

Ant Quantitative Algorithms Applied to Electronic Systems Diagnosis

Józef Drabarek, Piotr Kania

Faculty of Electronics and Informatics

Koszalin University of Technology

75 – 453 Koszalin, ul. Śniadeckich 2, POLAND

e-mail:drabarek@ie.tu.koszalin.pl

ABSTRACT

Main problems of using ant algorithms in diagnosis are presented in this article. A knowledge representation and a searching solution mechanism with help of ant algorithms were elaborated. The usefulness of an ant-quantitative algorithm in faults localizations is presented on TV set.

1. INTRODUCTION

In recent years one can observe the increasing interest in hybrid expert systems applied to diagnostics [6][7]. It consists of different known artificial intelligence methods, enhancing the intellectual potential of the entire system. To build a hybrid expert system one can actually use classic rule-object systems, artificial neural network and genetic algorithms[6][7].

In author's opinion ant algorithms can improve a fault location process. They were introduced by M. Dorigo and his colleagues to help optimization problems solving [1][2]. The authors have worked out the ant algorithm implementation to electronic system diagnostics, which was realized using SCILAB, the free math software from INRIA.

The reason of the interest shown in ant algorithms were such characteristic features as:

- a small complexity of single agents (ants);
- the mechanism of their cooperation.

Features, such as a low complexity, in combination with a relatively simple implementation in the form of an iterative computer algorithm are decisive as regards the attractiveness of this algorithm. The author has found out similarities in fault location process carried out by a technician and ants searching for the shortest way from an anthill to a food source. The algorithm concept [1][3] is based on a digraph representation of the ant paths where the graph distance is described by the intensity of a trail. The algorithm is carried out iteratively and after each cycle the trail increment $\Delta\tau_{ij}(t, t+1)$ is put on the edge ij with intensity inversely proportional to the tour length for every visiting ant, and the new trail is given by (1)

$$\tau_{ij}(t+1) = \rho\tau_{ij}(t) + \Delta\tau_{ij}(t, t+1) \quad (1)$$

where ρ , denotes the evaporation of the a trail.

The ant which is located at the i -node selects further way to the j -node with probability given by

$$P_{ab} = \frac{[\tau_{ab}(t)]^\alpha [\eta_{ab}]^\beta}{\sum_{b \in D} [\tau_{ab}(t)]^\alpha [\eta_{ab}]^\beta} \quad (2)$$

where:

η_{ij} - denotes the j th-node visibility form the i th-node,
 G - the list of nodes which can be chosen by an ant at the i th-node,
 α, β algorithm constants.

In those algorithms, a group of single agents (ants) searches jointly a solution to the problem posed to them. Each ant searches a solution on its own and the result of its work has a partial influence on its final solution. The ants do not communicate with one another but leave hinds (pheromone traces), which have an influence on the choices made by other individuals. The activities undertaken by ants have a clear and precious optimizing feature: an effort to find the shortest way in the shortest time.

The way covered by each ant can be simplified to a directed graph (Fig. 1) with one vertex (it is assumed that this is an anthill – starting point (S) for each ant) and with edges going away from the vertex, which lead to successive nodes (1 to 6 on the Figure): points in which the ants make further decisions concerning the choice of the route. The ants finish their route once they reach one of the targets (A, B, C, or D).

The algorithm is done in an iterative manner: in the first step, ant no. 1 appears on the start; it makes a choice and goes to another node, while leaving a trace on the branch which it visited. In another step, ant no. 1 makes yet another choice, goes to another node, leaves a trace, while on the start of the graph, there is another ant, which makes choices according to the pattern presented above.

Between the serious of further steps taken by all the active ants, there is a process of pheromone evaporation, i.e. from all the branches of the graph, certain determined pheromone value is subtracted. This process is necessary to prevent an accumulation of the pheromone trace, and to prevent the ants from going the same routes all the time. There are 3 basic types of Ant Algorithms, which differ from each other with regard to the manner in which the pheromone track is left:

1. Ant-density: in which the ants leave a constant, pheromone value on their way, independent of the length of the graph edge;

2. Ant-quantity: in which the ants leave on its way a pheromone value inversely proportional to the edge length, which they covered on their way:

3. Ant-cycle: in which the ant leaves a pheromone trace on the graph branches it visited only after reaching the target.

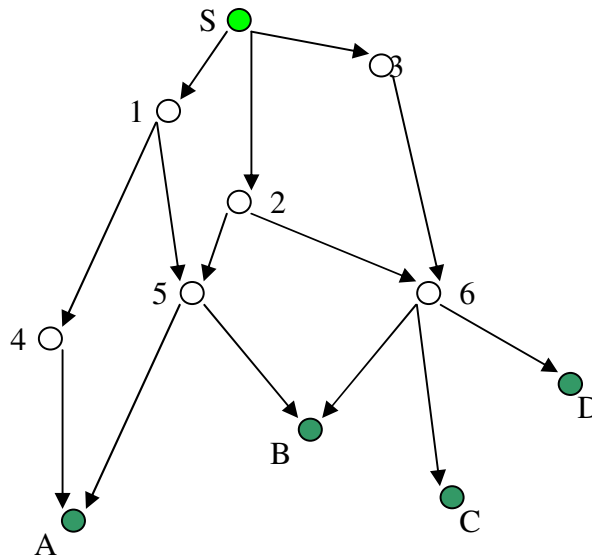


Fig. 1. An example graph representing the route of searches

In the present article, a quantitative algorithm was applied, in which the process of the movement of consecutive ants occurs in accordance with the patterns:

