BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **9**, No 2

Sofia • 2009

Intuitionistic Fuzzy Estimation of the Ant Methodology

S. Fidanova, P. Marinov

Institute of Parallel Processing, 1113 Sofia

E-mails: stefka@parallel.bas.bg pencho@parallel.bas.bg

Abstract: *Ant Colony Optimization is a stochastic search method that mimic the social behavior of real ants colonies, which manage to establish the shortest routs to feeding sources and back. Such algorithms have been developed to arrive at near-optimum solutions to large-scale optimization problems, for which traditional mathematical techniques may fail. In this paper a generalized net model of the process of ant colony optimization is constructed and on each iteration intuitionistic fuzzy estimations (see [3]) are made of the start nodes of the ants. Several start strategies are prepared and combined. The study presents ideas that should be beneficial to both practitioners and researchers involved in solving optimization problems.*

Keywords: *Ant colony optimization, Metaheuristics, Discrete optimization, Generalized nets, Intuitionistic fuzzy estimation.*

1. Introduction

The difficulties associated with using mathematical optimization on large-scale engineering problems, have contributed to the development of alternative solutions. Linear programming and dynamic programming techniques, for example, often fail in solving NP-hard problems with large number of variables. To overcome these problems, researchers have proposed mataheuristic methods for searching near-optimal solutions to problems. One of the most successful metaheuristic is Ant Colony Optimization (ACO).

Real ants foraging for food lay down quantities of pheromone (chemical cues) marking the path that they follow. An isolated ant moves essentially at random but an ant encountering a previously laid pheromone will detect it and decide to follow it with high probability and thereby reinforce it with a further quantity of pheromone. The repetition of the above mechanism represents the auto-catalytic behavior of a real ant colony where the more the ants follow a trail, the more attractive that trail becomes. ACO is inspired by real ant behavior to solve hard combinatorial optimization problems. Examples of hard optimization problems are Traveling Salesman Problem [11], Vehicle Routing [12], Minimum Spanning Tree [10], Constrain Satisfaction [8], Knapsack Problem [6], etc. The ACO algorithm uses a colony of artificial ants that behave as cooperative agents in a mathematical space where they are allowed to search and reinforce pathways (solutions) in order to find the optimal ones. The problem is represented by graph and the ants walk on the graph to construct solutions. The solutions are represented by paths in the graph. After the initialization of the pheromone trails, the ants construct feasible solutions, starting from random nodes, and then the pheromone trails are updated. At each step the ants compute a set of feasible moves and select the best one (according to some probabilistic rules) to continue the rest of the tour. The structure of the ACO algorithm is shown by the pseudocode below. The transition probability $p_{i,j}$, to choose the node j when the current node is i, is based on the heuristic information $\eta_{i,j}$ and the pheromone trail level $\tau_{i,j}$ of the move, where $i, j = 1, \ldots, n$.

$$p_{i,j} = \frac{\tau_{i,j}^{a}\eta_{i,j}^{b}}{\sum\limits_{k\in\text{Unused}} \tau_{i,k}^{a}\eta_{i,k}^{b}},$$

where "Unused" is the set of unused nodes of the graph.

The higher the value of the pheromone and the heuristic information, the more profitable it is to select this move and resume the search. In the beginning, the initial pheromone level is set to a small positive constant value τ_0; later, the ants update this value after completing the construction stage. ACO algorithms adopt different criteria to update the pheromone level.

```
Ant Colony Optimization
Initialize number of ants;
Initialize the ACO parameters;
while not end-condition do
    for k=0 to number of ants
        ant k choses start node;
        while solution is not constructed do
            ant k selects higher probability node;
        end while
    end for
    Update-pheromone-trails;
end while
```

Fig. 1. Pseudocode for ACO

The pheromone trail update rule is given by:

$$\tau_{i,j} \leftarrow \rho\tau_{i,j} + \Delta\tau_{i,j},$$

where ρ models evaporation in the nature and $\Delta\tau_{i,j}$ is new added pheromone which is proportional to the quality of the solution.

