Check for updates

Parameter Optimization Strategy of Fuzzy Petri Net Utilizing Hybrid GA-SFLA Algorithm

Wei Jiang, Kai-Qing Zhou(✉), and Li-Ping Mo

College of Information Science and Engineering, Jishou University, Jishou 416000, Hunan, China
kqzhou@jsu.edu.cn

Abstract. Fuzzy Petri net (FPN) is a powerful tool to model and analyze the knowledge-based systems (KBSs) or expert systems (ESs). The accuracy of the reasoning result is a bottleneck to hinder the further development of FPN because of lacking self-learning capability. To overcome this issue, a hybrid GA-SFLA algorithm is proposed in this paper to improve the precision of each parameter of a given FPN model. The proposed algorithm combines the advantages both of GA and SFLA and includes three phases, which are generating chromosome by encoding the multi-dimensional solution which reflects all initial frogs, gaining a better individual as well as seeking the optimal solution by executing the local search and global search operations of SFLA. Finally, an FPN model is used to test the feasibility of the proposed algorithm. Simulation results reveal that all parameters of the given FPN model have the higher precision by implementing the GA-SFLA than that of implementing GA and SFLA, respectively.

Keywords: Parameter optimization · Fuzzy Petri net · Genetic Algorithm (GA) · Shuffled Frog-Leaping Algorithm (SFLA)

1 Introduction

Knowledge-based systems (KBSs) or expert systems (ESs), are a form of computerized artificial intelligence programming to capture and employ knowledge for settling complex problems, such as fault diagnosis or inference [1–3]. However, the uncertainties of objective rooted in people's information and knowledge are widely existed in the real world. Hence, it is required that the ESs need to reflect these uncertainties and fuzzy information in the knowledge representation and modeling processing [4–6]. The last few decades have witnessed a series of new methods for representing knowledge and automatic reasoning implementation, such as fuzzy production rule (FPR) [7], fuzzy Petri net (FPN) [8], Semantic Web [9] and frame-based representation [10], etc.

FPN is kind of high-level Petri nets (HLPNs) based on the backward extension principle [11]. Due to the graphical description capability and the systemic mathematical analysis mechanism, FPN can accurately depict the uncertainty and is commonly used in the modeling, analyzing, and reasoning for KBSs and ESs [7, 12, 13].

H. Song et al. (Eds.): SIMUtools 2019, LNICST 295, pp. 416–426, 2019.
https://doi.org/10.1007/978-3-030-32216-8_40

Nowadays, fruitful reasoning algorithms using FPN were borne and employed into different industrial areas for specific functions, such as fault diagnosis, path recognized, traffic schedule, process monitor, and so on [14, 15].

There are various successful FPN and its industrial applications. However, another bottleneck of FPN-how to obtain the more accurate final reasoning result of the goal output place-is still in the initial phase because the existing FPN formalism lacks of self-learning ability to improve the accuracy of relevant parameters value. Shen et al. developed two kinds of machine learning PN (MLPN) models to enhance the self-learning of the Petri net (PN) by supervised and unsupervised learning algorithms based on artificial neural network (ANN) [16]. Similarly, Tsang et al. proposed a learning strategy of a kind of 14-tuple FPN by using ANN. However, in the training process, training of thresholds of the FPN was neglected because the authors assumed all transitions of FPN can be enabled and fired [17]. Wang et al. employed an efficient genetic particle swarm optimization (GPSO) learning algorithm to execute self-learning function for the parameters of FPN. But the proposed GPSO learning algorithm is not suitable for some more complex and large-scale FPN models [18]. Above three literatures, it reveals that it is a feasible thinking to enhance the self-learning of FPN model and to practice the accuracy of each kind of parameters by using soft computing techniques.

Based on the similar thinking, a hybrid algorithm, namely GA-SFLA approach, is presented in this manuscript at first by combining the advantages of GA and SFLA. Then, the proposed hybrid algorithm is used to execute the training process for improving the accuracy of each type of parameters of FPN. The simulation results indicate that the FPN parameters which are optimized by GA-SFLA own better precision than that of which are optimized by SFLA and GA, respectively.

