Journal of Engineering and Applied Sciences 12 (17): 4330-4333, 2017
ISSN: 1816-949X

A Particle Swarm Optimization Conflict Resolution Model for Computer Network Diagnostics

Aaron Don M. Africa
Department of Electronics and Communications Engineering,
De La Salle University, 2401 Taft Avenue, 1004 Manila, Philippines

Abstract: Computer networks are sensitive systems and are prone to error. Every time there is an error in a computer network it needs to be solved at the soonest possible time so productivity will not be affected. One problem encountered in diagnosing an error is we do not know it's possible cause and because it is unknown, fixing the problem takes a lot of time. Trial and error is often employed to diagnose the problem. The predicament with trial and error is instead of fixing the problem it might make the problem worse. Knowing the possible cause of the problem before hand saves a lot of time in diagnostics. One tool that can be used to find the possible cause of problems in computer networks is an expert system. This system simulates human experts in diagnosing the problem. The problem with expert systems is that there may be multiple rules and the system may not know which one to fire. This research tries to solve that problem by applying the Particle Swarm Optimization (PSO) to the rules of an expert system, so it can give Impasse Weights (IW) to the rules and determine which rule is to fire. The conflict resolution algorithm for this research was tested on sample data of the problems encountered in computer networks. This research showed that particle swarm optimization can be used for an expert system conflict resolution.

Key words: Expert systems, computer networks, conflict resolution, particle swarm optimization, sensitive systems, problem

INTRODUCTION

Many companies integrate computer infrastructures in their business (Kahar *et al*., 2012) in order to do their work faster and more efficient. Their important documents and information are in their email, database, word documents etc. One part of this computer infrastructure is computer networks. A computer network is composed of the network infrastructure of the company and is used for communications (Steinder and Sethi, 2014). Parts of it are hardware devices like routers switch and modems (Flammini *et al*., 2005). There are instances when the computer network of the company encounters a problem which causes productivity to be hindered because employees of the company may not finish their research. It is important that when a problem in the computer network is encountered it should be solved at the soonest possible time. Diagnosing the problem manually often results in trial and error. Diagnosing through trial and error often takes a lot of time and sometimes the improper use of applying this method may damage the computer network and make the problem worse (Stevens, 2014).

One method to efficiently diagnose problems in computer networks is to use an expert system. An expert system simulates a human expert and gives the possible cause of the problem by providing it with the symptoms encountered. An expert system reduces trial and error (Garibaldi *et al*., 2012). This system searches its database for the rule that relates the symptoms entered and fires those rules. A common problem of an expert system is there may be 2 or more rules given a set of symptoms. Because of this the expert system will be conflicted not knowing which rule to fire (Yager, 2000). This research proposes a novel conflict resolution algorithm for the rules of an expert system. This algorithm uses particle swarm optimization to give Impasse Weights (IW) to the rules of the expert system to determine which of the conflicting rules are more suitable to fire.

MATERIALS AND METHODS

Expert systems: In the field of artificial intelligence, an expert system is a computer-based system that simulates a human expert (Stoia, 2013). These systems are designed to solve complex diagnostic problems through advance reasoning. Expert systems are usually composed of IF THEN rules which the system follows to solve a problem (Ellis and Mathews, 2002). An expert system is divided into two parts, namely: the knowledge base and inference engine. The knowledge base of an expert system contains the data and rules while in interference engine contains new information and deduce data (Nan *et al*., 2008). Expert systems were introduced by the stanford heuristic programming project. This project was led by Edward Feigenbaum in 1977. The research identified domains where human expertise was needed like in organic molecular biology and diagnosing infectious diseases. An expert system is in the category of a knowledge base system. These systems have knowledge base architecture. The knowledge base of an expert system contains the facts about the field where it is applied (Burdorf and Swuste, 1999). For example an expert system for computer networks contains data about the symptoms and possible causes of problems encountered in the field of computer networks.

The main advantage of an expert system is it reduces trial and error. The possible cause of the problem can be known given enough information (Garibaldi *et al*., 2012). Expert systems are also easy to maintain it is because if new information is obtained there is no need to write a new code about it. Expert systems also retain the information to the company unlike in a human expert. An example is if the human expert resigns from the company his knowledge about the system may be lost with him while an expert system will survive changes with company employees.