Step 1. (Fig.2)

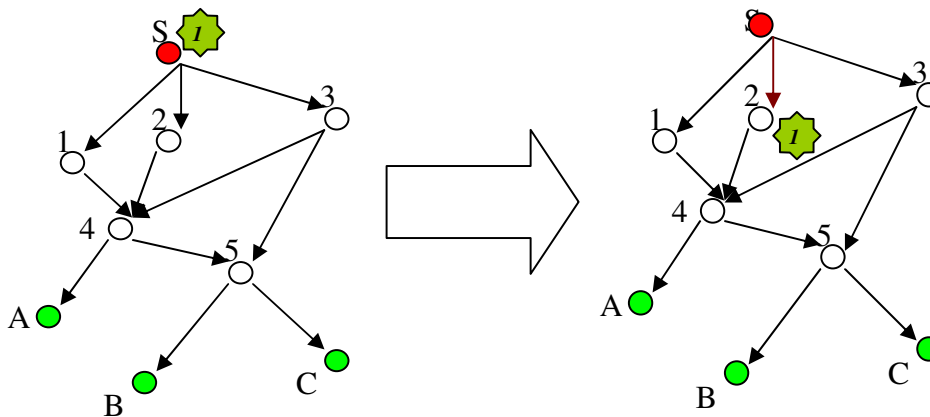


Fig.2. Quantitative Algorithm– step 1

Ant no. 1 stands at the beginning of the route (S) and samples the route to another point (1, 2 or 3) and leaves a pheromone trace behind it with the intensity inversely proportional to the length of the route branch selected (the shorter the branch the more intensive the trace).

Step 2. (Fig. 3)

Pheromone evaporation from all the branches of the graph occurs.

Ant no. 1 continues its route while selecting at random its further route.

Ant no. 2 appears at the start of the route (S) and selects at random its route to another point (1, 2, or 3).

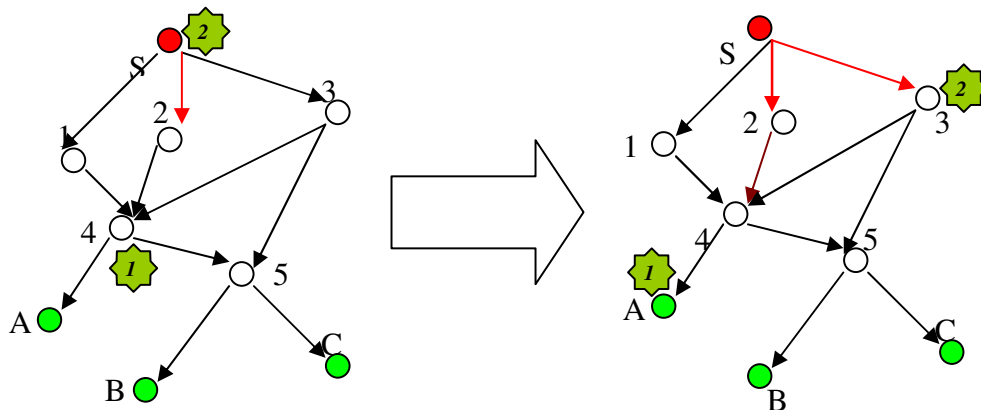


Fig. 3. Quantitative Algorithm – step 2

Step 3. (Fig. 4)

Pheromone evaporation from all the branches of the graph occurs.

Ant no. 1 continues its travel while selecting at random its further route – it leaves at trace on the edge.

Ant no. 2 continues its travel while selecting at random its further route – it leaves at trace on the edge.

Ant no. 3 appears at the start of the route (S) and selects at random its route to another point (1, 2, or 3)

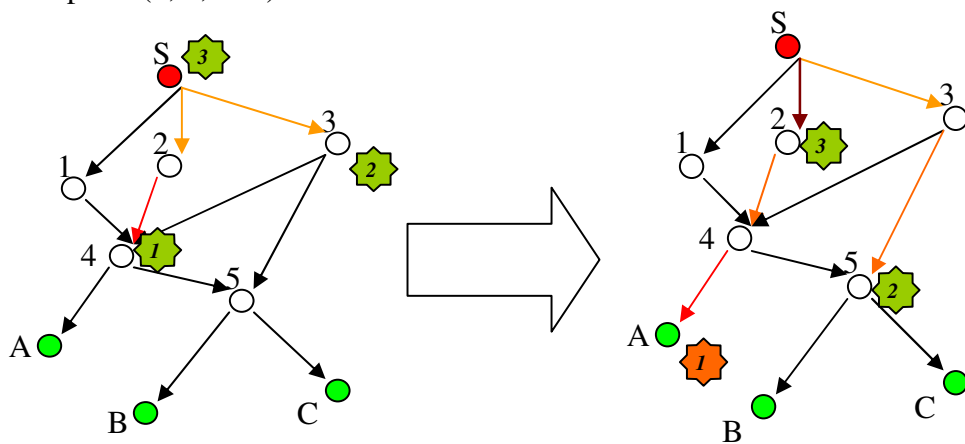


Fig. 4. Quantitative Algorithm – step 3

...etc. until all the ants reach the target.

