

# A generalized net with an ACO-algorithm optimization component

Vassia Atanasova<sup>1</sup>, Stefka Fidanova<sup>1</sup>, Panagiotis Chountas<sup>2</sup>, Krassimir Atanasov<sup>3</sup>

<sup>1</sup> Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, Acad. G. Bonchev Str., bl. 2, Sofia 1113, Bulgaria  
`vassia.atanasova@gmail.com`, `stefka@parallel.bas.bg`

<sup>2</sup> University of Westminster, Dept. of Business Information Systems, School of Electronics & Computer Science, 115 New Cavendish Street, London W1W 6UW, United Kingdom  
`p.i.chountas@westminster.ac.uk`

<sup>3</sup> Bioinformatics and Mathematical Modelling Dept., Institute of Biophysics and Biomedical Engineering, Bulgarian Academy of Sciences, Acad. G. Bonchev Str., bl. 105, Sofia 1113, Bulgaria  
`krat@bas.bg`

**Abstract.** In the paper we describe a generalized net  $G_{ACO}$  realizing an arbitrary algorithms for ant colony optimization. In this sense, this net is universal for all standard algorithms for ant colony optimization, since it describes the way of functioning and results of their work. Then, we discuss the way of constructing a GN that includes the  $G_{ACO}$  as a subnet. In this way, we ensure the generalized net tokens' optimal transfer with regard to the results of  $G_{ACO}$ . Thus, we construct a generalized net, featuring an optimization component and thus optimally functioning.

## 1 Introduction

Generalized nets (GNs; see [1, 3, 5]) are an apparatus for modelling of parallel and concurrent processes, developed as an extension of the concept of Petri nets and some of their modifications. During the last 25 years it was shown that the GNs can be used for constructing of universal tools, describing the functioning and the result of the work of the other types of Petri nets, of the finite automata and Turing machine, of expert systems and machine learning processes, data bases and data warehouses, etc. (see, e.g. [2, 4, 6, 11]).

In a series of papers by some of the authors, it was shown that the GNs can represent the functioning and the result of the work of different Ant Colony Optimization (ACO) algorithms (see, e.g. [7–9]). On the other hand, in [1] it was shown that we can construct special types of GNs, featuring an optimization component. By optimization component we will understand a subnet of a given generalized net, which describes a particular optimization problem and whose

results from functioning can be used in the main net with the purpose of optimizing its behaviour. So far, such optimization components have been defined for the Transportational Problem and the Travelling Salesman Problem [1, ?].

Here, for the first time it will be proposed an optimization problem, realizing the ACO-algorithm. This component guarantees that the GN functions optimally with respect of the results from the ACO-algorithm performance.

## 2 Short remarks on GNs

Formally, every GN-transition is described by a seven-tuple:

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

where:

- (a)  $L'$  and  $L''$  are finite, non-empty sets of places (the transition's input places (*inputs*) and output places (*outputs*), respectively).
- (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 [1, 3]):

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

$r_{i,j}$  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, its elements being natural numbers corresponding to the number of tokens that may transfer through the transition at a time (see [5]);

- (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 from variables and the Boolean connectives  $\wedge$  and  $\vee$ . 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

$$E = \langle \langle A, \pi_A, \pi_L, c, f, \theta_1, \theta_2 \rangle, \langle K, \pi_K, \theta_K \rangle, \langle T, t^o, t^* \rangle, \langle X, \Phi, b \rangle \rangle$$

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

- (a)  $A$  is a set of transitions;

- (b)  $\pi_A$  is a function giving the priorities of the transitions, i.e.,  $\pi_A : A \rightarrow N$ , where  $N = \{0, 1, 2, \dots\} \cup \{\infty\}$ ;
- (c)  $\pi_L$  is a function giving the priorities of the places, i.e.,  $\pi_L : L \rightarrow N$ , where  $L = pr_1A \cup pr_2A$ , and  $pr_iX$  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.,  $c : L \rightarrow N$ ;
- (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)  $\theta_1$  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)  $\theta_2$  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;
- (i)  $\pi_K$  is a function giving the priorities of the tokens, i.e.,  $\pi_K : K \rightarrow N$ ;
- (j)  $\theta_K$  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)  $t^o$  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)  $\Phi$  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 particular token can receive, i.e.,  $b : K \rightarrow N$ .

When the GN has only a part of the above components, it is called reduced GN. Below we shall use a reduced GN without the temporal components characterising the net or the transition, without tokens', places' and transitions' priorities, as well as without arcs' and places' capacities.