The novelty in this work is the use of Intuitionistic Fuzzy Estimations (IFE, see [3]) of start nodes with respect to the quality of the solution and thus to better menage the search process. Various start strategies and their combinations are offered.

2. Short remarks on GN theory

At the beginning will be presented the main definitions related with GN. It is generalization of Petri nets. A GN is shown in Fig. 2. Its *places* are marked with circles. Each vertical part of the net is called *transition*. GNs, like other nets, contain *tokens* which are transfered from place to place. Every token enters the net with an initial characteristic. During each transfer, the token receives new characteristics. So, they accumulate their "history". Every GN-place has at most one arc entering and at most one arc leaving it. The places with no entering arcs are called *input places* for the net and those with no leaving arcs are called *output places*. A transition may contain several input and several output places, the number of input places can be different from the number of output places.

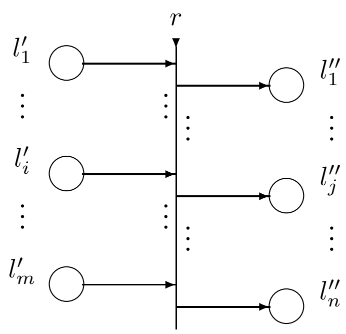

Fig. 2. Generalized Net

Following [1, 2, 4], will be mentioned that every GN-transition is described by a seven-tuple (Fig. 2.)

$$Z = \langle L', L'', t_1, t_2, r, M, \square \rangle,$$

where:

(a) L' and L'' are finite, non-empty sets of places (the transition's input and output places, respectively); for the transition in Fig. 2. these are $L' = \{l'_1, l'_2, \dots, l'_m\}$ and $L'' = \{l''_1, l''_2, \dots, l''_n\}$;

(b) t_1 is the current time-moment of the transition's firing;

(c) t_2 is the current value of the duration of its active state;

(d) r is the transition's *condition* determining which tokens will pass (or *transfer*) from the transition's inputs to its outputs; it has the form of an Index Matrix (IM, see

[2,4]):

$$r = \begin{array}{c|c} & l''_1 \ \dots \ l''_j \ \dots \ l''_n \\ \hline l'_1 & \\ \vdots & r_{i,j} \\ l'_i & (r_{i,j} \ - \ \text{predicate}\ ) \\ \vdots & (1 \leq i \leq m, 1 \leq j \leq n) \\ l'_m & \end{array} \quad ;$$

$r_{i,j}$ is the predicate which corresponds to the i-th input and j-th output places. When its truth value is "$true$", a token from i-th input place can be transferred to j-th output place; otherwise, this is not possible;

(e) M is an IM of the capacities of transition's arcs:

$$M = \begin{array}{c|c} & l''_1 \ \dots \ l''_j \ \dots \ l''_n \\ \hline l'_1 & \\ \vdots & m_{i,j} \\ l'_i & (m_{i,j} \geq 0 - \ \text{natural number}\ ) \\ \vdots & (1 \leq i \leq m, 1 \leq j \leq n) \\ l'_m & \end{array} \quad ;$$

(f) $\square$ is an object having a form similar to a Boolean expression. It may contain as variables the symbols which serve as labels for transition's input places, and is an expression built up of variables and the Boolean connectives $\wedge$ and $\vee$ whose semantics is defined as follows:

$\wedge(l_{i_1}, l_{i_2}, \dots, l_{i_u})$	–	every place l_{i_1}, l_{i_2}, $\dots$, l_{i_u} must contain at least one token,
$\vee(l_{i_1}, l_{i_2}, \dots, l_{i_u})$	–	there must be at least one token in all places l_{i_1}, l_{i_2},$\dots$, l_{i_u}, where $\{l_{i_1}, l_{i_2}, \dots, l_{i_u}\} \subset L'$.

When the value of a type (calculated as a Boolean expression) is "$true$", the transition can become active, otherwise it cannot.

