduling system to mimic the gravitational force.

C. View the original chromosomes produced

The number of genes in the chromosomes of the gravitational force can be raised using the algorithm works is equivalent to the total operations result in chromosome numbers obtained above from Left to right shows scheduled operations tasks. The structure of each gene as a "*Jms*" that the [1] has been achieved in this way display, parameter "*J*" number‹ the parameter "*m*" car number and the "s" maintenance mode is indicated, if the parameter "*s*" to mean that a car after this operation is to maintain in the value of this parameter will be zero otherwise. The number of repeats in a chromosome represents the number of operations it is produced. Figure 3 for a sample of chromosomes, Table 1 shows the FMS.

1120	1221	2120	1330	2220	2320	1431	2430	2530

Fig.3. Structure of a sample chromosome

Figure 3 gene on chromosome II in "1221", which means a second operation in the second car is doing well after operation, the machine goes to maintenance mode.

D. Fitness function

Using generated chromosome, the sequence of transition fires in Petri net is specified. Therefore, duration of scheduling (production time) would be determined through system simulation and it is termed as chromosome fitness function.

.

E. Definition of response

Response of the proposed method can provide the number of Jobs to machines that have a good time to do *N* work in *M* machine to be considered. Priority given to each machine, a neighbor of the response and the response in the desired response will be equal.

F. Define the neighborhood

Of the algorithm "GELS" unlike other search algorithms do not accidentally answer neighbor, but each has a different answer now is that each of them is based on a specific change, It called for moving to ward the neighbor and the neighbor's response that is obtained by this method is only based on the neighbor. The proposed method is to find a neighbor who used to work at any stage that is not run by machines assigned acquire and then we change the number of machines assigned to each dimension, resulting in a neighbor on the current car will be available again after storage.

G. Method of problem solving

This should be done in a proper schedule *N* work in *M* machine if any of this work consists of a number of operations and should there fore be implemented. And also to perform any operation that require a car.

Besides the car after work when the need for maintenance will keep the time machine is dependent on car servicing – so factors such as reducing work time and increase system efficiency. Of the algorithm "GELS" proposal is a randomly selected and assigned to a machine that will meet all the above is the issue of limitations and when to keep fighting, this will be done until the work completed. The problem with this condition must be defined in an appropriate reserve factor. Factor reservation the number of jobs are reserved for cars in the future. Total assignment of tasks to machine would-be a solution. This solution could be that the two matrices is equal to the number of jobs and machines are used to display an answer. Each row and column of the matrix, represents one of the tasks in groups and kept in each row and column represents the number of machines that run on them will work.

This can reduce the time and work on cars and appropriate timetable for doing *N* work in *M* car took. As the method "GELS" was described, this method with a random solution with experimental values of operating parameters can be adjusted, your work has started. And then executing "GELS" slash and when the algorithm was completed, the solution for each of tasks, machines that have booked their reservations with the invoice shows.

5 Simulation Results

Algorithm for the implementation of programming Language C# is used. Programs on the computer with a CPU 2.4 GHz Pentium–IV and memory card 1GB were implemented.

Proposed algorithm with two algorithms PN–GA [3] and GADG [6] has been compared. For comparison, five of the above algorithms are designed to test systems for small, medium and large cover. These data to test name "*Job X – M-N-O*".

The "*X*" number of test data, "*M*" number of cars, "*N*" number of things, "*O*" of the operation is done. For example, "Job 2-10-20-5" – shows the second test data set, with 10 cars and 5 for each work operation is shown in this test, 20 operations there are other words that indicate a moderate system. To obtain and view the results of the algorithms presented in the paper, each algorithm was run independently for 50 times. Table 2 shows the results "PN–GA and GADG, GELS–PT" comparison of three algorithms. This table shows the number "success" for the proposed algorithm is higher than the other two algorithms. Number "success" explains the 50 times run the algorithm several times is the best answer. So we can conclude the algorithm "GELS–PT" have a more stable than the other two algorithm are.

Due to the use of random search algorithms mimic the gravitational force as a local search technique.

