JOURNAL OF CURRENT RESEARCH IN SCIENCE (ISSN 2322-5009)
CODEN (USA): JCRSDJ
2014, Vol. 2, No. 3, pp: 298-305
Available at www.jcrs010.com

REVIEW ARTICLE

UNIVERSITY COURSE TIMETABLING USING ADAPTIVE FUZZY PETRI NETS

Nehmatollah Khatami, Homayoun Motameni*, Majid Aboutalebi
Sari Branch, Islamic Azad University

ABSTRACT: A Petri net is an abstract formal model of the behavior of a system and information flow. The properties, concepts, and techniques of Petri net are so as to present it as a simple and strong method for describing and analyzing information flow and systems control. Fuzzy Petri Net (FPN) is an appropriate powerful model to emulate knowledge base systems using fuzzy rules. Yet, FPN model does not own learning capability in fuzzy systems. Recently, artificial neural nets have been used to add this capability to FPN. The parameters of fuzzy rules in FPN can be learned by adding the back propagation algorithms of neural nets to FPN. Petri nets with learning ability can be exploited in defining dynamic knowledge base and extrapolating in expert systems. To do so, mostly a generalization of fuzzy Petri net is suggested. This is known as adaptive fuzzy Petri nets. Adaptive fuzzy Petri nets are appropriate for modeling expert systems like university course timetabling system. In this study, university course timetabling is modeled using adaptive fuzzy Petri nets. MATLAB software is suitable for implementing adaptive fuzzy Petri nets. This software is applied to simulate the results of the study.

KEYWORDS: Petri nets, fuzzy systems, university course timetabling, artificial neural nets, adaptive fuzzy Petri nets.

INTRODUCTION

The uncertainty and ambiguity of decision makings are among the issues of today world. Hence, inaccurate and fuzzy data is unavoidable. On the other hand, analyzing such data requires special analytic system known as fuzzy set theory. Expert systems are those systems simulating an expert's thinking and cognition.

The Petri net model using the three elements of the model:

Place: Indicates changing the system state

Transition: The events that cause changing the system state are Indicates.

Arc: Indicates the relationship between states.

Also during the implementation of petri net are used to describe the current status of the token. In fact, the token are in place.

Petri nets with concepts like Concurrent, Asynchronous, Distributed, Parallel, Non-deterministic and Stochastic are used. Petri nets became further completed in time. And, more applied and newer concepts were added to the concepts used in them. Hybrid Petri nets are widely used in computer systems, industry, robotics, knowledge base, process control, and other applications in engineering. To model and analyze, FPN is used in smart systems to process fuzzy knowledge. But, FPNs lack learning ability. They become problematic when they are used in smart systems like expert and automatic systems. Regarding the self-adaptability and self-leaning capability of neural nets, self-assessment ability is expanded in Petri net (Lpeterson, 1977).

In the second section of this article, we will present Petri nets, fuzzy sets, and their hybrid model. And, in the third section, neural nets are presented. Learning capability in neural nets is also added to fuzzy Petri nets. Hybrid Petri nets models will be discussed. Then, in the fourth section, algorithms proposed for university course timetabling will be examined. We will also mention the limits considered for our proposed methods. In the fifth section, we will discuss university course timetabling using adaptive Petri nets. Then, we will present the results of simulation. In the end, we will discuss the results of the study and provide our suggestions.

FUZZY PETRI NETS

Fuzzy logic is a powerful tool for solving problems related to hybrid and complex systems. Normally, these systems are rarely understood. And, human being has the faintest information about them. Generally, fuzzy logic is introduced as opposed to classical logic. Its new view toward issues and breaking zero-one logic require the production and development of this logic (Baghmisheh, 2010). Fuzzy systems are knowledge-based or rule-based systems.

Corresponding Author: Homayoun Motameni, Sari Branch, Islamic Azad University. **E-mail:** motameni@iausari.ac.ir

A knowledge base is the heart of a fuzzy system. Hence, the beginning point of a fuzzy system construction will be to obtain a set of if-Then fuzzy rules of experts' knowledge or of a domain under study (Barzegar *et al.*, 2011).