After all the ants go, the most optimal route will be the one with the strongest pheromone trace.

As it can be seen in this type of algorithm, the key role is played by the lengths of individual edges, because the quantity of pheromone left depends from them, and so does the selection of the optimal path – solution.

2. ANT ALGORITHMS APPLIED TO DIAGNOSTICS

2.1. Knowledge representation for ant algorithms

The most important part of the implementation is to build a graph representation of possible ant tours. For the case of electronic system diagnosis one needs to elaborate the graph generation technique using service information or an expert system data base. A convenient knowledge representation is a block diagram containing service activities used to carry out a failure. Such knowledge is published in service manuals.

Usually it contains boxes with service questions about system symptoms. The boxes are connected by lines with answers: yes or no. For such representation one can build a digraph where the boxes denote the ant algorithm nodes. To describe the graph one can use the sentences (3-4) binding graph edges with symbols.

$$s < n > \quad s < m > y / n \quad (1)$$

or

$$s < n > \quad o < m > y/n \quad (2)$$

where:

- $s < k >$ -denotes a k th-symptom
- $o < k >$ -denotes a k th-object
- y/n – *yes* or *no* edge.

In Figure 5 a simple graph example is presented. It can be described by the sentences shown in Table 1.

Table 1. Knowledge data for the graph in Figure 5.

S1	S2	n
S2	S3	y
S2	O1	n
S3	O2	n

To implement the ant algorithm one needs to assign weight values to each graph edge. In the case of diagnosis a service efficiency coefficient expressed by eq (3) seems to be optimal for such task [2].

$$U_f = \frac{q_f}{t_f}, \quad (3)$$

where:

- q_f – denotes the probability of the f -fault occurrence;
- t_f – service activity duration needed to check the f -symptom state.

For the ant algorithm the visibility coefficient is given by (4).

$$\eta_{ij} = \frac{1}{U_f}. \quad (4)$$

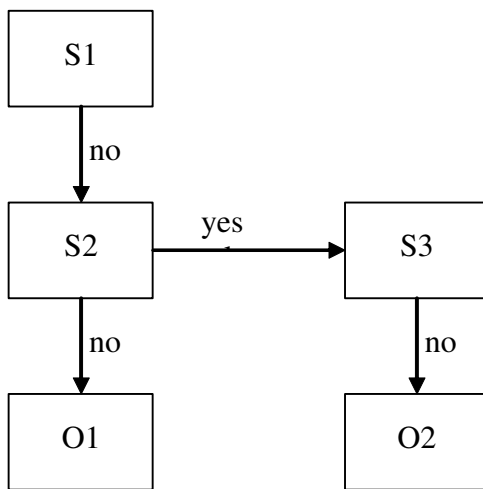


Fig.5. An example block diagram for a simple fault location process

The technique we will be presented using the service efficiency coefficients shown in Table 2.

Table 2. Service efficiency coefficients η_f for the graph in Figure 5.

symptom/object	q_f	t_f [hours]	η_f
s1	0.1	0.2	2
s2	0.01	0.3	30
s3	0.3	0.2	0.4
o1	0.1	0.8	8
o2	0.5	0.5	1

The coefficient η_f is assigned to edges pointed to f-node. Using Table 2 one can build the visibility coefficients shown in Table 3.

Table 3. Visibility coefficients.

	s1	s2	s3	o1	o2
s1	0	30	0	0	0
s2	0	0	0.666666	8	0
s3	0	0	0	0	1
o1	0	0	0	0	0
o2	0	0	0	0	0

Simulations of ant movements were performed for 10 ants. The results are given in Table 4.

Table 4. Simulations of ant movements

ant tours				tour length
s1	s2	s3	o2	31.666669
s1	s2	o1	-	38.000002
s1	s2	s3	o2	31.666669
s1	s2	s3	o2	31.666669
s1	s2	s3	o2	31.666669
s1	s2	o1	-	38.000002
s1	s2	s3	o2	31.666669
s1	s2	s3	o2	31.666669
s1	s2	s3	o2	31.666669
s1	s2	s3	o2	31.666669

One can conclude from Table 4 that the optimal fault location strategy is: check symptoms s1, s2, s3 and then check the object o2.

3. PRACTICAL EXAMPLE:DESIGN OF PROGRAMME

3.1. Main algorithm of the program

Once the program has been started, an option appears of the selection of the desired programme's function. There are five options to choose from:

1. Diagnosis, creation or modification of the knowledge base about the symptoms and objects of the device,

2. Creation or modification of the knowledge base about the symptoms and objects of the device,
3. Creation or modification of the knowledge base about the damage search track,
4. Search of an optimal way to locate the damage with the aid of AM,
5. Theory of Ant Algorithms.

Once the user decides which option to use in the program, he/she clicks on the button of the option. When option **1** is selected, the dialogue window opens “Open file with knowledge base about S and O” (Fig.7).