TABLE 2

COMPARING OF THE THREE ALGORITHM THE RESULTS (GELS-PT, GADG, PN-GA)

Problem	**GELS-PT**			**GADG [6]**			**PN-GA [3]**		
	Time	Ave.	success	Time	Ave.	success	Time	Ave.	success
Job 1_10_10_5	2	556	45	8	634	19	15	762	13
Job 2_10_20_5	5	718	42	23	1531	22	39	1879	16
Job 3_10_30_5	9	1349	40	41	2295	21	78	2730	12
Job 4_20_30_10	16	1592	37	143	2871	18	212	3624	14
Job 5_20_30_20	26	2150	33	391	5906	14	1036	7211	7

Success: Number of times that algorithm arrives at the best answer in 50 times execution.
Ave: Average of production time in 50 times execution (production time equals the maximum time required for execution of jobs)
Time: Time required for algorithm execution

Figure 4 shows comparison of three algorithms in terms of "GELS–PT and PN–GA, GADG" the schedule. This fig shows the algorithm "GELS–PT" compared with other algorithms, often gets a better answer. Results of the algorithm "GELS–PT". An improvement of about "51.91" percent of the algorithm "GADG" and "60.72" percent of the algorithm "PN–GA" will show. Of this improved system with larger, more visible.

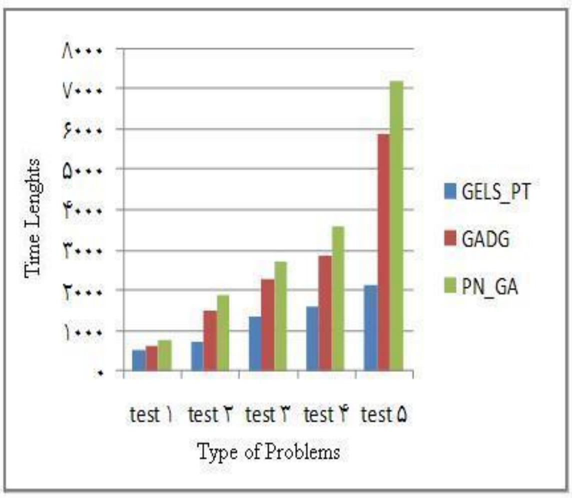

Fig.4. Comparison of the three algorithms (GELS-PT, PN-GA, GADG) in terms of the Length scheduling

International Journal of Scientific & Engineering Research, Volume 4, Issue 9, September 2013
ISSN 2229-5518

Figure 5 shows comparison of three algorithms in terms of time required to run "GELS–PT and PN–GA, GADG". This algorithm takes the forms of expression "GELS–PT" the other two algorithms it will need less time to run.

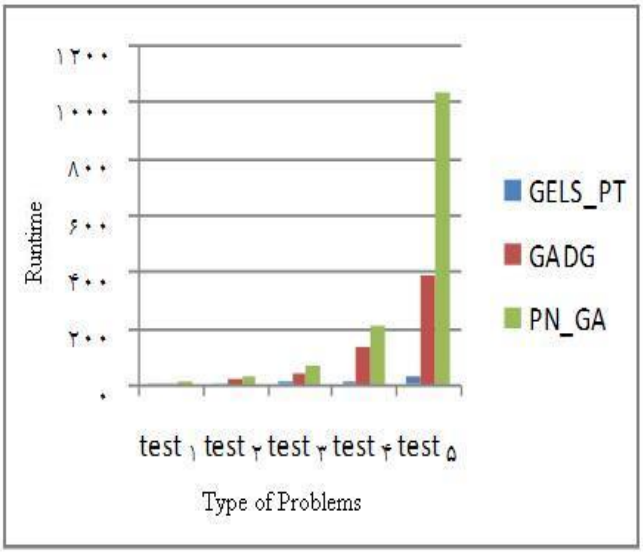

Fig.5. Comparison of the three algorithms (GELS-PT, PN-GA, GADG) in terms of runtime

6 Conclusion

In this paper a Petri net algorithm combined with the gravitational force when the name "GELS–PT" for scheduling FMS is presented. The advantages of this algorithm to reduce the time to do things, and consider the problem of system efficiency is maintained. Purpose of this algorithm will minimize the completion time of tasks. This algorithm has been compared with algorithm "PN–GA and GADG" performance. The results showed a recovery algorithm "GELS–PT" about "51.91" percent and "60.72" percent compared to the algorithm "DADG" algorithm "PN–GA" will – of this improvement is more apparent in large systems.