Petri net usually adapts to classical logic. Yet, we normally deal with complex systems described with a degree of uncertainty. Accordingly, if we aim to apply Petri net to model systems like expert systems, we have to be able to display uncertainty based on vague and inaccurate terms in Petri model. As a result, designers of Petri net and Artificial Intelligence Association designed a variety of fuzzy Petri nets (Li *et al.*, 2002).

A fuzzy mark is a generalization of a sign in standard Petri net. In the standard Petri net, the value of the mark belongs to {0,1} set. Yet, in fuzzy sign, our value can be assigned to [0,1] interval. A stronger idea is that a verbal term (such as little, medium, much) is assigned to a mark. A membership function is also attributed to each of these verbal terms. The membership function determines the extent of membership from a place with the if-True of assumption. A fuzzy place has a statement or attribute of that place. A mark in that place is set by that attribute and a number indicating the extent of membership of that mark in the respective statement of that place. In this method, we obtain a fuzzy design or conclusion. For instance, weather is a little warm.

A fuzzy transfer, for example, corresponds with an if-Then fuzzy production rule. It is realized by the correspondent values if-True of fuzzy extrapolation algorithms (Motameni *et al.*, 2008).

A fuzzy arch determines a value required by a mark. If a mark is not accurately fitted to the requirement, an approximate value will exist between the mark and required value. And, if the interval is larger than a predetermined maximum value, its transfer will be fired.

PETRI NETS WITH LEARNING CAPABILITY

Neural nets partially act like Petri nets. We can model neural nets using Petri nets.

3.1. Neural Nets

Neural nets are established based on their structure. The structure includes one or more layers. Each layer owns a number of neurons. Each neuron is connected to others by weighed synapses. Each neuron has input(s) and output(s). Each neuron calculates the number of input(s) based on which it produces output(s).

Fuzzy systems obtained from learning algorithms and use neural nets theory to determine parameters (fuzzy sets and rules) are called neural-fuzzy systems. In many cases, accurately obtaining data may be difficult. Namely, to properly present the knowledge of the real world, fuzzy rules must be applied to present knowledge (Li *et al.*, 2002; Li and Rosano, 2000).

The main characteristic of a neural net is the net's capability to learn from its environment and improvement of efficiency via learning process. A neural net is capable of understanding its surrounding better by means of learning processes and by implementing regulations required on synoptic weights. On the other hand, after each repetition of the learning process, the knowledge of neural net about its environment increases.

Apparently, it can be realized that there is no single algorithm for designing neural nets. We have a set of tools provided for a wide spectrum of learning algorithms. Each of the sets has its own specific advantages. The only difference between learning algorithms is in the method of presenting the relations of organizing synoptic weights or net parameters. In Figure (1), simple neural net with three neurons is illustrated (Menhaj, 2002).

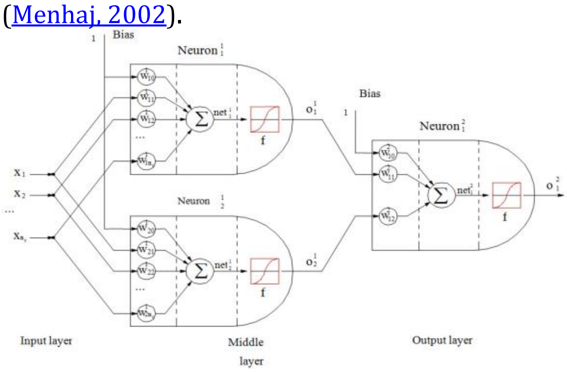

Figure 1: neural network with three neurons and Two activate layer

Therefore according the Figure (1) equations of input - output of the network with three neurons are as follows:

$$neuron_1^1 = \begin{cases} net_1^1 = \sum_{i=1}^{n_0} w_{1i}^1 . x_i + x_0 \\ o_1^1 = f\left(net_1^1\right) \end{cases} \quad (1)$$

$$neuron_1^2 = \begin{cases} net_1^2 = \sum_{i=1}^{2} w_{1i}^2 . o_i^1 + o_0^1 \\ o_1^2 = f\left(net_1^2\right) \end{cases} \quad (2)$$

$$neuron_2^1 = \begin{cases} net_2^1 = \sum_{i=1}^{n_0} w_{2i}^1 . x_i + x_0 \\ o_2^1 = f\left(net_2^1\right) \end{cases} \quad (3)$$

In these equations o_0^1, x_0 respectively, the first and second bias input neurons.