Reading in of this file allows the program to create a list of questions (symptoms) assigned to the user and to give possible solutions (objects).Next, a similar dialogue window opens: “Open file with knowledge base about relationships” (search track) (Fig.8).

Reading in of this file allows the program to ask questions in the proper order. The files of knowledge bases were divided as it is possible to construct many different tracks of the search of the damage by various experts. Once both files of the databases have been read in, the main form of the program opens on the “DIAGNOSIS” tab – Fig. 6.

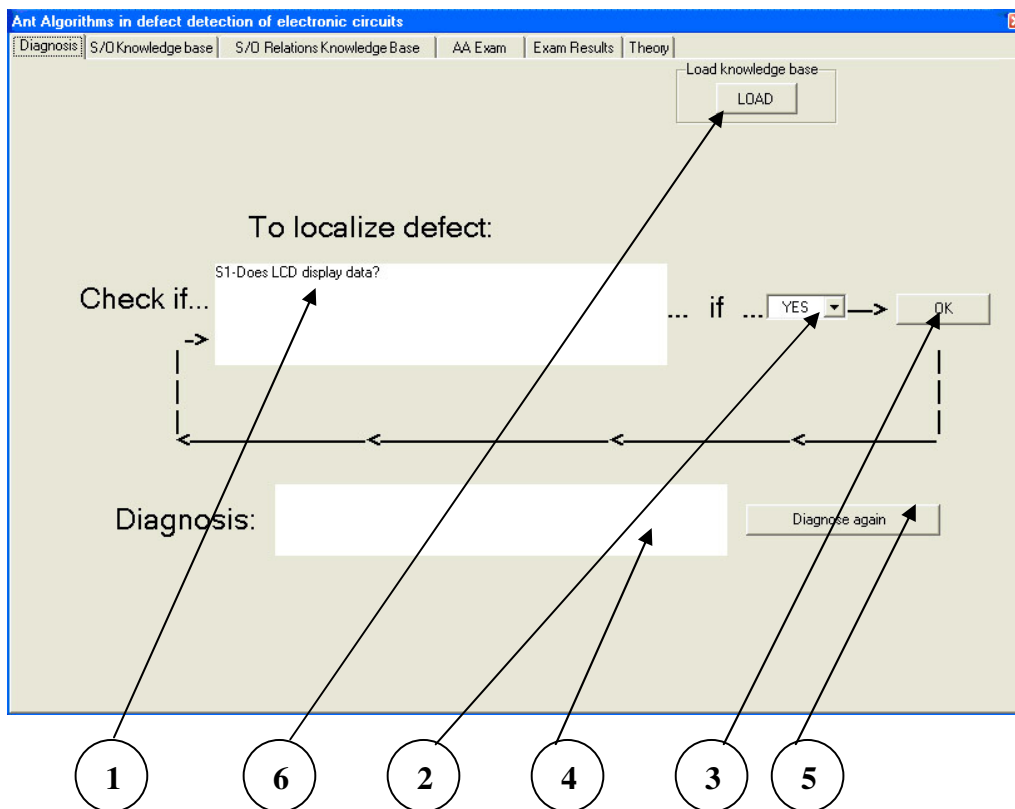


Fig.6. Diagnosis tab.

Automatically after the start, in the window **(1)** the first question appears, to which the user seeking the damage may answer: YES, NO or Y/N. The answer is selected by marking the suitable option on the expandable list **(2)**.The answer is confirmed by the “OK” button **(3)**. If there is no such option in the search track, the user will receive the appropriate message. If there is

such an option in the window (1), there will appear another symptom or message about the solution having been found. In such a case, in the window (4) of diagnosis there will appear a description of the activity (activities) to be performed in order to repair the damaged device. Additionally, should the user make a mistake while answering the questions or if the diagnosis turns out to be erroneous, the user may start the process again by pressing the button: “DIAGNOSE ONCE AGAIN” (5). If the user wants to diagnose a different device and read in different databases, this can be done with the help of the “READ IN” button (6) After the selection of option 2 from the start window, another selection window appears on the screen. At this moment, the user decides to do the following:

- A. to create a new knowledge base about S and O from the very beginning,
- B. to modify the existing knowledge base about S and O.

For example, if the option is selected – button (1) – the main form opens of the program on the tab called “KNOWLEDGE BASE ABOUT S/O” – Fig.7.