Remain parts are organized as follows. Section 2 gives the related concepts of FPN and FPR. Section 3 illustrates the framework and implementation steps of the GA-SFLA algorithm in details after analyzing GA and SFLA briefly. Section 4 shows the experimental results of parameters' optimization by performing GA-SFLA, GA and SFLA algorithms one-by-one on the same FPN case. Section 5 recalls and summarizes the entire manuscript.

2 Fuzzy Petri Net and Fuzzy Production Rule

FPN and FPR are two major formalisms which have been applied to fulfil the KBS requirements. This section introduces the basic concepts both of FPN and FPR. Then, the corresponding FPN model of different types of FPR is generated, respectively.

2.1 Fuzzy Petri Net

FPN generally defined as the following 8-tuple formalism.

$$FPN \quad \sum = (P, T, I, O, M, \mu, W, CF), \quad \text{where}$$

- $P = \{p_1, p_2, \cdots, p_n\}$ represents a finite set of places, where n represents the number of places in the rule;
- $T = \{t_1, t_2, \cdots, t_m\}$ represents a finite set of transitions, where m represents the number of transitions in the rule;
- $I(O)$ is the input (output) function, i.e., the mapping relation between places and transitions;
- $M = (m_1, m_2, \cdots, m_n)^T$ indicates the identity of places;
- w_i indicates the weight of places p_i, i.e., the support degree for the rule establishment by preconditions p_i;
- CF_j indicates the credibility, i.e., the true extent of the conclusion after transitions t_j fired;
- $\mu : \mu \to (0, 1]$, μ_i is the threshold of transitions t_j.

2.2 Fuzzy Production Rule

FPR is a commonly used to represent the uncertainties in expert systems [19–21]. General FPRs are formalized and described as follows.

$$if \quad D(\lambda) \quad then \quad Q \quad (CF, \mu, w), \quad \text{where}$$

- D is a limited set of preconditions, $D = \{D_1, D_2, \cdots D_n\}$;
- Q is a limited set of conclusions, $Q = \{Q_1, Q_2, \cdots Q_m\}$;
- λ is the true extent of each precondition, $\lambda \in [0, 1]$;
- CF is the credibility of the rule; $CF \in (0, 1]$ is the credibility of the conclusion obtained after the rule is executed;
- μ is the threshold of the rule, $\mu \in (0, 1]$;
- w is the weight of each precondition, $w \in (0, 1]$.

2.3 Correspondence Between FPN and FPR

After comparing with these formalisms, the correspondence between an FPR and FPN could be listed in Table 1

Table 1. The corresponding relationship between an FPR and FPN

FPR	FPN
FPRs	FPN model
FPR	Transition
Precondition and Conclusion	Place
Range of application of rule	Extension of transition
Weight of rule (w)	Input weight from place to transition (w)
True extent of each precondition (λ)	Value of Token ($M(p_i)$)
Threshold of rule (μ)	Threshold of transition (μ)
Credibility of the rule (CF)	Credibility from transition to place (CF)

FPRs can be divided into three main types, which are 'simple', 'or', and 'and' rules according to different relationship among conditions.

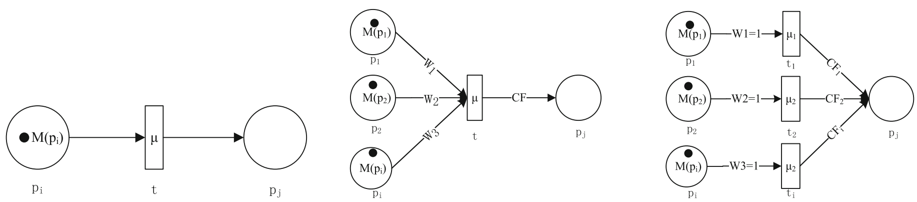
Type 1: Simple Rule
if $D(\lambda)$ *then* Q $(w = 1, \mu, CF)$

Type 2: And Rule

$$if \quad D_1(\lambda_1) \quad and \quad D_2(\lambda_2) \quad and \quad \cdots \quad and \quad D_n(\lambda_n) \quad then \quad Q \quad (\sum_{i=1}^{n} w_i = 1, \mu, CF)$$

Type 3: Or Rule
if $D_1(\lambda_1)$ *or* $D_2(\lambda_2)$ *or* $\cdots$ *or* $D_n(\lambda_n)$ *then* $Q(w_i = 1, \mu_i, CF_i)$

The corresponding FPN models of three types of FPR are illustrated in Fig. 1.



a. FPN model of Simple rule b. FPN model of 'And' rule c. FPN model of 'Or' rule

Fig. 1. The corresponding FPN model for each type of FPR

3 Hybrid GA-SFLA Algorithm

GA and SFLP are two common powerful evolutionary optimization algorithms to handle various complex engineering problems. In this section, brief introductions of GA and SFLP algorithms are given at first. Next, a hybrid algorithm based on the advantages both GA and SFLA is demonstrated in details.