3.2. Different Models of Petri Nets with learning Capability

In his method, Fissore and Sokeipirm, (2011) presents the simulation and analysis of energy consumption by plant of recovering propane from natural gas using Self-learning Fuzzy Petri Nets (SFPN). Based on the examinations, this method has led to the optimization of heat reduction to %25.

Kasirolvalad, (2009) has applied Adaptive Fuzzy Petri Net (AFPN) to control lathes. In the provided model, variables including machine vibration, shear force, spindle speed, and progress rate are used machining operation for determining surface inequality. Then, based on results, a model is presented with respect to a knowledge-based expert system.

In his study, Abed, (2011) used Fuzzy Neural Petri Nets (FNPN) as a controller for temperature control system. This is because these systems have characteristics such as non-linearity and high variance time. Besides, it is difficult to overcome these factors and achieve appropriate results using ordinary controllers. Simulation results indicate that this smart control method owns considerable capability against jobbery as well as good dynamic performance and speed.

Fuzzy Neural Petri Nets (FNPN) is used for fuzzy knowledge base and fuzzy logic. It is obtained from the integration of Neural Petri Net with fuzzy logic. NPN^2= (P,T,Z,A,C,M) is a progressive net composed of two types of knots as places and transfers. NPN receives signals from input places and sends them to output places. NPN can be applied to classification and is mainly used in regulating separate synopses weight.

Using NPN and FNPN, computations have become more feasible. Their required memory is also reduced (Ahson, 2002).

$FNPN = (P, T, Z, C, G, V, \alpha, \beta)$

$P = \{p_1, p_2, ... p_k\}$ is a finite set of places

$T = \{t_1, t_2, ..., t_j\}$ is a finite set of transitions

$Z \subseteq (P \times T) \cup (T \times P)$ is a finite set of arcs

C = {X,Y,G} is a finite set of propositions

Where $X = \{x_1, x_2, ..., x_n\}$, $Y = \{y_1, y_2, ..., y_m\}$, $G = \{g_1, g_2, ..., g_q\}$

$V = \{v_1, v_2, ..., v_l\}$ is a finite set of fuzzy truth values

α = P $\rightarrow$ C is a mapping from places to propositions

β = T $\rightarrow$ V is an association function from transitions to tuth values

Adaptive Fuzzy Petri Net (AFPN) model has both characteristics – the learning ability of neural nets and fuzzy Petri net modeling. In this model, logic algorithm and weight learning algorithm are expanded. The generalized AFPN of fuzzy Petri nets is for being used in expert systems (Li *et al.*, 2002; Li and Rosano, 2000).

AFPN = (P,T,D,I,O,α,β,TH,W)

Self-learning Fuzzy Petri Nets (SFPN) save the results existing in knowledge base. When previously followed results are developed or a new decision is made, the model can read changes and apply them to effectively improve the system (Meng and Yan, 2009).

SFPN = (P,T,D,I,O,M,β,λ,W)

Other similar models can be implied, as follow:

(AFHOPN) Adaptive Fuzzy Higher Order Petri Nets (Li and Rosano, 2000)
(AFCPN) Adaptive Fuzzy Colored Petri Nets (Chen *et al.*, 2003)
(OOFPN) Object Oriented Fuzzy Petri Nets (Li and Yu, 2001)

Ordinary Petri nets are integrated with fuzzy logic. And, then, learning ability is added to fuzzy Petri nets.