One disadvantage of an expert system is the conflict between its rules. One set of symptoms may produce 2 or more possible causes and the expert system will not know which one to fire (Yager, 2000). This research solved that problem by applying a conflict resolution algorithm using particle swarm optimization to the rules of the expert system. The conflict resolution algorithm gives Impasse Weights (IS) to the rules to know which rule to fire.

Particle Swarm Optimization (PSO): Particle Swarm Optimization (PSO) is a type of computational method where a problem is iterated several times for certain duration or an ideal solution is discovered (Tsafarakis *et al*., 2011). THE PSO optimizes a problem by having population of candidate solutions or particles search each state space to find the optimum solution. The position of the particles is influenced by its best local point and also guided by the best positions in the state space (Liu *et al*., 2005). The positions are updated constantly for the best solution. The PSO optimization model was created by Kennedy and Eberhart. These two inventors originally developed a computer software to track the social behaviors of bird flocking to find food sources but later discovered that their algorithms can be used to solve optimization problems. Since, the time it was invented the PSO has been applied to many research

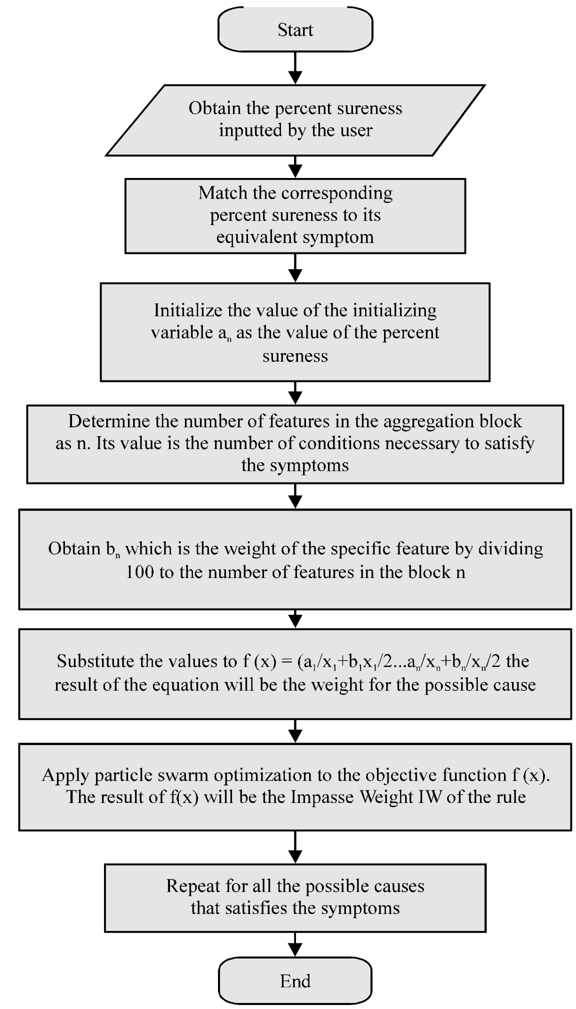

Fig. 1: Conflict Resolution algorithm

areas and applications (Trelea, 2003). The rules of the expert system can be represented into an objective function. The PSO searches for the best value of the objective function after several iterations. The value of the objective function becomes its Impasse Weight (IW) (Kahar *et al*., 2012). The higher the IW the higher is the likelihood that the rule contains the possible cause of the problem, so that the rule will be fired.

Conflict resolution algorithm: Figure 1 presents the conflict resolution algorithm. The first step is to obtain the percent sureness of the symptoms from the user. The corresponding percent sureness is then matched with its equivalent symptom. The initializing variable a_n is initialized as the value of the percent sureness. The number of conditions necessary to satisfy the symptom becomes the value of the aggregation block n. The weight of the specific feature is then obtained by dividing 100 to the number of features in block n. The result becomes b_n. The values are then substituted to the formula of the objective function which is: $f(x) = (a_1/x_1+b_1x_1/2 \ldots a_n/x_n+b_n/x_n/2)$. Particle Swarm Optimization (PSO) is then applied to the objective function f(x). The result of the objective function will be the Impasse Weight (IW) of the rule. The procedure is then repeated for all the rules of the expert system.