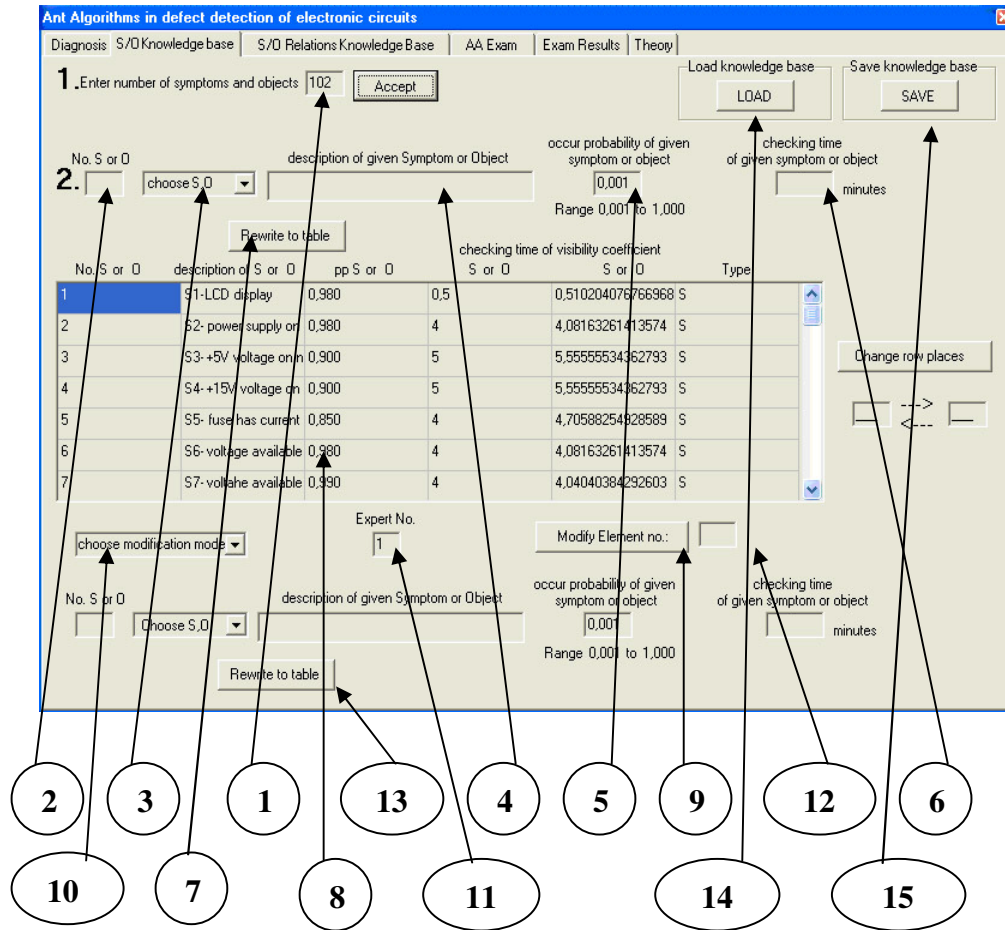


Fig.7. Knowledge base tab about S/O.

First, the number of symptoms and objects is entered (1).

After the numbers of S and O have been confirmed in the table (8), the number of lines is set which corresponds to the numbers of S and O. Starting from this moment, the user may begin to create the knowledge base.

The order in which data is to be provided:

- 1) The user gives number S or O (2).
- 2) The user selects the type from the list (3): Symptom or Object,
- 3) The user describes a given Symptom or Object (4),
- 4) The user gives the probability of the occurrence of a given Symptom or Object (5),
- 5) The user gives the time of the checking of a given Symptom or performing of activities connected with a given Object (6).

Next, all the data is copied to the table (8) with the button: “COPY TO TABLE”.

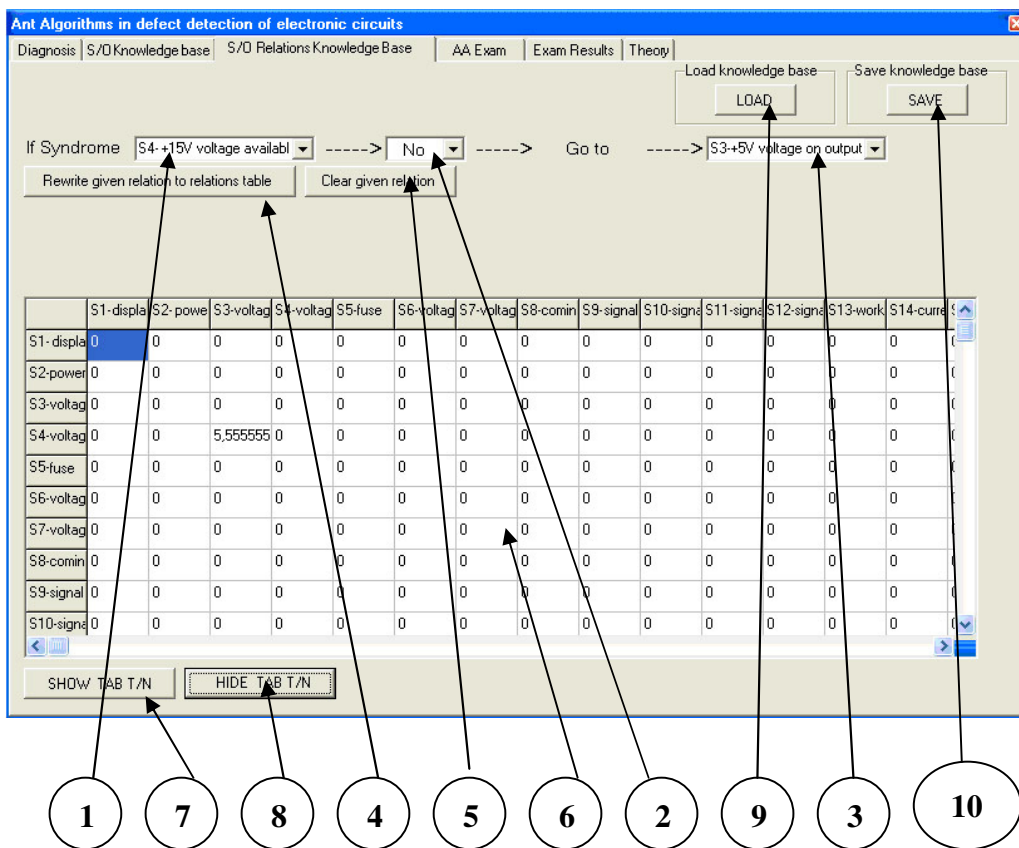


Fig.8. Knowledge base tab about S/O relations.