UNIVERSITY COURSE TIMETABLING

One of the main educational issues in universities is how to design weekly schedule. This is an example of the general issue; that is, scheduling. It is defined in a variety of forms and based on different environments. Designing a weekly schedule of an educational unit includes the scheduling of various courses or sessions, classroom, teacher in the schedule of the educational semester. It is done with respect to the matter that different groups of students use the same teachers, classrooms, and facilities.

As mentioned, the issue of developing schedules is among NP-Complete matters. Numerous solutions and studies are presented on this subject. In general, since 1960s, many studies have been carried out on scheduling courses. Hertz proposed the use of taboo search method (Tabu search) to solve course scheduling problem in two steps (TATI/TAG). He emphasized that the trend is suitable for solving large scale problems like developing schedule of courses and exams. Ross presented the use of modern knowledge in applied developmental algorithm for resolving different schedules (Abdullah and Turabieh, 2008). Cambazard *et* al., (2012) applied local search and limited programming techniques for solving the problems after registering courses in schedule. An elementary solution is presented by Gunawan *et al.*, (2012) in the Largrangian relaxation model. And, he developed and improved it by simulating the inflammation of the subject and providing a solution in postgraduate schedule. Genetic Algorithm (GA) was used as opposed to university schedule which was prepared from different constraints and requirements (Burke *et al.*, 2007). Ant Colony Optimization (ACO) algorithm was used by Nothegger for solving problems after registering courses in schedule (Nothegger *et al.*, 2012). Besides, Shiau, (2012) solved courses scheduling problems using hybrid particle optimization. Tassopoulos and Beligiannis, (2012) used hybrid particle swarm optimization for developing schedule capable of being used at different high schools in Greece. A community behavioral similarity plan as Hybrid Harmony Search (HHS) was presented by Al-Betar *et al.*, (2012) for solving university scheduling problems. To balance the exploitation and recognition of search, hybrid harmony algorithm integrates hybrid particle swarm optimization and hill climbing algorithm. A genetic hybrid algorithm and Tabu search approach were also suggested by Jot and Yung for solving scheduling problems after registering courses (Jat and Yang, 2010).

4.1. Constraints of University Course Timetabling

In an issue regarding the satisfaction of constraints, these constraints are divided into two groups: hard and soft. Strong constraints are those conditions satisfaction of which is necessary for having correct and applied schedules. Weak constraints are extra conditions satisfaction of which improves the efficiency of schedules and their desirability (Kohshori *et al.*, 2008).

Hard constraints of university course timetabling:

H1: Courses shall not be administered in a time interval.

H2: Two courses shall not be administered in the same classroom.

H3: Teachers shall not simultaneously lecture in different classes. They shall teach in one class in a time interval.

Soft constraints of university course timetabling:

SC1: Class capacity shall not be less than the capacity of the students enrolling in that course.

SC2: If a series of courses administered are named by 1 to 24, then default will be set as a series of courses for each teacher.

UNIVERSITY COURSE TIMETABLING METHOD USING ADAPTIVE FUZZY PETRI NETS

To improve university course timetabling by Petri nets, the variables of scheduling system are required to be recognized. And, then, they have to be adapted to Petri nets. In this study, adaptive fuzzy Petri nets were used to schedule courses. They are so effective in solving various problems in expert systems.

Fuzzy Petri Net on its own can describe Weighed Fuzzy Productive Rules (WFPR) well. Yet, it is not able to adjust its weights to update input data. Namely, it is not capable of learning. Accordingly, we use AFPN model. The model is a ninth one. AFPN = (P, T, D, I, O, α, β, TH, W) where P stands for the set of places, T for the set of transfers, D for the set of terms, and I for input-output function which defines a mapping of transfers to places. α is the relation function which assigns a real value between 0 and 1 to each place. β is a mapping between terms and a place label for each knot. The share of P, T, and D is blank. And, P and D capacity is the same. TH is a function which assigns a threshold value λ from 0 to 1 to transfer t. W is the community of a set of input and output weights. It assigns weights to each arch of the net. In this problem, three types of fuzzy production rule are applied.