RESULTS AND DISCUSSION

The variables S1 to Sn represent the symptoms while D represents the possible cause Sn will have a value of 1 if it exists while 0 if it does not exist. The objective function of the rules is represented as f(x). Particle Swarm Optimization (PSO) will be applied to f(x) and the result will be the Impasse Weight (IW) of the rule. In case there are conflicting rules the rule with the highest IS will be fired as it is most likely the possible cause.

Table 1 presents sample symptoms in computer networks. Table 2 shows a list of possible cause in computer networks. Table 3 presents the test data of the conflict resolution algorithm. Table 3 also shows that the rules can be represented into an objective function and Particle Swarm Optimization (PSO) can be used to give Impasse Weights (IW) to the rules. In case there are conflicting rules the rule with a higher IW will be fired. An example is in rules 13 and 14. The two rules have the same set of symptoms but applying the algorithm rule number 14 has a higher IW therefore, its possible cause is most likely the problem and this rule will be the one to be fired.

Table 1: List of symptoms in computer networks

ID	Symptom
S1	Runtime errors
S2	Divide errors
S3	Network dataport problem
S4	Firewall disabled
S5	Pinged server error
S6	Firewall software cannot be detected
S7	MAC address conflict
S8	LAN malfunction
S9	Domain name service error
S10	MOM alerts critical error
S11	Server harddisk full
S12	Mapping error
S13	WIN driver error
S14	HTTP error
S15	MAC Portal error

Table 2: List of possible causes in computer networks

ID	Symptoms
PC1	NET defective
PC2	Card error
PC3	Mapping error
PC4	Wire problem
PC5	Login error
PC6	Server full
PC7	OS error
PC8	Shared folder error