The ordered four-tuple

$$E = \langle\langle A, \pi_A, \pi_L, c, f, \theta_1, \theta_2\rangle, \langle K, \pi_K, \theta_K\rangle,$$

$$\langle T, t^0, t^*\rangle, \langle X, \Phi, b\rangle\rangle$$

is called a *Generalized Net* (GN) if:

(a) A is a set of transitions;

(b) π_A is a function giving the priorities of the transitions, i.e., $\pi_A : A \to N$, where $N = \{0, 1, 2, \dots\} \cup \{\infty\}$;

(c) π_L is a function giving the priorities of the places, i.e., $\pi_L : L \to N$, where $L = pr_1 A \cup pr_2 A$, and $pr_i X$ is the i-th projection of the n-dimensional set, where $n \in N, n \geq 1$, and $1 \leq k \leq n$ (obviously, L is the set of all GN-places);

(d) c is a function giving the capacities of the places, i.e., $c : L \to \ N$;

(e) f is a function which calculates the truth values of the predicates of the transition's conditions (for the GN described here let the function f have the value "$false$" or "$true$", i.e., a value from the set $\{0, 1\}$;

(f) θ_1 is a function giving the next time-moment when a given transition Z can be activated, i.e., $\theta_1(t) = t'$, where $pr_3Z = t, t' \in [T, T + t^*]$ and $t \leq t'$. The value of this function is calculated at the moment when the transition terminates its functioning;

(g) θ_2 is a function giving the duration of the active state of a given transition Z, i.e., $\theta_2(t) = t'$, where $pr_4Z = t \in [T, T + t^*]$ and $t' \geq 0$. The value of this function is calculated at the moment when the transition starts functioning;

(h) K is the set of the GN's tokens. In some cases, it is convenient to consider this set in the form

$$K = \bigcup_{l \in Q^I} K_l ,$$

where K_l is the set of tokens which enter the net from place l, and Q^I is the set of all input places of the net;

(i) π_K is a function giving the priorities of the tokens, i.e., $\pi_K \colon K \to N$;

(j) θ_K is a function giving the time-moment when a given token can enter the net, i.e., $\theta_K(\alpha) = t$, where $\alpha \in K$ and $t \in [T, T + t^*]$;

(k) T is the time-moment when the GN starts functioning. This moment is determined with respect to a fixed (global) time-scale;

(l) t^0 is an elementary time-step, related to the fixed (global) time-scale;

(m) t^* is the duration of the GN functioning;

(n) X is the set of all initial characteristics the tokens can receive when they enter the net;

(o) Φ is a characteristic function which assigns new characteristics to every token when it makes a transfer from an input to an output place of a given transition.

(p) b is a function giving the maximum number of characteristics a given token can receive, i.e., $b : K \to N$.

For example, if $b(\alpha) = 1$ for some token α, then this token will enter the net with some initial characteristic (marked as its zero-characteristic) and subsequently it will keep only its current characteristic.

When $b(\alpha) = \infty$, the token α will keep all its characteristics. When $b(\alpha) = k < \infty$, except its zero-characteristic, the token α will keep its last k characteristics (characteristics older than the last k will be "forgotten"). Hence, in the general case, every token α has $b(\alpha) + 1$ characteristics when it leaves the net.

A GN may lack some of the components, and such GNs give rise to special classes of GNs called *reduced GNs*. The omitted elements of the reduced GNs are marked by "$*$".

Different operations and relations are defined over the transitions of the GNs and over the same nets.

The idea of defining operators over the set of GNs in the form suggested below dates back to 1982.

A variety of different types of GN-extensions are defined and each of them is proved [2, 4] to be a conservative extension of the ordinary GNs.

3. Generalized net description of ACO

In this paper the ACO method is described by Generalized Net (GN) to can deeply understand the processes and to improve them.