3.1 Genetic Algorithm (GA)

GA is one of the most popular optimization algorithms based on stochastic search mechanism. Three basic operators-selection, crossover and mutation-are used to present a population of solutions in the implementation process of GA. In the initial phase, an initial population is created by a set of random solutions. A new population will be generated from the previous population by using three basic operators repeatedly till the termination criteria is reached [22]. The main advantages of GA could be summarized into three points. First, fit solutions could be found in a very less time. Next, a wide range of solutions could be evaluated based on the random mutation operator. Finally, it is easy to realize the coding operation for each solution [23–25].

3.2 Shuffled Frog-Leaping Algorithm (SFLA)

SFLA is a kind of optimization algorithms which is inspired by analyzing the behavior of frogs located in swamps to seek optimum location of food [26]. SFLA owns two different search abilities, local search as well as global search, to ensure obtain the optimum solution for complex problems [27]. Compared with other intelligent computing techniques, SFLA can gain optimal solution by the better performance of the global search because SFLA integrates the advantages both genetic from memetic algorithm (MA) and social behavior from particle swarm optimization (PSO) [28, 29].

3.3 Hybrid GA-SFLP Algorithm

In this manuscript, a hybrid GA-SFLP algorithm is proposed in this article to improve the self-learning capability of FPN by combining the advantages of GA and SFLA.

The GA-SLFP algorithm could be classified into three phases.

Function of the first phase is to generate each chromosome in the initial population by encoding the multi-dimensional solution which reflects all initial frogs.

The second phase is to gain a better individual by implementing the main algorithm frame of GA based on the obtained in the population under a give a fixed number.

The third phase is to the global optimal solution by implementing the local search and global search operations of SFLA for the better individuals got from phase 2.

The entire flowchart of the proposed GA-SFLA algorithm as shown in Fig. 2.

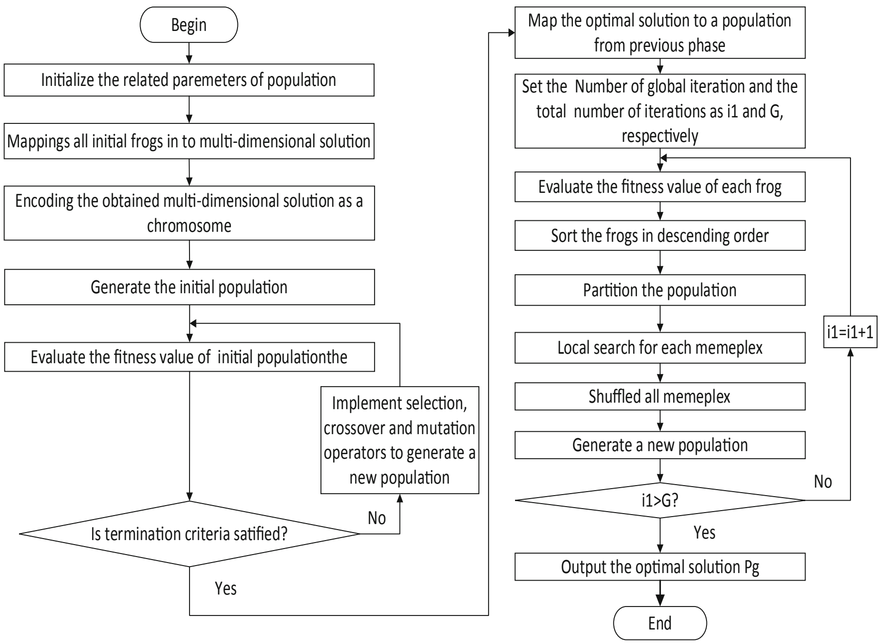


Fig. 2. Flowchart of the GA-SFLA algorithm

4 Experiment and Analysis

In this section, an FPN model is selected to reveal the feasible of the proposed GA-SFLA algorithm. Meanwhile, GA, SFLA and GA-SFLA algorithms are executed to optimize the different types parameters of the same FPN model.