In [1, 3] different operations, relations and operators are defined over GNs. One of them, namely  $\cup$ , can merge two given GN-models.

### 3 A GN universal for the ACO-algorithms

Following [7], we shall construct the GN that is universal for the ACO-algorithms. This GN will describe how a set of  $n$  artificial ants move along the ACO-algorithm graph and let this graph consist of  $m$  vertices and  $k$  arcs. Let us assume that any quantity of pheromone left in the graph is known in advance.

Let us denote the so described GN by  $G_{ACOA}$ . It has 3 transitions, 16 places (see Fig. 1) and four types ( $\alpha, \beta, \gamma$  and  $\varepsilon$ ) of tokens. The  $\alpha$  token is designed

to contain information about the ants' location within the graph. The  $\beta$  token obtains the values of the number of vertices ( $m$ ) and the number of arcs ( $k$ ). The  $\gamma$  token contains the structure of the graph, i.e. a matrix of incidence, while the  $\epsilon$  token contains the information about the vertices/arcs where pheromone has been laid by the ants, and the quantities of this pheromone. All these tokens enter the input places of the GN with certain initial characteristics, that are explained below.

Token  $\alpha$  enters place  $l_1$  with the initial characteristic “The  $n$ -dimensional vector with elements being the ants' locations within the graph”.

The  $\beta$  token enters place  $l_2$  with the initial characteristic

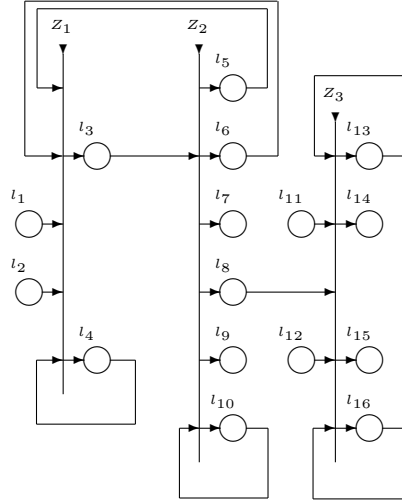
“ $\langle m$ -dimensional vector with elements – the graph vertices  
or  $l$ -dimensional vector with elements – the graph arcs;  
objective function  $\rangle$ ”,

where  $m$  is the number of the nodes 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 and  $l$  arcs”;

$l_{12}$  – with the initial characteristic

“initial data for the places and quantities of the pheromones”.



**Fig. 1.** GN net model for ACO

$$Z_1 = \langle \{l_1, l_2, l_4, l_5, l_6\}, \{l_3, l_4\},$$

	$l_3$	$l_4$
$l_1$	<i>true</i>	<i>false</i>
$l_2$	<i>false</i>	<i>true</i>
$l_4$	<i>false</i>	<i>true</i>
$l_5$	<i>true</i>	<i>false</i>
$l_6$	<i>true</i>	<i>false</i>

$$> .$$

Token  $\alpha$ , occupying either place  $l_1$ , or place  $l_5$ , or place  $l_6$  enters place  $l_3$  with a characteristic “*Vector of current transition function results  $\langle \varphi_{1,cu}, \varphi_{2,cu}, \dots, \varphi_{n,cu} \rangle$* ”, while token  $\varepsilon$  remains looping in place  $l_4$  obtaining the characteristic

“new  $m$ -dimensional vector with elements – the graph vertices,

or new  $l$ -dimensional vector with elements – the graph arcs”.

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

	$l_5$	$l_6$	$l_7$	$l_8$	$l_9$	$l_{10}$
$l_3$	$W_{3,5}$	$W_{3,6}$	$W_{3,7}$	<i>true</i>	$W_{3,9}$	$W_{3,10}$
$l_{10}$	<i>false</i>	<i>false</i>	<i>true</i>	<i>true</i>	$W_{10,9}$	$W_{10,10}$

$$> ,$$

where

$W_{3,5}$  = “The current iteration has not finished”,

$W_{3,6} = W_{3,10} = \neg W_{3,5} \vee \neg W_{10,9}$ ,

$W_{10,7}$  = “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*”,

$W_{10,10} = \neg W_{10,9}$ ,

where  $C_1, C_2$  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  $\alpha$  from place  $l_3$  enters place  $l_5$  with a characteristic

$$\langle S_{1,cu}, S_{2,cu}, \dots, S_{n,cu} \rangle ,$$

where  $S_{i,cu}$  is the current partial solution for the current iteration, made by the  $i$ -th ant ( $1 \leq i \leq n$ ).