The present GN is an extension of the GN from [7]. All notations from [7] are kept so the reader may easily compare both models. Let the graph of the problem have m nodes. The set of nodes is divided in N subsets. There are different ways for dividing. Normally, the nodes of the graph are randomly enumerated. An example for creation of the subsets, without loss of generality, is: node number one goes to the first subset, node number two - to the second subset, etc., node number N is in the N-th subset, node number $N+1$ is in the first subset, etc. Thus the number of nodes in the separate subsets are almost equal.

The GN in this work has 4 transitions, 20 places ($l_1, ..., l_{20}$) and four types (β, γ, ϵ, and δ) of tokens (see Fig. 3.).

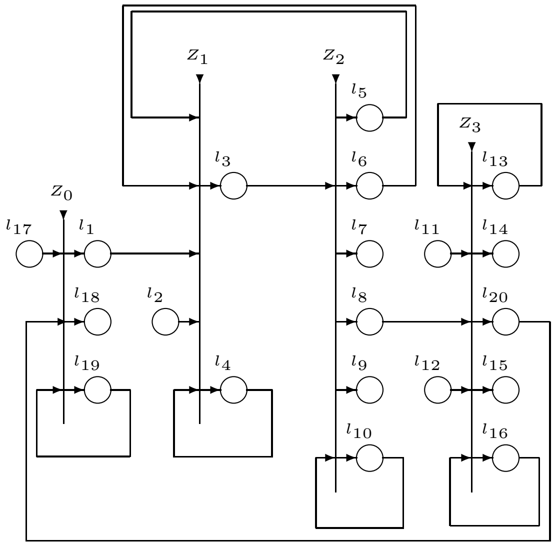

Fig. 3. GN net model for ACO

These tokens enter, respectively, places: l_2 – with the initial characteristic: "$\langle$ m-dimensional vector of heuristics with elements – the graph vertices or l-dimensional vector of heuristics with elements – the graph arcs; objective function $\rangle$",
where m is the number of the nods of the graph of the problem and l is the number of the arcs of the graph;

l_{11} – with the initial characteristic: "the graph structure with m vertices's and l arcs".

l_{12} – with the initial characteristic: "initial data for the places and quantities of the pheromones";

l_{17} – with the initial characteristic: "$\langle$ values of parameters A and B; number n of the ants; number N of the subsets of the nodes of the graph $\rangle$".

$$Z_0 =< \{l_{17}, l_{19}, l_{20}\}, \{l_1, l_{18}, l_{19}\},$$

$$\begin{array}{c|ccc} & l_1 & l_{18} & l_{19} \\ \hline l_{17} & false & false & true \\ l_{19} & W_{19,1} & W_{19,18} & W_{19,19} \\ l_{20} & false & false & true \end{array} >,$$

where:

$W_{19,1}$ ="the present is the second time-moment of GN-functioning",

$W_{19,18}$ =“truth-value of expression $C_1 \vee C_2 \vee C_3$ is $true$”,

$W_{19,18} = \neg W_{19,19}$,

and C_1, C_2 and C_3 are the following end-conditions:

C_1 – “computational time (maximal number of iterations) is achieved”,

C_2 – “number of iterations without improving the result is achieved”,

C_3 – “if the upper/lower bound is known, then the current results are close (e.g., less than 5%) to the bound”.

Token δ from place l_{17} enters place l_{19} with a characteristic: “$\{\langle\langle j, 1\rangle, D_j(1), E_j(1)\rangle | 1 \leq j \leq N\}$”, where the initial values of these coefficients are: $D_j(1) = 1, E_j(1) = 0$.

On the second time-moment token δ splits to two tokens: α that enters place l_1 with a characteristic: “n-dimensional vector with elements – the couples of the ants coordinates", where n is the number of used ants and it is determined in the initial δ-token characteristic; and token δ that does not obtains any characteristic.