After the selection of the 3rd option from the start window (Fig.7) of the program the user may create or modify the knowledge base about the way of the damage location. The user saves the knowledge base filled in by pressing the “SAVE” button.

After the selection of the 4th option from the start window, the main form of the program opens on the tab of “AA testing” (Fig.9).

In order to conduct the search of an optimal (with the shortest possible time of duration and the largest probability of occurrence) search track, the user should perform the following activities:

1. Select a given algorithm from the expandable list (1):
 - a. (AD) Density,
 - b. (AQ) Quantity,
 - c. (AC) Cyclic.
2. Set with the slider (2) the number of ants searching for a solution,
3. Set with the slider (3) the pheromone dose left on the graph edges visited by the ant,
4. Set with the slider (4) the pheromone evaporation coefficient after one step has been taken by all the active ants,
5. Set with the slider (5) the ALFA coefficient, which refers to the significance of the pheromone trace while the ant makes a decision concerning going through,

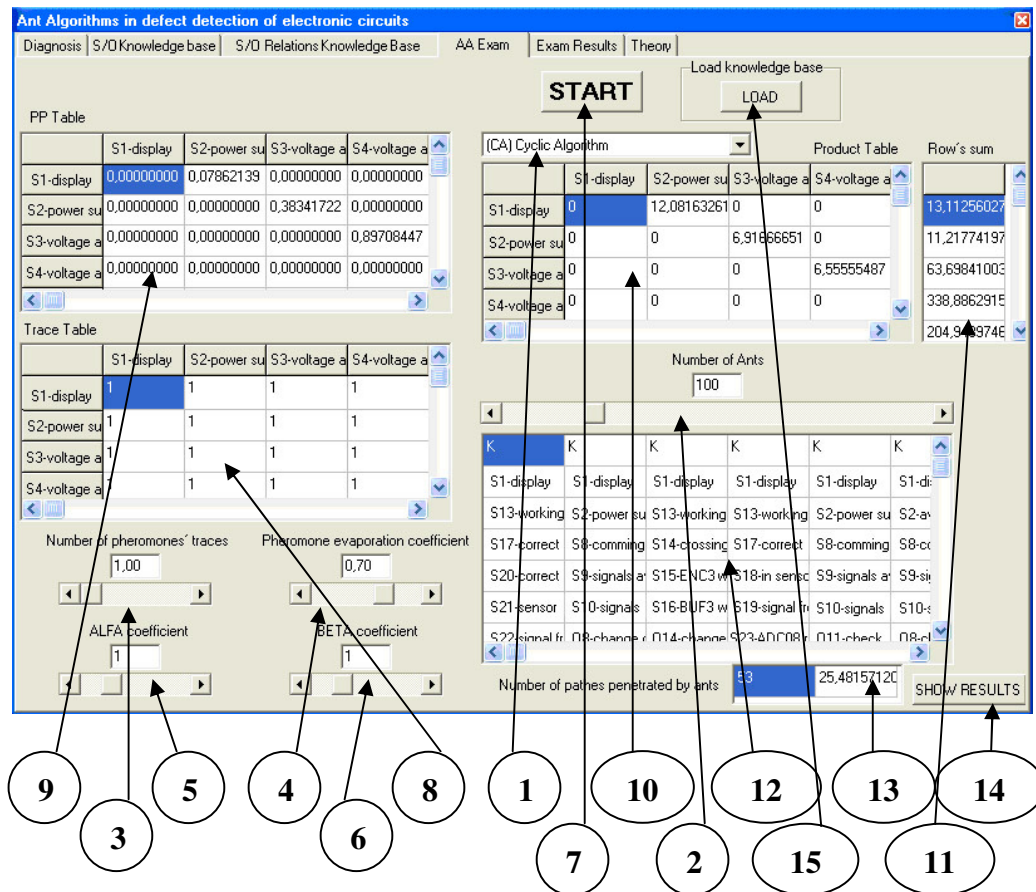


Fig.9. AA Examination tab.

6. Set with the slider (6) the BETA coefficient, which refers to the significance of the visibility coefficient of the target node while the ant makes a decision concerning going through,
7. The starting up of the Ant Algorithm occurs once the “START” (7) button has been pressed.

After the start of the Algorithm, on the basis of the data from tables (8), (9), (10) and (11), the ants commence a cooperative search of the solution. In table (12) in the individual rows (which correspond to given ants), subsequent nodes appear which have been visited by them during the trip. Once all the ants have completed their trip, the user has the possibility to see the results on the further bookmark in the main form.