4.1 Experiment Design

In general, FPN model, there are only three types of parameters. Hence, GA, SFLA and GA-SFLA algorithms are employed to optimize the three types of parameters (weight, threshold, credibility) one-by-one based on following principles (Take the credibility optimization as a case) in this experiment. In the initialization phase, the individual solutions are generated based on the range $CF_i \in (0,1](i = 1, \cdots, 5)$ randomly. The maximum and the minimum value of the individual could be set as $[1,1,1,1,1]$ and $[CF_1, CF_2, CF_3, CF_4, CF_5]$ (listed in Table 2).

- Calculate the fitness of each solution and implement the corresponding algorithm.
- Output the gained optimal individual solution.

FPN Model Selection Criteria

In this experiment, a simple KBS with 4 FPRs is selected to generate the corresponding FPN model. These four FPRs include three types of FPRs, which are simple rule, ‘or’ rule, and ‘and’ rule. Meanwhile, the meaning of each place is neglected because the goal of this experiment is to discuss the parameter optimization issue of the FPN model. The fours FPRs are listed below.

$$\begin{array}{llllllll} R1 & if & d_1 & or & d_2 & then & d_3 & (\mu_1, CF_1, \mu_3, CF_3) \\ R2 & if & d_1 & then & d_2 & (\mu_2, CF_2) & & \\ R3 & if & d_3 & and & d_4 & and & d_5 & then \quad d_6 \quad (w_1,\ w_2,\ w_3,\ \mu_4,\ CF_5) \\ R4 & if & d_3 & then & d_7 & (w_4,\ w_5,\ \mu_5,\ CF_5) & & \end{array}$$

The corresponding FPN model of above FPRs is generated as shown in Fig. 3.

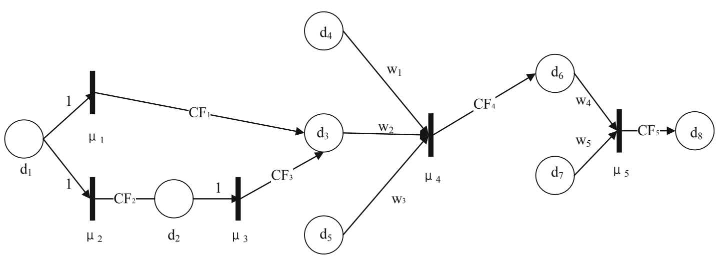


Fig. 3. The corresponding FPN model of 4-FPR KBS

Experiment Parameters' Setting

In this experiment, the expected values of three types of parameter, which are given by expert, are shown in Table 2.

Table 2. The expected values of three types of parameters

Parameter														
Weight (w)					Threshold (μ)					Credibility (CF)				
w_1	w_2	w_3	w_4	w_5	μ_1	μ_2	μ_3	μ_4	μ_5	CF_1	CF_2	CF_3	CF_4	CF_5
0.2	0.3	0.4	0.5	0.6	0.7	0.9	0.6	0.8	0.7	0.3	0.4	0.2	0.5	0.4

- For the entire GA-SFLA algorithm, the population size = 50 and the max number of iterations G = 300
- Other parameters are assigned based on the classical GA and SFLA algorithms.

4.2 Experimental Results and Analysis

Each algorithm is implemented five times. The final experimental results of each algorithm are listed in Tables 3, 4 and 5, respectively.