Table 3: Test data of the conflict resolution algorithm

IF THEN rule	Initializing variables	Conflict resolution formula	Impasse Weights (IW)
IF (S1 = 1) and (S3 = 1) => D = PC1)	$a_1 = 70$, $a_2 = 50$, $b_1 = 0.5$, $b_2 = 0.5$	$f(X) = (a_1/x_1+b_1x_1/2)+(a_2/x_2+b_2x_2/2)$	35.00
IF (S1 = 1) and (S4 = 1) and (S5 = 1) => (D = PC1)	$a_1 = 70$, $a_2 = 30$, $a_3 = 40$, $b_1 = 0.333$, $b_2 = 0.333$, $b_3 = 0.333$	$f(X) = (a_1/x_1+b_1x_1/2)+(a_2/x_2+b_2x_2/2))+(a_3/x_3+b_3x_3/2$	26.92
IF (S2 = 1) and (S4 = 1) and (S5 = 1) and (S6 = 1) => (D = PC2)	$a_1 = 70$, $a_2 = 60$, $a_3 = 40$, $a_4 = 40$, $b_1 = 0.25$, $b_2 = 0.25$, $b_3 = 0.25$, $b_4 = 0.25$	$f(X) = (a_1/x_1+b_1x_1/2)+(a_2/x_2+b_2x_2/2)+(a_3/x_3+b_3x_3/2)+(a_4/x_4+b_4x_4/2)$	30.00
IF (S2 = 1) and (S4 = 1) and (S6 = 1) and (S7 = 1) => (D = PC3)	$a_1 = 50$, $a_2 = 40$, $a_3 = 40$, $a_4 = 60$, $b_1 = 0.25$, $b_2 = 0.25$, $b_3 = 0.25$, $b_4 = 0.25$	$f(X) = (a_1/x_1+b_1x_1/2) +(a_2/x_2 + b_2x_2/2)+(a_3/x_3+b_3x_3/2)+(a_4/x_4+b_4x_4/2)$	29.87
IF (S6 = 1) and (S7 = 1) and (S8 = 1) and (S9 = 1) => (D = PC3)	$a_1 = 80$, $a_2 = 80$, $a_3 = 60$, $a_4 = 70$, $b_1 = 0.25$, $b_2 = 0.25$, $b_3 = 0.25$, $b_4 = 0.25$	$f(X) = (a_1/x_1+b_1x_1/2)+(a_2/x_2+b_2x_2/2)+(a_3/x_3+b_3x_3/2)+(a_4/x_4+b_4x_4/2)$	37.16
IF (S8 = 1) and (S9 = 1) and (S10 = 1) and (S11 = 1) => (D = PC4)	$a_1 = 30$, $a_2 = 30$, $a_3 = 80$, $a_4 = 90$, $b_1 = 0.25$, $b_2 = 0.25$, $b_3 = 0.25$, $b_4 = 0.25$	$f(X) = (a_1/x_1+b_1x_1/2)+(a_2/x_2+b_2x_2/2)+(a_3/x_3+b_3x_3/2)+(a_4/x_4+b_4x_4/2)$	31.63
IF (S8 = 1) and (S9 = 1) and (S12 = 1) and (S15 = 1) => (D = PC4)	$a_1 = 90$, $a_2 = 90$, $a_3 = 70$, $a_4 = 70$, $b_1 = 0.25$, $b_2 = 0.25$, $b_3 = 0.25$, $b_4 = 0.25$	$f(X) = (a_1/x_1+b_1x_1/2)+(a_2/x_2+b_2x_2/2)+(a_3/x_3+b_3x_3/2)+(a_4/x_4+b_4x_4/2)$	36.00
IF (S2 = 1) and (S5 = 1) and (S9 = 1) and (S11 = 1) => (D = PC4)	$a_1 = 80$, $a_2 = 50$, $a_3 = 50$, $a_4 = 70$, $b_1 = 0.25$, $b_2 = 0.25$, $b_3 = 0.25$, $b_4 = 0.25$	$f(X) = (a_1/x_1+b_1x_1/2)+(a_2/x_2+b_2x_2/2)+(a_3/x_3+b_3x_3/2)+(a_4/x_4+b_4x_4/2)$	32.00
IF (S11=1) and (S12 = 1) and (S13 = 1) and (S14 = 1) => (D = PC5)	$a_1 = 90$, $a_2 = 60$, $a_3 = 90$, $a_4 = 50$, $b_1 = 0.25$, $b_2 = 0.25$, $b_3 = 0.25$, $b_4 = 0.25$	$f(X) = (a_1/x_1+b_1x_1/2)+(a_2/x_2+b_2x_2/2)+(a_3/x_3+b_3x_3/2)+(a_4/x_4+b_4x_4/2)$	34.00
IF (S1 = 1) and (S3 = 1) and (S10 = 1) and (S15 = 1) => (D = PC6)	$a_1 = 70$, $a_2 = 60$, $a_3 = 90$, $a_4 = 90$, $b_1 = 0.25$, $b_2 = 0.25$, $b_3 = 0.25$, $b_4 = 0.25$	$f(X) = (a_1/x_1+b_1x_1/2)+(a_2/x_2+b_2x_2/2)+(a_3/x_3+b_3x_3/2)+(a_4/x_4+b_4x_4/2)$	35.43
IF (S2 = 1) and (S4 = 1) and (S5 = 1) and (S14 = 1) => (D = PC6)	$a_1 = 80$, $a_2 = 90$, $a_3 = 40$, $a_4 = 60$, $b_1 = 0.25$, $b_2 = 0.25$, $b_3 = 0.25$, $b_4 = 0.25$	$f(X) = (a_1/x_1+b_1x_1/2)+(a_2/x_2+b_2x_2/2)+(a_3/x_3+b_3x_3/2)+(a_4/x_4+b_4x_4/2)$	47.00
IF (S5 = 1) and (S7 = 1) and (S8 = 1) and (S10 = 1) => (D = PC7)	$a_1 = 70$, $a_2 = 60$, $a_3 = 70$, $a_4 = 40$, $b_1 = 0.25$, $b_2 = 0.25$, $b_3 = 0.25$, $b_4 = 0.25$	$f(X) = (a_1/x_1+b_1x_1/2)+(a_2/x_2+b_2x_2/2)+(a_3/x_3+b_3x_3/2)+(a_4/x_4+b_4x_4/2)$	50.67