After the first iteration, when token β^* enters place l_{19} from place l_{20}, token δ unites with token β^* and obtains characteristic: “ $\{\langle j, D_j(i), E_j(i)\rangle | 1 \leq j \leq N\}$", where $i \geq 2$ is the number of the current iteration and $D_j(i)$ and $E_j(i)$ are weight coefficients of j-th nodes subset ($1 \leq j \leq N$) and they are calculated by following formulas:

$$D_j(i) = \frac{i.D_j(i-1) + F_j(i)}{i},$$

$$E_j(i) = \frac{i.E_j(i-1) + G_j(i)}{i},$$

where $i \geq 1$ is the current process iteration and for each j $(1 \leq j \leq N)$:

$$(1) \qquad F_j(i) = \begin{cases} \frac{f_{j,A}}{n_j} & \text{if } n_j \neq 0 \\ F_j(i-1) & \text{otherwise} \end{cases},$$

$$(2) \qquad G_j(i) = \begin{cases} \frac{g_{j,B}}{n_j} & \text{if } n_j \neq 0 \\ G_j(i-1) & \text{otherwise} \end{cases},$$

and $f_{j,A}$ is the number of the solutions among the best $A\%$, and $g_{j,B}$ is the number of the solutions among the worst $B\%$, where $A + B \leq 100$, $i \geq 1$ and $\sum_{j=1}^{N} n_j = n$,, where n_j $(1 \leq j \leq N)$ is the number of solutions obtained by ants starting from nodes subset j.

When $W_{19,18} = true$, token δ leaves the net through place l_{18} without any characteristic.

$$Z_1 =< \{l_1, l_2, l_4, l_5, l_6\}, \{l_3, l_4\}, \begin{array}{c|cc} & l_3 & l_4 \\ \hline l_1 & true & false \\ l_2 & false & true \\ l_4 & false & true \\ l_5 & true & false \\ l_6 & true & false \end{array} > .$$

Token α from places l_1, l_5 or l_6 enters place l_3 with a characteristic: "vector of current transition function results $\langle \phi_{1,cu}, \phi_{2,cu}, ..., \phi_{n,cu} \rangle$", while token ϵ stays only in place l_4 obtaining the characteristic: "new m-dimensional vector of heuristics with elements – the graph vertices's or, new l-dimensional vector of heuristics with elements – the graph arcs".

$$Z_2 =< \{l_3, l_{10}\}, \{l_5, l_6, l_7, l_8, l_9, l_{10}\},$$

$$\begin{array}{c|cccccc} & l_5 & l_6 & l_7 & l_8 & l_9 & l_{10} \\ \hline l_3 & W_{3,5} & W_{3,6} & W_{3,7} & false & W_{3,9} & W_{3,10} \\ l_{10} & false & false & true & false & W_{10,9} & W_{10,10} \end{array} >,$$

where
$W_{3,5} =$ "the current iteration is not finishe",
$W_{3,6} = W_{3,10} = \neg W_{3,5} \vee \neg W_{10,9}$,
$W_{10,7} =$ "the current best solution is worse than the global best solution",
$W_{10,9} =$"truth-value of expression $C_1 \vee C_2 \vee C_3$ is $true$",
$W_{10,10} = \neg W_{10,9}$.

Token α from place l_3 enters place l_5 with a characteristic: "$\langle S_{1,cu}, S_{2,cu}, \dots, S_{n,cu} \rangle''$, where $S_{k,cu}$ is the current partial solution for the current iteration, made by k-th ant ($1 \leq k \leq n$).

If $W_{3,6} = true$ it splits to three tokens α, α' and α'' that enter places l_6 – token α – with a characteristic: "new n-dimensional vector with elements – the couples of the new ants coordinates", place l_8 – token α' – with the last α-characteristic, and place l_{10} – token α'' – with a characteristic: "$\langle$ the best solution for the current iteration; its number $\rangle$". Token α'' can enter place l_9 only when $W_{10,9} = true$ and there it obtains the characteristic: "the best achieved result".