Table 3. Five times' experimental results by implementing GA

Parameter		1st result	2nd result	3rd result	4th result	5th result	Means
Weight	w_1	0.2548	0.2256	0.2323	0.2907	0.2656	0.25380
	w_2	0.3240	0.3219	0.3362	0.3205	0.3170	0.32392
	w_3	0.4511	0.4158	0.4146	0.4022	0.4386	0.42446
	w_4	0.5067	0.5133	0.5119	0.5033	0.5028	0.50760
	w_5	0.6106	0.6245	0.6100	0.6176	0.6040	0.61334
Threshold	μ_1	0.7404	0.7238	0.7148	0.7200	0.7149	0.72278
	μ_2	0.9216	0.9025	0.9230	0.9182	0.9115	0.91428
	μ_3	0.6123	0.6248	0.6172	0.6088	0.6031	0.61324
	μ_4	0.8085	0.8124	0.8344	0.8115	0.8191	0.81718
	μ_5	0.7240	0.7206	0.7057	0.7237	0.7201	0.71882
Credibility	CF_1	0.3049	0.3198	0.3246	0.3214	0.3343	0.32100
	CF_2	0.4196	0.4104	0.4216	0.4140	0.4092	0.41496
	CF_3	0.2010	0.2059	0.2079	0.2262	0.2220	0.21260
	CF_4	0.5344	0.5037	0.5205	0.5044	0.5207	0.51674
	CF_5	0.4481	0.4812	0.4124	0.4443	0.4176	0.44066

Table 4. Five times' experimental result by implementing SFLA

Parameter		1st result	2nd result	3rd result	4th result	5th result	Means
Weight	w_1	0.4450	0.3352	0.5907	0.3804	0.3521	0.42068
	w_2	0.4275	0.3451	0.4698	0.4881	0.5611	0.45832
	w_3	0.4899	0.5374	0.4750	0.6868	0.5384	0.54550
	w_4	0.6052	0.5488	0.5514	0.6200	0.6226	0.58960
	w_5	0.8161	0.7158	0.6641	0.6750	0.6579	0.70578
Threshold	μ_1	0.7188	0.8276	0.7296	0.7335	0.8319	0.76828
	μ_2	0.9044	0.9999	0.9388	0.9550	0.9488	0.94938
	μ_3	0.7188	0.6019	0.7004	0.6886	0.6399	0.66992
	μ_4	0.8826	0.8693	0.8225	0.8659	0.8881	0.86568
	μ_5	0.7752	0.7270	0.7493	0.7415	0.7536	0.74932
Credibility	CF_1	0.4707	0.3001	0.5079	0.5033	0.3567	0.42774
	CF_2	0.6056	0.4885	0.4183	0.5583	0.7120	0.55654
	CF_3	0.3525	0.2019	0.4060	0.3753	0.3105	0.32924
	CF_4	0.6617	0.5920	0.5505	0.6853	0.5058	0.59906
	CF_5	0.5748	0.6260	0.5055	0.5408	0.5572	0.56086

Table 5. Five times' experimental result by implementing GA-SFLA

Parameter		1st result	2nd result	3rd result	4th result	5th result	Means
Weight	w_1	0.2297	0.2203	0.2145	0.2177	0.2221	0.22086
	w_2	0.3218	0.3017	0.3062	0.3245	0.3089	0.31262
	w_3	0.4043	0.4156	0.4192	0.4114	0.4232	0.41474
	w_4	0.5003	0.5062	0.5033	0.5158	0.5052	0.50616
	w_5	0.6056	0.6114	0.6047	0.6005	0.6035	0.60414
Threshold	μ_1	0.7025	0.7029	0.7038	0.7134	0.7022	0.70496
	μ_2	0.9033	0.9123	0.9083	0.9040	0.9199	0.90956
	μ_3	0.6056	0.6135	0.6140	0.6083	0.6157	0.61142
	μ_4	0.8030	0.8053	0.8078	0.8033	0.8033	0.80454
	μ_5	0.7023	0.7049	0.7060	0.7020	0.7078	0.70460
Credibility	CF_1	0.3097	0.3021	0.3007	0.3002	0.3196	0.30646
	CF_2	0.4042	0.4074	0.4219	0.4007	0.4017	0.40764
	CF_3	0.2100	0.2290	0.2159	0.2068	0.2042	0.21318
	CF_4	0.5124	0.5086	0.5025	0.5071	0.5020	0.50652
	CF_5	0.4137	0.4089	0.4003	0.4025	0.4150	0.40826

Table 6 lists the expected value of the parameters and the related means of simulation results by implementing GA, SFLA and GA-SFLA algorithms, respectively.