If  $W_{3,6} = \text{true}$ , then token  $\alpha$  splits to three tokens  $\alpha, \alpha'$  and  $\alpha''$ .

- Token  $\alpha$  enters place  $l_6$  with a current characteristic “*New  $n$ -dimensional vector with elements being the new ants’ locations*”.
- Token  $\alpha'$  enters place  $l_8$  with the last  $\alpha$ ’s characteristic (before the splitting).
- Token  $\alpha''$  enters place  $l_{10}$  with a current characteristic being the pair “*The best solution for the current iteration; its number*”.

When  $W_{10,9} = \text{true}$  token  $\alpha''$  can enter place  $l_9$  where it obtains the characteristic “*The best achieved result*”.

In place  $l_7$  one of both tokens from place  $l_{10}$  enters, which has worse; the worse values as a current characteristic, while in place  $l_{10}$  the token, containing the best values as a current characteristic, is preserved to keep looping.

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

	$l_{13}$	$l_{14}$	$l_{15}$	$l_{16}$	
$l_8$	<i>false</i>	<i>false</i>	<i>false</i>	<i>true</i>	$>$ ,
$l_{11}$	<i>true</i>	<i>false</i>	<i>false</i>	<i>false</i>	
$l_{12}$	<i>false</i>	<i>false</i>	<i>false</i>	<i>true</i>	
$l_{13}$	$W_{13,13}$	$W_{13,14}$	<i>false</i>	<i>false</i>	
$l_{16}$	<i>false</i>	<i>false</i>	$W_{16,15}$	$W_{16,16}$	

where

$W_{13,14} = W_{16,15} = \text{“truth-value of expression } C_1 \vee C_2 \vee C_3 \text{ is true”}$ ,

$W_{13,13} = W_{16,16} = \neg W_{13,14}$ .

Tokens  $\gamma$  from place  $l_{11}$  and  $\beta$  from place  $l_{12}$  with above mentioned characteristics enter, respectively, places  $l_{13}$  and  $l_{16}$  without any characteristic.

Token  $\alpha$  from place  $l_8$  enters place  $l_{16}$  and unites with token  $\beta$  (the new token is again  $\beta$ ) with characteristic “*Value of the pheromone updating function with respect of the values of the objective function*”.

Tokens  $\beta$  and  $\gamma$  enter, respectively, places  $l_{14}$  and  $l_{15}$  without any characteristics.

#### 4 GNs with optimization component $G_{ACO}$

Based on the definition of the concept of a GN, we shall construct a new type of GNs – GNs with optimization components (GNOC).

The tokens in GN-transitions transfer in the GN by the algorithms discussed in [1, 3]. These algorithms are based on the checking of the truth-values of the transition condition predicates.

Let  $E$  be an arbitrary GN and let us desire to control its tokens transfer optimally on the base of solutions obtained by the ACO-algorithm. Now, we can unite (by operation  $\cup$  between two GNs, mentioned above) the GNs  $E$  and  $G_{ACO}$  and in the new GN we can organize the optimal way for tokens transfer.

According to the algorithm for tokens transfer from [3], this transfer is realized at every time-moment of the functioning of the given GN in the frames of one active transition - the so called “abstract transition”, which is a union of all active GN-transitions at this moment.

By this reason, we can describe the functioning of only one GN-transition, and, in particular, the abstract transition.

Let the transition  $Z$  from  $E$  be given. Each of its input places (in the terms of the ACO-algorithm) will correspond to an input vertex of the graph. Let us determine all GN-tokens in the inputs of  $Z$ . Their number will be the initial

characteristic of token  $\alpha$  from  $G_{ACOA}$  (in the form of input vertices of the given graph). The information about the places that contain these tokens will be given as an initial characteristic of token  $\beta$  of the same GN. The information about the graph can be inserted in the GN  $G_{ACOA}$  before the beginning of the process and it will be kept during the whole simulation.

Having in mind the characteristic functions of the output places of  $Z$ , we can determine those of the vertices of the same graph that exhibit sufficiently high weights.

Each token from the  $Z$ -inputs will correspond to an ant that has to search for a path to the vertex with the maximal weight.

The GN-transition components  $t_1, t_2$  and  $M$  are not necessary (for example, we can assume that the values of  $t_1$  are sequential natural numbers,  $t_2 = 1$  and the elements of  $M$  are equal to  $\infty$ ). The transition type has the form of disjunction.