In place l_7 one of the two tokens from place l_{10} enters, which has the worst values as a current characteristic, while in place l_{10} the token containing the best values as a current characteristic stays.

$$Z_3 =< \{l_8, l_{11}, l_{12}, l_{13}, l_{16}\}, \{l_{13}, l_{14}, l_{15}, l_{16}, l_{20}\},$$

$$\begin{array}{c|ccccc} & l_{13} & l_{14} & l_{15} & l_{16} & l_{20} \\ \hline l_8 & false & false & false & true & false \\ l_{11} & true & false & false & false & false \\ l_{12} & false & false & false & true & true \\ l_{13} & W_{13,13} & W_{13,14} & false & false & false \\ l_{16} & false & false & false & false & W_{16,20} \end{array} >,$$

where
$W_{13,14} = W_{16,15} =$"truth-value of expression $C_1 \vee C_2 \vee C_3$ is $true$",
$W_{13,13} = W_{16,16} = \neg W_{13,14}$,
$W_{16,20} =$ "the current iteration is finished" $\& \neg W_{13,14}$.

Tokens γ from place l_{11} and β from place l_{12} with above mentioned characteristics enter, respectively, places l_{13} and l_{16} without any characteristic.

Token α from place l_8 enters place l_{16} and unites with token β (the new token is again β) with characteristic: "value of the pheromone updating function with respect of the values of the objective function".

Tokens β and γ enter, respectively, places l_{14} and l_{15} without any characteristics. When $W_{16,20} = true$, token β splits to two token: β that continue to stay in place l_{16} without a new characteristic and token β^* that enters place l_{20} with characteristic: "$\{\langle j, F_j(i), G_j(i) \rangle | 1 \leq j \leq N\}''$.

4. Start strategies

The known ACO algorithms create a solution starting from random node. But for some problems, especially subset problems, it is important from which node the search process starts. For example if an ant starts from node which does not belong to the optimal solution, probability to construct it is zero. In this paper is offered several start strategies. The aim is to use the experience of the ants from previous iteration to choose the better starting node. Other authors use this experience only by the pheromone, when the ants construct the solutions. Let threshold E for $E_j(i)$ and D for $D_j(i)$ be fixed, than several strategies to choose start node for every ant are constructed, the threshold E increase every iteration with $1/i$ where i is the number of the current iteration:

1. If $\dfrac{E_j(i)}{D_j(i)} > E$ then the subset j is forbidden for current iteration and the starting node is chosen randomly from $\{j \mid j \text{ is not forbidden}\}$.

2. If $\dfrac{E_j(i)}{D_j(i)} > E$ then the subset j is forbidden for current simulation and the starting node is chosen randomly from $\{j \mid j \text{ is not forbidden}\}$.

3. If $\dfrac{E_j(i)}{D_j(i)} > E$ then the subset j is forbidden for K_1 consecutive iterations and the starting node is chosen randomly from $\{j \mid j \text{ is not forbidden}\}$.

4. Let $r_1 \in [R, 1)$ is a random number. Let $r_2 \in [0, 1]$ is a random number. If $r_2 > r_1$ a node is chosen randomly from subset $\{j \mid D_j(i) > D\}$, otherwise a node is chosen randomly from the not forbidden subsets, R is chosen and fixed at the beginning.

5. Let $r_1 \in [R, 1)$ is a random number. Let $r_2 \in [0, 1]$ is a random number. If $r_2 > r_1$ a node is randomly chosen from subset $\{j \mid D_j(i) > D\}$, otherwise a node is randomly chosen from the not forbidden subsets, R is chosen at the beginning and increase with r_3 every iteration.

Where $0 \leq K_1 \leq$ “number of iterations” is a parameter. If $K_1 = 0$, thån strategy 3 is equal to the random choose of the start node. If $K_1 = 1$, thån strategy 3 is equal to strategy 1. If $K_1 =$“maximal number of iterations”, than strategy 3 is equal to the strategy 2. The strategies 1, 2 and 3 can be called forbid strategies, and strategies 4 and 5 can be called stimulate strategies. By stimulate strategies the ants are forced to start there search from subsets with high value of $D_j(i)$. If $R = 0.5$, than the probability an ant to start from nodes subset with high value of $D_j(i)$ is two times high than to start from other subset. For forbidden strategies is used fraction between $E_j(i)$ and $D_j(i)$. Thus is prevented some regions with several bad and with several good solutions to be forbidden.