First, we write the rules in WFPR form. And, then, we transform them into AFPN rules. The mapping between these two is as follow: Each transfer ends in one of the three rules. And, each place turns into a (primary or end) term. Fuzzy inference algorithm and back propagation algorithm are highly effective for cases where we do not have AFPN weights. After a learning process, we can gain a good mapping from input to output.

Final scheduling will satisfy all hard constraints reassuring the validity of the problem. And, it will remain valid for different educational periods, unless a new constraint is implemented in system. It is also considered to be a suitable solution since it regards higher coefficient for soft constraints. The presented model is a general model. It can be used for any data group. And, it can be adapted to certain formulations by changing weights in target function. However, response should be checked by personnel before being applied in real situation so as to reassure the satisfaction of all constraints.

5.1. Target Function

Target function is applied to formulate priorities for various periods of day. Since constraints will always reassure the validity of resulted schedule, target function can be selected for indicating any constraint having majority. It is done by weighing variables related to different periods. Regarding the matter that it is a minimizing problem, a large weight will increase target function. And, as a result, it cannot be useful for solution. Hence, weights control result. And, the method of selecting them is important.

5.2. Fuzzy Rules and the Structure of Fuzzy Nets

1. Inputs Set

The random selection includes indices such as presented courses, capacity of classes, name of halls where courses are administered, selecting time intervals and teachers' name.

2. Membership Functions Layer

In membership functions layer, we will have fuzzy step for inputs set and their mapping in [0,1] interval by Gaussian functions. They will be defined as follow:

$$\mu ji = e^{-0.5(\frac{fj-cij}{sji})^2} \qquad i,j = 1,2,3,4,5$$

1. First, the set of inputs are mapped in a local matrix x with dimensions 50*4. The elements of the matrix will include courses selections, time intervals, hall number, and teacher's name.
2. If hard and soft conditions were true, fj=0. Otherwise, fj=1.
3. Variable Sji indicates the transformation of knot from +1 to -1 or vice versa. If there is error in adjusting conditions, the value will fluctuate between 1 and -1.
4. Accordingly, the centers of these functions will fluctuate between 0 and 1 based on the validity and rightness of hard and soft conditions.

3. Petri Net Layer

This layer is for producing the values of tokens used for firing knots based on the comparison between rules. These values will be valued as follow:

$$tij = \begin{cases} 1 \ if \ \mu ji \geq dth \\ 0 \ if \ \mu ji \leq dth \end{cases} \tag{8}$$

Where, tij stands for transit rules and dth for the dynamic threshold of net. They are defined as follow:

$$dth = \frac{\alpha e^{-\beta E}}{1 + e^{-\beta E}} \tag{9}$$

α and β are positive values that can be chosen randomly and E same amount of error and here the cost function is defined as the sum of squared errors in which case there is a direct correlation between and only if the mismatch conditions (error) is the amount of cost function. In a sense that represents the error rate is. In this example purpose of the adaptive weighted, minimizing error or the same cost function is.

It is clear that the threshold value by increasing the error is further reduced.

4. Rule Layer

The output of each node is multiplied by its inputs (In other words, the values of membership functions in transition state or Fire) That as below are displayed.

$$\emptyset j = \begin{cases} \prod_i^n \mu ji & , if \ tij = 1 \\ 0 & , if \ \ tij = 0 \end{cases} \tag{10}$$

That $\emptyset j$ Is the j-th Rule output layer node.

5. The output Layer

In this phase error value choices made in previous stages based on the weight finding values are expressed $\emptyset j$.

$$costNew = \sum_i^{ni} wji \times \emptyset j \tag{11}$$

5.3. Learning algorithm University Course Timetabling

In this part of the fuzzy rules, the initial conditions of membership functions and the initial values of the parameters are selected.