Table 3: Continue

IF THEN rule	Initializing variables	Conflict resolution formula	Impasse Weights (IW)
IF (S10 = 1) and (S11 = 1) and (S12 = 1) and (S15 = 1) => (D = PC7)	$a_1 = 80$, $a_2 = 80$, $a_3 = 40$, $a_4 = 90$, $b_1 = 0.25$, $b_2 = 0.25$, $b_3 = 0.25$, $b_4 = 0.25$	$f(X) = (a_1/x_1+b_1x_1/2)+(a_2/x_2+b_2x_2/2)+(a_3/x_3+b_3x_3/2)+(a_4/x_4+b_4x_4/2)$	34.59
IF (S10 = 1) and (S11 = 1) and (S12 = 1) and (S15 = 1) => (D = PC8)	$a_1 = 90$, $a_2 = 60$, $a_3 = 40$, $a_4 = 90$, $b_1 = 0.25$, $b_2 = 0.25$, $b_3 = 0.25$, $b_4 = 0.25$	$f(X) = (a_1/x_1+b_1x_1/2)+(a_2/x_2+b_2x_2/2)+(a_3/x_3+b_3x_3/2)+(a_4/x_4+b_4x_4/2)$	59.00
IF (S9 = 1) and (S10 = 1) and (S11 = 1) and (S15 = 1) => (D = PC8)	$a_1 = 80$, $a_2 = 50$, $a_3 = 50$, $a_4 = 40$, $b_1 = 0.25$, $b_2 = 0.25$, $b_3 = 0.25$, $b_4 = 0.25$	$f(X) = (a_1/x_1+b_1x_1/2)+(a_2/x_2+b_2x_2/2)+(a_3/x_3+b_3x_3/2)+(a_4/x_4+b_4x_4/2)$	36.00

CONCLUSION

The sample data in this research showed that conflicting rules are possible in expert systems because the same set of symptoms can have different possible causes. When conflicting rules exist the expert system will have a problem on which rule to fire. This research solved that problem by proposing a novel conflict resolution algorithm which can be used for the conflicting rules of an expert system. The algorithm converted the rules of the expert system into an objective function. The Particle Swarm Optimization (PSO) was used to solve the objective function and give Impasse Weights (IW) to the rules. In theory the higher the IW the most likely that it is the possible cause, so that rule will be fired.

The algorithm was tested only with a sample data for computer networks. For future studies we recommend that the algorithm be tested on live data. This algorithm is not just for expert systems in computer networks but can also be used in other fields.

REFERENCES

Burdorf, A. and P. Swuste, 1999. An expert system for the evaluation of historical asbestos exposure as diagnostic criterion in asbestos-related diseases. Ann. Occup. Hyg., 43: 57-66.

Ellis, M.W. and E.H. Mathews, 2002. Needs and trends in building and HVAC system design tools. Build. Environ., 37: 461-470.

Flammini, M., A. Navarra and A. Proskurowski, 2005. On routing of wavebands for all-to-all communications in all-optical paths and cycles. Theor. Comput. Sci., 333: 401-413.

Garibaldi, J.M., S.M. Zhou, X.Y. Wang, R.I. John and I.O. Ellis, 2012. Incorporation of expert variability into breast cancer treatment recommendation in designing clinical protocol guided fuzzy rule system models. J. Biomed. Inf., 45: 447-459.

Kahar, R., F. Yamimi, G. Bunari and H. Habil, 2012. Trusting the social media in small business. Procedia Soc. Behav. Sci., 66: 564-570.

Liu, B., L. Wang, Y.H. Jin, F. Tang and D.X. Huang, 2005. Improved particle swarm optimization combined with chaos. Chaos Solitons Fractals, 25: 1261-1271.

Nan, C., F. Khan and M.T. Iqbal, 2008. Real-time fault diagnosis using knowledge-based expert system. Process Saf. Environ. Prot., 86: 55-71.

Steinder, I.M. and A.S. Sethi, 2004. A survey of fault localization techniques in computer networks. Sci. Comput. Programming, 53: 165-194.

Stevens, E., 2014. Fuzzy front-end learning strategies: Exploration of a high-tech company. Technovation, 34: 431-440.

Stoia, C.L., 2013. A study regarding the use of expert systems in economics field. Procedia Econ. Finance, 6: 385-391.

Trelea, I.C., 2003. The particle swarm optimization algorithm: Convergence analysis and parameter selection. Inf. Process. Lett., 85: 317-325.

Tsafarakis, S., Y. Marinakis and N. Matsatsinis, 2011. Particle swarm optimization for optimal product line design. Int. J. Res. Marketing, 28: 13-22.

Yager, R.R., 2000. Approximate reasoning and conflict resolution. Int. J. Approximate Reasoning, 25: 15-42.