More than one strategy for choosing the start node can be used, but there are strategies which can not be combined. The strategies are distributed into two sets: St1= Strategy 1, Strategy 2, Strategy 3 and St2= Strategy 4, Strategy 5 . The strategies from same set can not be used at once. Thus it can be used strategy from one set or combine it with strategies from other set. Exemplary combinations are (Strategy1), (Strategy2; Strategy5), (Strategy3; Strategy4).

5. Conclusion

This paper is addressed to the modelling of the process of ant colony optimization method by generalized net using intuitionistic fuzzy estimations, combining five start strategies. So, the start node of each ant depends of the goodness of the respective region. The future work will be focused on parameter settings which manage the starting procedure. It will be investigated on influence of the parameters to algorithm performance. The aim of this representation is to study in detail the methodology and relationships between the processes. Thereby one can see the weaknesses of the method and improve it implementation.

Acknowledgments: This work has been partially funded by the Bulgarian Ministry of Education and Science by the grant VU-MI-204/2006.

R e f e r e n c e s

1. A l e x i e v a, J., E. C h o y, E. K o y c h e v. Review and Bibloigraphy on Generalized Nets Theory and Applications. – In: E. Choy, M. Krawczak, A. Shannon and E. Szmidt, Eds. A Survey of Generalized Nets. Raffles KvB Monograph **10**, 2007, 207-301.
2. A t a n a s s o v, K. Generalized Nets. Singapore. World Scientific, 1991.
3. A t a n a s s o v, K. Intuitionistic Fuzzy Sets. Heidelberg, Springer,1999.
4. A t a n a s s o v, K. On Generalized Nets Theory. Sofia, Prof. M. Drinov Academic Publishing House, 2007.
5. D o r i g o, M., L. M. G a m b a r d e l l a. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. – IEEE Transactions on Evolutionary Computation **1**, 1997, 53-66.
6. F i d a n o v a, S. Evolutionary Algorithm for Multiple Knapsack Problem. – In: Int. Conference Parallel Problems Solving from Nature, Real World Optimization Using Evolutionary Computing, 2002, Granada, Spain.
7. F i d a n o v a, S., K. A t a n a s s o v. Generalized Net Model of the Process of Ant Colony Optimization, Issues on Intuitionistic Fuzzy Sets and Generalized Nets **7**, 2008, 108-114.
8. L e s s i n g, L., I. D u m i t r e s c u , T. S t u t z l e. A Comparison between ACO Algorithms for the Set Covering Problem. –In: ANTS Workshop, 2004 , 1-12.
9. M a r t e l l o, S., P. T o t h. A Mixtures of Dynamic Programming and Branch-and-Bbound for the Subset-Sum Problem. – Management Science, **30**, 1984, 756-771.
10. R e i m a n, M., M. L a u m a n n s. A Hybrid ACO Algorithm for the Capacitate Minimum Spanning Tree Problem.–In: Workshop on Hybrid Metahuristics, Valencia, Spain, 2004, 1-10.
11. S t u t z l e, T., M. D o r i g o. ACO Algorithm for the Traveling Salesman Problem. – In: K. Miettinen, M. Makela,P. Neittaanmaki, J. Periaux, Eds. Evolutionary Algorithms in Engineering and Computer Science, Wiley, 1999, 163-183.
12. Z h a n g, T., S. W a n g, W. T i a n, Y. Z h a n g, ACO-VRPTWRV: A New Algorithm for the Vehicle Routing Problems with Time Windows and Re-Used Vehicles Based on Ant Colony Optimization. –In: Conference on Intelligent Systems Design and Applications, IEEE Press, 2006, 390-395.