References

[1] Takahashi, Masakazu and Setsuya Kurahashi, "Tabu Search Algorithms for Multimodal and Multi-Objective Function Optimizations", International Journal of Computer Science and Network Security,Vol. 7, No. 10, 257–264, 2007.

[2] R. Kumar, M. K Tiwari and R. Shankar, "Scheduling of flexible manufacturing systems: an ant colony optimization approach", IMechE, 1443–1453, 2003.

[3] J. Caballero and G. Mejia, "A Hybrid Approach that Combines Petri Net Models And Genetic Algorithms for Scheduling of Flexible Manufacturing Systems", Conferencia Andina de Investigación de Operaciones, CCIO, 2004.

[4] Li. Aiping and X. Nan, "A Robust Scheduling for Reconfigurable Manufacturing System Using Petri Nets and Genetic Algorithm", Proceedings of the 6th World Congress on Intelligent Control and Automation, 7302-7306, 2006.

[5] A. Rossi and B. Elena, "A hybrid heuristic to solve the parallel machines job-shop scheduling problem", Advances in Engineering Software (Elsevier Ltd), 2008.

[6] F. Chan, S. H Chung, L. Y Chan, G. Finke and M K Tiwari, "Solving distributed FMS scheduling problems subject to maintenance: Genetic algorithms approach", Robotics and Computer-Integrated Manufacturing (Elsevier Ltd) 22, 493–504, 2006.

[7] M. Yadollahi and A. M Rahmani, "Solving distributed flexible manufacturing systems scheduling problems subject to maintenance: Memetic algorithms approach", International Conference on Computer and Information Technology, Xiamen, China, 36-41, 2009.

[8] F. Peres and D. Noyes, "Evaluation of a maintenance strategy by the analysis of the rate of repair", Qual Reliab Eng Int 19, 129–148, 2003.

[9] Felix T.S. Chan, S.H. Chung, L.Y. Chan, G. Finke, M.K. Tiwari, "Solving distributed FMS scheduling problems subject to maintenance: Genetic algorithms approach", Robotics and Computer-Integrated Manufacturing 493–504, 2006.

[10] C. Voudouris and E. Tsang, ″Guided Local Search," Department of Computer Science, University of Essex, UK, August, 1995.

[11] B. Webster, ″Solving Combinatorial Optimization Problems Using a New Algorithm Based on Gravitational Attraction", Ph.D., thesis, Melbourne, Florida Institute of Technology,2004.

[12] A. S. Rostami, H. M.Bernety and A. R. Hosseinabadi, "A Novel and Optimized Algorithm to Select Monitoring Sensors by GSA," ICCIA, 829–834, 2011.

[13] A. R. Hosseinabadi, M. Yazdanpanah and A. S. Rostami, ″New Search Algorithm for Solving Symmetric Traveling Salesman Problem Based on Gravity," World Applied Sciences Journal, Vol. 16 (10), 1387-139, 2012.

[14] A. R. Hosseinabadi, A. B. Farahabadi, M. S. Rostami, A. F. Lateran, "Presentation of a New and Beneficial Method Through Problem Solving Timing of Open Shop by Random Algorithm Gravitational Emulation Local Search", International Journal of Computer Science Issues, Vol. 10, Issue 1, 745-752, 2013.

[15] A. R. Hosseinabadi, M. R. Ghaleh, S. E. Hashemi, "Application of Modified Gravitational Search Algorithm to Solve the Problem of Teaching Hidden Markov Model", International Journal of Computer Science Issues, Vol. 10, Issue 3, 1-8, 2013.