Table 6. Five times' experimental result by implementing SFLA

Parameter		Expected value	Means of each parameter of simulation results		
		1st result	GA	SFLA	GA-SFLA
Weight	w_1	0.2	0.25380	0.42068	0.22086
	w_2	0.3	0.32392	0.45832	0.31262
	w_3	0.4	0.42446	0.54550	0.41474
	w_4	0.5	0.50760	0.58960	0.50616
	w_5	0.6	0.61334	0.70578	0.60414
Threshold	μ_1	0.7	0.72278	0.76828	0.70496
	μ_2	0.9	0.91428	0.94938	0.90956
	μ_3	0.6	0.61324	0.66992	0.61142
	μ_4	0.8	0.81718	0.86568	0.80454
	μ_5	0.7	0.71882	0.74932	0.70460
Credibility	CF_1	0.3	0.32100	0.42774	0.30646
	CF_2	0.4	0.41496	0.55654	0.40764
	CF_3	0.2	0.21260	0.32924	0.21318
	CF_4	0.5	0.51674	0.59906	0.50652
	CF_5	0.4	0.44066	0.56086	0.40826

According to Table 6, the obtained means of each parameter of the FPN by executing GA-SFLA algorithm are much better than that of GA and SFLA. Take a case as CF5, the expected value given by expert is 0.4, gained value by executing GA, SFLA and GA-SFLA is 0.44066, 0.56086 and 0.40826 based on 300 iterations. Hence, compared with GA and SFLA, the simulation results own higher precision by implementing GA-SFLA. It is further indicated that the FPN owns a stronger self-learning capability by using the GA-SFLA algorithm.

5 Conclusion

Focusing on the self-learning issue of FPN, a hybrid GA-SFLA algorithm has been presented in this paper to improve the precision of each parameter of the given FPN model. The proposed algorithm includes three steps: each chromosome in the initial population is generated by encoding the multi-dimensional solution which reflects all initial frogs at first. Then, the classical GA is used to gain a better individual. Finally, the local search and global search operations of SFLA are executed to obtain the optimal solution. A case study was used to illustrate advantages of the proposed algorithm by comparing the simulation results based on different algorithms. The results show that the FPN owns a stronger self-learning capability by using the GA-SFLA algorithm.

Acknowledgement. This work is supported by the National Natural Science Foundation of China (Nos. 61741205, 61462029).