Learning parameters of the optimization objective function value will be set in this part that the example values are:

1. Learn rate weights ethaw=0.1;
2. Membership functions of the centers of learning rate ethac=0.1;

Weights training algorithm is as follows:

1- Knowing the certainty factors for all output locations (right side of rules).
2- Only one layer of weights can be trained.
3- If more than one transition will fire for the rules the Third Kind need to Know which transition makes the tokens to output place.
4- These conditions are very strict because this information may not be available in real expert systems. We've simplified the conditions for the more general cases. Main idea is that if certainty factor to all places of destination are. Layer weights can be updated via backpropagation algorithm.

The weights are updated in this phase will be expressed as follows:

Wt(1): Weight depends on the setting of non-overlapping courses.

Wt(2): Weight depends on the setting of appropriate timeframe for each course.

Wt(3): Weight depends on the setting of choosing the right venues.

Wt(4): Weight depends on the setting of professor suitable for the desired course.
Wt(5): Weight depends on the setting of processing time and effort searching the input matrices.

Now monitored parameter error back propagation algorithm, the network would like to update.

Cost function to consider sum of the squares error and we have:

$$E = \frac{1}{2}\sum costNew - tIni \qquad (12)$$

In Equation 12 Variable *costIni* The previous value *costNew* The efforts of previous stage will be.
Rules update *w* (weights) and *cij* (centers of membership functions) are defined as follows:
A. The calculation of the cost function changes than the weights

$$\frac{\partial \mathrm{E}}{\partial \mathrm{wj}} = (costNew - costIni\ )\emptyset j \qquad (13)$$

B. The calculation of the cost function changes than the main centers

$$\frac{\partial \mathrm{E}}{\partial \mathrm{cij}} = \begin{cases} (costNew - costIni\ )\emptyset j\ \frac{(fj - cij)}{sji^2} & if\ tij = 1 \\ 0 & if\ tij = 0 \end{cases} \qquad (14)$$

fj s in equation (14) represents adapt the conditions problem are that If not adapt. Amount equal to one And will zero in some adaptation state.

C. Update values

$$wji(k+1) = wji(k) - ethaw \times \frac{\partial \mathrm{E}}{\partial \mathrm{wj}} \qquad (15)$$

$$cij(k+1) = cij(k) - ethac \times \frac{\partial \mathrm{E}}{\partial \mathrm{cij}} \qquad (16)$$

5.4. Simulation Results
Final output Table 1 would be. error choices will depend on the initial values of weights.

Table 1: Results of the simulation output

Timeslot	Course	Teachers	Venue	Capacity	# of Students
1	AGE 304	Motahari	NLT	250	90
2	FSE 508	Saeedi	BEL	85	75
3	CVE 500	Khatami	MICOM	90	60
4	EEE 524	Saeedi	MICOM	90	75
5	MEE 208	Saeedi	LH	100	80
6	CVE 526	Khateri	MICOM	90	45
7	CSE 306	Tahmasebi	NLT	250	90
8	EEE 200	Oladali	NLT	250	65
9	EEE 316	Saeedi	1200LT	1200	90
10	CHE 310	Esmaeili	1200LT	1200	90
11	MEE 232	Saeedi	1200LT	1200	850
12	MEE 208	Jafari	MICOM	90	80
13	MEE 232	Tahmasebi	1200LT	1200	850
14	CHE 210	Sadat Hoseini	MKO	750	80
15	CSE 306	Esmaeili	NBEL	95	90
16	CHE 206	Taheri	LH	100	80
17	MEE 208	Taghavi	NBEL	95	80
18	EEE 524	Oladali	LH	100	75
19	FSE 508	Taheri	MICOM	90	75
20	FSE 302	Tahmasebi	1200LT	1200	90
21	EEE 200	Oladali	NBEL	95	65
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.

Simulation time (seconds) =31.1222
Violations
Number of course clashing =0; Courses given lower venue capacity =0; Lab courses given wrong venues =5

Computing time Will depend on to the determined effort in program and also initial conditions That in This program is considered amount of effort 300 and computing time also as you can see In Table 1 is 31.1222. According to the above results all the hard conditions compliance has been quite and In output table gives a good time. In Figure 2 you can see the values of the weights That converges To the constant desire will eventually.