On the basis of the above determined characteristics for the tokens of  $G_{ACOA}$ , these tokens go through the GN and in a result determine the trajectories of the separate ants (and therefore tokens from  $Z$ ) from input vertices of the graph to its vertices with highest weights (that correspond respective to the input and output places of  $Z$ ). So, using the solutions from  $G_{ACOA}$  we can determine the way of transfer of the tokens from input to output places of  $Z$ .

This procedure can be used for control of each one of the transitions of GN  $E$ . In this case, in  $G_{ACOA}$  we will put respective number of  $\alpha$ - and  $\beta$ -tokens, so that their number will correspond to the number of the transitions in  $E$ .

## 5 Conclusion

The described procedure is universal in nature. It is applicable to each one GN. In [1] a GN was described and this net has an optimization component solving a transportation problem. The present model is the second one, but in future similar models can be constructed for other optimization procedures, e.g., traveling salesman problems, knapsack-problem and others.

The herewith proposed procedure can find various applications, for instance in emergency medicine. Following the ideas proposed in [10, 12] it is possible to construct generalized net models of the organization and management of emergency medicine, and using ant colony optimization component to search for optimality in the decision making processes.

## 6 Acknowledgments

The authors would like to acknowledge the support provided by the Bulgarian National Science Fund under Grants Ref. No. DID-02-29 "Modelling Processes with Fixed Development Rules" and DTK-02-44 "Effective Monte Carlo Methods for Large-Scale Scientific Problems". This work has also been partially supported by Royal Society, Grant Ref. No. JP100372 "Generalised net models and intuitionistic fuzzy sets in intelligent systems".

## References

1. Atanassov K.: Generalized Nets, World Scientific, Singapore, New Jersey, London (1991)
2. Atanassov, K.: Generalized Nets in Artificial Intelligence. Vol. 1: Generalized nets and Expert Systems. "Prof. M. Drinov" Academic Publishing House, Sofia (1998)
3. Atanassov, K.: On Generalized Nets Theory. Prof. M. Drinov Publishing House, Sofia (2007)
4. Atanassov, K., Aladjov, H.: Generalized Nets in Artificial Intelligence. Vol. 2: Generalized nets and Machine Learning. "Prof. M. Drinov" Academic Publishing House, Sofia (2001)
5. Atanassova, V.: Generalized nets ([http://ifigenia.org/wiki/Generalized\\_nets](http://ifigenia.org/wiki/Generalized_nets)), Transition (<http://ifigenia.org/wiki/Transition>), Ifigenia, the wiki for intuitionistic fuzzy sets and generalized nets (2009)
6. Chountas, P., Kolev, B., Rogova, E., Tasseva, V., Atanassov, K.: Generalized Nets in Artificial Intelligence. Vol. 4: Generalized nets, Uncertain Data and Knowledge Engineering. "Prof. M. Drinov" Academic Publishing House, Sofia (2007)
7. Fidanova, S., Atanassov, K.: Generalized net models of the process of ant colony optimization with intuitionistic fuzzy estimations. Proceedings of the Ninth International Workshop on Generalized Nets (K. Atanassov and A. Shannon, Eds.), Sofia, 4 July 2008, Vol. 1, 41-48 (2008)
8. Fidanova, S., Atanassov, K.: Generalized net models for the process of hybrid ant colony optimization. Comptes Rendus de l'Academie bulgare des Sciences, Tome 61, No. 12, 1535-1540 (2008)
9. Fidanova, S., Atanassov, K.: Generalized nets as tools for modelling of the ant colony optimization algorithms. Lecture Notes in Computer Science, Vol. 5910, 326-333 (2010)
10. Ibri, S., Drias, H., Nourelfath, M.: Integrated Emergency Vehicle Dispatching and Covering: A Parallel Ant-Tabu Approach. 8th International Conference of Modeling and Simulation (MOSIM'10) May 10-12, 2010, Hammamet, Tunisia, "Evaluation and optimization of innovative production systems of goods and services"
11. Kolev, B., El-Darzi, E., Sotirova, E., Petronias, I., Atanassov, K., Chountas, P., Kodogiannis, V.: Generalized Nets in Artificial Intelligence. Vol. 3: Generalized nets, Relational Data Bases and Expert Systems. "Prof. M. Drinov" Academic Publishing House, Sofia (2006)
12. Nan Liu, Bo Huang, Xiaohong Pan: Using the Ant Algorithm to Derive Pareto Fronts for Multiobjective Siting of Emergency Service Facilities. Transportation Research Record: Journal of the Transportation Research Board, No. 1935, Transportation Research Board of the National Academies, Washington, D.C., 2005, pp. 120129.