References

1. Paredes-Frigolett, H., Gomes, L.F.A.M.: A novel method for rule extraction in a knowledge-based innovation tutoring system. Knowl.-Based Syst. **92**, 183–199 (2016)
2. Nasiri, S., Zenkert, J., Fathi, M.: Improving CBR adaptation for recommendation of associated references in a knowledge-based learning assistant system. Neurocomputing **250**, 5–17 (2017)
3. Merone, M., Soda, P., Sansone, M., Sansone, C.: ECG databases for biometric systems: a systematic review. Expert Syst. Appl. **67**, 189–202 (2017)
4. Yusup, N., Zain, A.M., Hashim, S.Z.M.: Evolutionary techniques in optimizing machining parameters: review and recent applications (2007–2011). Expert Syst. Appl. **39**(10), 9909–9927 (2012)
5. Zain, A.M., Haron, H., Sharif, S.: Application of GA to optimize cutting conditions for minimizing surface roughness in end milling machining process. Expert Syst. Appl. **37**(6), 4650–4659 (2010)
6. Adnan, M.M., Sarkheyli, A., Zain, A.M., Haron, H.: Fuzzy logic for modeling machining process: a review. Artif. Intell. Rev. **43**(3), 345–379 (2013)
7. Zhou, K.Q., Mo, L.P., Jin, J., Zain, A.M.: An equivalent generating algorithm to model fuzzy Petri net for knowledge-based system. J. Intell. Manuf. **30**, 1831–1842 (2017)
8. Yeung, D.S., Wang, X.Z., Tsang, E.C.: Handling interaction in fuzzy production rule reasoning. IEEE Trans. Syst. Man Cybern. Part B (Cybern.) **34**(5), 1979–1987 (2004)
9. Spätgens, T., Schoonen, R.: The semantic network, lexical access, and reading comprehension in monolingual and bilingual children: an individual differences study. Appl. Psycholinguist. **39**(1), 225–256 (2018)
10. Ghimire, D., Jeong, S., Lee, J., Park, S.H.: Facial expression recognition based on local region specific features and support vector machines. Multimed. Tools Appl. **76**(6), 7803–7821 (2017)
11. Zhou, K.Q., Zain, A.M., Mo, L.P.: Dynamic properties of fuzzy Petri net model and related analysis. J. Central South Univ. **22**(12), 4717–4723 (2015)
12. Zhou, K.Q., Zain, A.M., Mo, L.P.: A decomposition algorithm of fuzzy Petri net using an index function and incidence matrix. Expert Syst. Appl. **42**(8), 3980–3990 (2015)
13. Zhou, K.Q., Gui, W.H., Mo, L.P., Zain, A.M.: A bidirectional diagnosis algorithm of fuzzy Petri net using inner-reasoning-path. Symmetry **10**, 192 (2018)
14. Zhou, K.Q., Zain, A.M.: Fuzzy Petri nets and industrial applications: a review. Artif. Intell. Rev. **45**(4), 405–446 (2016)
15. Liu, H.C., You, J.X., Li, Z., Tian, G.: Fuzzy Petri nets for knowledge representation and reasoning: a literature review. Eng. Appl. Artif. Intell. **60**, 45–56 (2017)
16. Shen, V.R., Chang, Y.S., Juang, T.T.Y.: Supervised and unsupervised learning by using Petri nets. IEEE Trans. Syst. Man Cybern.-Part A: Syst. Hum. **40**(2), 363–375 (2010)
17. Tsang, E.C., Yeung, D.S., Lee, J.W.: Learning capability in fuzzy Petri nets. In: IEEE SMC 1999 Conference Proceedings, vol. 3, pp. 355–360. IEEE (1999)
18. Wang, W.M., Peng, X., Zhu, G.N., Hu, J., Peng, Y.H.: Dynamic representation of fuzzy knowledge based on fuzzy petri net and genetic-particle swarm optimization. Expert Syst. Appl. **41**(4), 1369–1376 (2014)
19. Yeung, D.S., Tsang, E.C.: Weighted fuzzy production rules. Fuzzy Sets Syst. **88**(3), 299–313 (1997)
20. Tsang, E.C., Yeung, D.S., Lee, J.W., Huang, D.M., Wang, X.Z.: Refinement of generated fuzzy production rules by using a fuzzy neural network. IEEE Trans. Syst. Man Cybern. Part B (Cybern.) **34**(1), 409–418 (2004)

21. Ding, Z., Zhou, Y., Zhou, M.: Modeling self-adaptive software systems by fuzzy rules and Petri nets. IEEE Trans. Fuzzy Syst. **26**(2), 967–984 (2018)
22. Nabaei, A., et al.: Topologies and performance of intelligent algorithms: a comprehensive review. Artif. Intell. Rev. **49**(1), 79–103 (2018)
23. İnkaya, T., Akansel, M.: Coordinated scheduling of the transfer lots in an assembly-type supply chain: a genetic algorithm approach. J. Intell. Manuf. **28**(4), 1005–1015 (2017)
24. Morini, M., Pellegrino, S.: Personal income tax reforms: a genetic algorithm approach. Eur. J. Oper. Res. **264**(3), 994–1004 (2018)
25. Hou, Y., Wu, N., Zhou, M., Li, Z.: Pareto-optimization for scheduling of crude oil operations in refinery via genetic algorithm. IEEE Trans. Syst. Man Cybern.: Syst. **47**(3), 517–530 (2017)
26. Sarkheyli, A., Zain, A.M., Sharif, S.: The role of basic, modified and hybrid shuffled frog leaping algorithm on optimization problems: a review. Soft. Comput. **19**(7), 2011–2038 (2015)
27. Hasanien, H.M.: Shuffled frog leaping algorithm for photovoltaic model identification. IEEE Trans. Sustain. Energy **6**(2), 509–515 (2015)
28. Kawaria, N., Patidar, R., George, N.V.: Parameter estimation of MIMO bilinear systems using a Levy shuffled frog leaping algorithm. Soft. Comput. **21**(14), 3849–3858 (2017)
29. Dash, R.: Performance analysis of a higher order neural network with an improved shuffled frog leaping algorithm for currency exchange rate prediction. Appl. Soft Comput. **67**, 215–231 (2018)