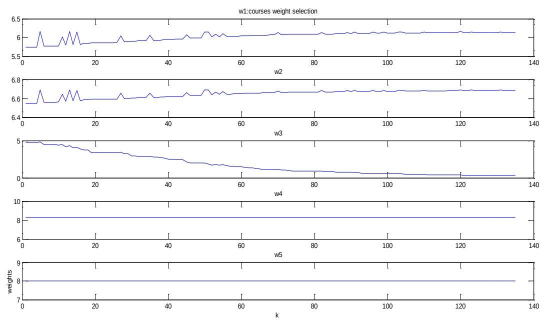

Figure 2: Values of The final weights

DISCUSSION

In simulation results, it is observed that five labs are wrongly considered. And, this error is the absence of suitable condition for distinguishing between different labs.

In weights correction diagram, weights reach fixed value after 120 steps. And, these long steps are random due to primary weights and data.

In addition, weight related to setting suitable teacher for respective course and the weight related to process time as well as the number of search attempts in input matrices have relatively a fixed value. They do not have significant effect on net training.

CONCLUSION

As implied in previous chapters, the issue of university course timetabling is a large scale issue with many variables. The amount of its computations increases as the number of variables increases. The method used in this thesis for solving the issue of university course timetabling is placed in the subset of meta-heuristic methods. In this method, we tended to solve the complicated issue of university course timetabling via establishing an adaptive fuzzy Petri net with learning ability. Namely, the adaptive fuzzy Petri net predicts next optimization mode via information gained from the previous modes of the system and reactions of dynamic environment. It also updates the current status of the system and changes the probability of events occurrence. It in fact activates the events based on their occurrence probability which has higher processing speed and further adaptability as compared to other meta-heuristic algorithms (including ant colony, genetic algorithm, and hybrid algorithm, etc). Yet, we face acceptable error in this algorithm as the time of computations decreases. As seen in Figure 5-1, it reaches a stable amount after 120 computing attempts.

The constraints of the presented system:

1- To solve the issue of university course timetabling using this method, we need primary desirable response to be able to measure the error and implement in the net.

2- Although membership functions value is between 0 and 1, they cannot cover a completely constant interval. This is because conditions are rational and valid.

3- We have to search for the information of the problem. And, it is an unavoidable time-consuming matter.

SUGGESTIONS

In some cases, we encountered with labs suitable arrangement error in this algorithm. Hence, this problem can be partially alleviated using multi-layer adaptive Petri nets to improve performance and reduce error in problem-solving process. That is, the final computations of the first Petri net is implemented as input for reprocessing and reforming the previous net's error.

We can use algorithm with non-random method rather than random search in the indices of data matrix and finding suitable neighbor. Namely, we can search for suitable neighbors for arranging schedule based on the existing priorities of the problem. On the other hand, we can present a constructive algorithm to form efficient responses with respect to the objectives of the problem (reducing the random space). This is because when the problem is too big, the efficiency of algorithm will decrease. Hence, using constructive algorithms which minimize problem-solving space by means of special strategy and ideology and intelligently proceed in scheduling for this certain problem will be highly effective.

We can include conditions - like the attendance of the students as well as the probable cost for using facilities of university - in the optimization problem to optimize the problem with a more suitable algorithm.

Regarding the dispersion of solutions and the issue of entropy, the pressure of selecting teachers for courses decreases based on their capacity and time. And, if we employ this concept during execution, the selection of teachers beyond their time and working capacity will be avoided. Yet, due to longer duration of algorithm execution, the presentation of a strategy on how to use it or presenting a method on how to prevent from the convergence of courses can be a part of future works.

And, in the end, by implementing mean-field algorithm using Petri net in problem solving as a suitable typical algorithm, it is also possible to reduce the volume of probable computations and processing time. That is, rather than computing knots mode probabilities, we can use their mean.