Additionally, in order to make the work easier, in the field (13) the shortest route found appears: the number of the ant which has found it and the length of the route. If the user wishes to check the routes covered by all the ants, it is possible to do so by pressing “SHOW RESULTS” (14). Then, the tab of “Test Results” opens in the main form of the program (Fig. 10).

This tab serves the purpose of the observation of the results and their possible saving to a file. From this table we can read ant number, which found the shortest path to remove this defect.

ANT NO.	PATH LENG.	LIST OF STEPS ON THE ANT'S PATH									
1	118,7962956	S1-display	S2-power sup	S8-coming	S9-signals pr	S10-signals p	D11-check				
2	165,4440918	S1-display	S13-working	S14-is conne	S15-ENC3 de	S16-BUF3 dz	D14-change				
3	115,4440842	S1-display	S13-working	S14-is conne	S15-ENC3 de	S16-BUF3 dz	D11-check				
4	25,48157120	S1-display	S13-working	S17-correct	S20-correct	S28-drivers	S33-drivers	S34-FR sens	S35-signal fr	S31-is repair	D25-
5	133,6454010	S1-display	S13-working	S17-correct	S20-correct	S21-sensor	S22-signal fr	S23-ADC08 r	S24-to ADC	S25-signals p	S26-

SHORTEST FOUND PATH											
ANT NO.	PATH LENG.	LIST OF STEPS ON THE ANT'S PATH									
4	25,48157120	S1-display	S13-working	S17-correct	S20-correct	S28-drivers	S33-drivers	S34-FR sens	S35-signal fr	S31-is repair	D25-

Fig.10. AA Test Results tab.

AN EXAMPLE

Presented program was applied to service a TV set. The fault location diagram is presented in Fig.11. and its description and service efficiency coefficients are given in Table 6 and Table 5, respectively.

Table 5. η_f for the diagram in Figure 6.

symptom/object f	q_f	t_f
s1	0.4	0.001
o1	0.17	0.45
s2	0.9	0.001
s3	0.008	0.15
s4	0.008	0.46
s5	0.02	0.05
s6	0.45	0.001
s7	0.85	0.035
s8	0.019	0.08
s9	0.18	0.08
s10	0.50	0.05
s11	0.008	0.15
o2	0.06	0.2
o3	0.17	0.15
o4	0.06	0.20
o5	0.008	0.48
o6	0.17	0.25
o7	0.06	0.20
A	0.01	0.7
B	0.01	0.7

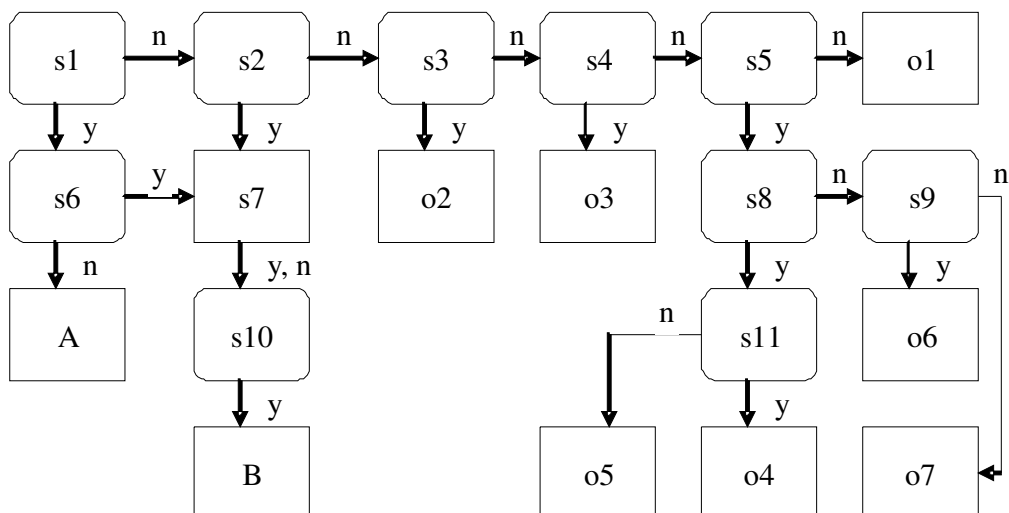


Fig.11. TV set fault location

Table 5. Graph description for Fig.11.

graph edge	Meaning
s1 s6 y	S1; Did the STANDBY diode light up once for 1 second with full light?
s1 s2 n	S2; Does the STANDBY diode light up continuously?
s2 s7 y	O2; Check and possibly replace D6582, C2651.
s2 s3 n	S3; Is the power supply sampling?
s3 o2 y	S4; Is there voltage at the secondary winding of the power converter?
s3 s4 n	O3; Check and possibly replace IC7600 in the control block A7.
s4 o3 y	S5; Is there the impulse P1 in the high voltage supply?
s4 s5 n	O1; Check and possibly replace SW0231, D6505, R3505 in A1.
s5 s8 y	S6; Are video and audio signals correct?
s5 o1 n	S7; Make sure that remote controller communicates with the TV set.
s6 s7 y	S8; Is there the impulse P2 in the high voltage supply?
s6 A n	S9; Is there the impulse P3 in the high voltage supply?
s7 s10 n	s10; Is there any defect recorded in the EM?
s10 B y	S11; Is there voltage at node 9 of TR5545?
s8 s9 n	O4; heck and possibly replace L5540, D6540, C2540, L5573, D6560 in the high voltage supply.
s8 s11 y	O5; Check and possibly replace TS7518,L5552 in the high voltage supply
s9 o6 y	O6; Check and possibly replace ic7520, R3525 in the high voltage supply.
s9 o7 n	O7; Check and possibly replace 40,C2540,D6510,R3510,R3529 in the high voltage supply.
s11 o4 y	A; Perform checks according schedule A.
s11 o5 n	B; Perform checks according schedule B.