REFERENCES

Abdullah S, Turabieh H. Generating university course timetable using genetic algorithm and local search. In Proceedings of the 3rd International Conference on Convergence and Hybrid Information Technology 2008; pp: 254–260.

Abed A. A Robust Neural Fuzzy Petri wet Controller for a Temperature Control System. Procedia Computer Science 2011; pp:881-890.

Ahson SI. Petri net models of fuzzy neural networks. Proceedings in IEEE Transactions on Systems, Man and Cybernetics 2002; 35(6):926–932.

Al-Betar MA, Khader AT, Zaman M. University Course Timetabling using a hybrid harmony search metaheuristic algorithm. IEEE Trans Syst Man Cybern Rev 2012;42: 664–681.

Baghmisheh M. Simulation, modeling and optimization of chemical processes, 2010.

Barzegar S, Davoudpour M, Meybodi MR, Sadeghian A, Tirandazian M. Formalized learning automata with adaptive fuzzy coloured Petri net; an application specific to managing traffic signals. Scientia Iranica, Transactions D: Computer Science & Engineering and Electrical Engineering 2011;18(3): 554–565.

Burke EK, Elliman DG, Weare RF. Application of genetic algorithm for solving university timetabling problems: A case study of Thai universities. In Proceedings of the 7th WSEAS International Conference on Simulation, Modelling and Optimization 2007; pp:128–133.

Cambazard H, Hebrard E, O'Sullivan B, Papadopoulos A. Local search and constraint programming for the post enrolment-based course timetabling problem, 2012.

Chen X, Jin D, Zhijan LI. Fuzzy Petri nets for rule-based pattern classification. Proceedings in IEEE International Conference on Communications 2003.

Fissore D, Sokeipirm D. Simulation and energy consumption analysis of a Propane Plus recovery plant form natural gas. Fuel Processing Technology 2011.

Gunawan A, Ng KM, Poh KL. A hybridized Lagrangian relaxation and simulated annealing method for the course timetabling problem. Comput Oper Res 2012;39: 3074–3088.

Jat S, Yang S. A hybrid genetic algorithm and tabu search approach for post enrolment course timetabling. J Sched 2010;14;617–637.

Kasirolvalad Z. Anintelligent Madular modeling approach for quality control of CNC machines production using adaptive fuzzy Petri nets. Control, Automation, Roboties and Vision Conference 2009.

Kohshori M, Saniee Abadeh M, Sajedi H. A Fuzzy Genetic Algorithm with Local Search for University Course Timetabling, 2008.

Li X, Rosano FL. Adaptive fuzzy Petri nets for dynamic knowledge representation and inference. Expert Syst Applicat 2000;19(3): 235–241.

Li X, Yu W, Lara-Rosano F. Dynamic knowledge inference and learning under adaptive fuzzy Petri net framework. Proceedings in IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 2002;30(4):442–450.

Li X, Yu W. Object Oriented Fuzzy Petri Net for Complex Knowledge System Modeling. Proceedings of IEEE 2001.

Lpeterson J. Petri Nets. Computing Surveys 1977;9(3):24.

Meng XG, Yan HS. Inventory Planning Based on Knowledgeable Base and Self-learning Fuzzy Petri Nets, 2009.

Menhaj MB. Foundations of Neural Networks, 2002.

Motameni H, Movaghar A, Daneshfar I, Nemat Zadehand H, Bakhshi J. Mapping to Convert Activity Diagram in Fuzzy UML to Fuzzy Petri Net. World Applied Sciences Journal 2008;3(3):514-521.

Nothegger C, Mayer A, Chwatal A, Raidl G. Solving the post enrolment course timetabling problem by ant colony optimization. Ann Oper Res 2012;194:325–339.

Shiau DF. A hybrid particle swarm optimization for a university course scheduling problem with flexible preferences. Expert Syst Appl 2011;38:235–248.

Tassopoulos IX, Beligiannis GN. A hybrid particle swarm optimization based algorithm for high school timetabling problems. Appl Soft Comput 2012;12:3472–3489.