The shortest ant tour simulation was performed for the following ant algorithm parameters:

- Pheromone evaporation coefficient = 0.7,
- ALFA coefficient = 1,
- BETA coefficient = 1.
- 100 ants.

Table 6. The ant shortest tour simulation result for the graph in Fig.11.

cycles	fault location	tour length
10 - 30	s1 s2 s7 s10 b	70.142288
40	s1 s6 a	70.002223
50 - 80	s1 s2 s3 o2	22.084444

From Table 6 one can conclude that the shortest fault location tour is s1 - s2 - s3 - o2 which corresponds to following rules:

- If s1 is incorrect then check s2
- If s2 is incorrect then check s3
- If s3 is incorrect then check o2.

If these rules don't apply to the current system state one needs to repeat the process taking into account the updated symptom states.

Further, examinations were conducted to find such settings of the algorithm parameters owing to which the shortest (the most optimal) way to the solution is found.

The following method to find the optimal settings of the parameters was chosen.

Default values for each of the four parameters were introductorily assumed:

- Quantity (doze) of the pheromone trace left = 1,
- Pheromone evaporation coefficient = 0.7,
- ALFA coefficient = 1,
- BETA coefficient = 1.

Next, the values of one parameter were swapped and it was noted down in which cycle the most optimal route was found (during the examinations it was discovered that this is the following route: **S1 -> S2 -> S3 -> O2**, whose total of the visibility coefficients of the symptoms examined and of the object is 22.084444. For the given settings, the examinations were repeated ten times and the average was calculated. Next, in the scope of the changes of a given parameter, such a value was searched for at which the shortest route was founded the fastest. This is presented in the diagrams shown after a series of examinations of every parameter. The following values of individual parameters were determined, for which examinations were conducted:

- Quantity (doze) of the pheromone trace left = 0.5, 1, 1.5, 2, 2.5, 3, 4, 5;
- Coefficient of pheromone evaporation = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8;
- ALFA coefficient = 0, 0.5, 1, 1.5, 2, 2.5, 3, 4, 5;
- BETA coefficient = 0, 0.5, 1, 1.5, 2, 2.5, 3, 4, 5.

4. CONCLUDING REMARKS

The paper demonstrates that ant algorithms can be applied to electronic system diagnosis. The simulation results show that the developed representation knowledge is useful to code a decision graph. For the presented example one needs over 50 - 80 cycles to obtain reasonable results.

Optimal settings of parameters for the Ant Quantitative Algorithm for the practical example examined are as follows:

- doze of pheromone left = 4;
- pheromone evaporation coefficient = 0.6;
- ALFA coefficient = 0.5 ;
- BETA coefficient = 0.5.

The other used types of AA(AD-Density, AC-Cyclic) gave worse results in diagnosis of the same TV set.

The ant algorithms presented here use objective information such as fault and service activity probability duration. It allows to minimize a heuristic knowledge impact on the results obtained.

REFERENCES

- [1] A. Coloni, M. Dorigo, V. Maniezzo.: “*An investigation of some properties of an “Ant Algorithm”*”, Proc. of the Parallel Problem Solving from Nature Conference, Belgium 1992, pp. 509-520.
- [2] M.Dorigo, M. Birattari.: “*Ant Colony Optimization – Artificial Ants as a Computational Intelligence Technique*”. IEEE, Computational Intelligence Magazine, 2006.
- [3] C. Blum.: “*Ant Colony Optimization: Introduction and recent trends*”, Physics of Life Reviews, 2(4):353 -374, 2005.
- [4] D.Merkle, M. Middendorf.: “*Modelling the Dynamics of Ant Colony Optimization Algorithms*”, Evolutionary Computation, 10(3) pp.235 – 262, 2002.
- [5] J.Drabarek, R. Wirski.: “*A Representation Language for A Neural Expert System Applied to Diagnostics*”, International Conference on Signals and Electronic Systems, October2000, Ustroń, Poland.
- [6] J. Drabarek, R. Wirski.: „*A Neural Expert System Concept Applied to Diagnostics*”, 8th IEEE ICECS, September, Malta 2001.
- [7] J. Drabarek, R. Wirski, W. Madej.: „*Genetic Algorithms Applied to Hybrid Expert Systems*”, 9th IEEE ICECS, September Dubrovnik, Croatia 2002.

Ilościowe algorytmy mrówkowe stosowane do diagnostyki systemów elektronicznych