Generalized Nets as Tools for Modeling of the Ant Colony Optimization Algorithms

Stefka Fidanova[1] and Krassimir Atanassov[2]

[1] IPP – Bulgarian Academy of Sciences, Acad. G. Bonchev, bl.25A, 1113 Sofia, Bulgaria
stefka@parallel.bas.bg
[2] CLBME – Bulgarian Academy of Science, Acad. G. Bonchev, bl 105, 1113 Sofia, Bulgaria
krat@bas.bg

Abstract. Ant Colony Optimization (ACO) has been used successfully to solve hard combinatorial optimization problems. This metaheuristic method is inspired by the foraging behavior of ant colonies, which manage to establish the shortest routes to feeding sources and back. We discuss some possibilities for describing of the ACO algorithms by Generalized Nets (GNs), that help us deeply to understand the processes and to improve them. Very important is that the GNs are expandable. For example, some of the GN places can be replaced with a new GN. Thus we can include procedures to improve the search process and the achieved results or various kind of estimations of algorithm behavior.

1 Introduction

Generalized Nets (GNs) [1,2] are extensions of ordinary Petri Nets (PNs) and their other extensions, as Time PN, Color PNs, Stochastic PNs, Self-organizing PNs, etc. They are not only a tool for modelling of parallel processes, but the GN-approach for representing of real processes can play the role of a generator of new ideas for extending of the initial processes. In the present paper we will discuss some possibilities for application of the GNs as tool for modelling of Ant Colony Optimization (ACO) algorithms.

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.

The ACO is inspired by real ant behavior to solve hard combinatorial optimization problems. Examples of optimization problems are Traveling Salesman Problem, Vehicle Routing, Minimum Spanning Tree, Constrain Satisfaction, Knapsack Problem, etc. The ACO algorithm uses a colony of artificial ants that behave as cooperative agents in a mathematical space where they are allowed to

I. Lirkov, S. Margenov, and J. Waśniewski (Eds.): LSSC 2009, LNCS 5910, pp. 326–333, 2010.

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.

We discuss some possibilities for describing of the ACO algorithms by GNs, that help us deeply to understand the processes and to improve them. The ACO algorithm has parallel nature, thus the computational time can be decreased running the algorithm on supercomputers. By the GN description the parallel implementation of ACO can be improved.

The paper is organized as follows. In section 2 are done short remarks on GNs. In section 3 GN is represented as tool for modeling ACO algorithm. In section 4 an example of GN for ACO applied on knapsack problem is given. At the end some conclusions and directions for future work are done.

2 Short Remarks on Generalized Nets

In this section we shall introduce the concept of a Generalized Net. The GN consists of transitions, places, and tokens.

Every GN-transition is described by a seven-tuple (Fig. 1):

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

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. 1 these are $L' = \{l'_1, l'_2, \ldots, l'_m\}$ and $L'' = \{l''_1, l''_2, \ldots, l''_n\}$;

(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 [4]):

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

$r_{i,j}$ 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;

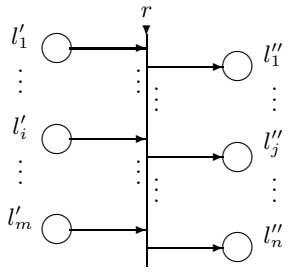

Fig. 1. GN-transition

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

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

(f) $\boxdot$ is an object having a form similar to a Boolean expression. It may contains 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:

$\wedge(l_{i_1}, l_{i_2}, \ldots, l_{i_u})$ - every place l_{i_1}, l_{i_2}, ..., l_{i_u} must contain at least one token,

$\vee(l_{i_1}, l_{i_2}, \ldots, l_{i_u})$ - there must be at least one token in all places l_{i_1}, l_{i_2},..., l_{i_u}, where $\{l_{i_1}, l_{i_2}, \ldots, l_{i_u}\} \subset L'$.

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

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

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

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 $N = \{0, 1, 2, \ldots\} \cup \{\infty\}$;

(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., $c : L \rightarrow N$;

(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) 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 $pr_3 Z = t, t' \in [T, T + t^*]$ 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 $pr_4 Z = t \in [T, T + t^*]$ 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

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

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., $\pi_K : K \rightarrow N$;

(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^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) 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., $b : K \rightarrow N$.

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 "*".

3 GNs as Tools for Modelling of ACO Algorithms

An interesting feature of GN is that they are expandable. A new transition can be included between every two transitions or at the beginning and the end of the net. Every one of the places of the net can be replaced with GN. Thus GN representation can show us the weakness of the method and possibilities for improvements. Hybridization of the method is represented by including new transitions or new GN between two transitions.

In [5] the functioning of standard ACO algorithm is described by GN. In [6] an extension of the first GN is constructed. It describes the functioning and the

results of the work of hybrid ACO algorithm including local search procedure. A next step of extension of the GN-model is done in [7], where ACO algorithm with fuzzy start of the ants on every iteration is described, including new transition at the beginning of the ACO-GN. Replacing a place from GN with new GN we can include new function of the ant.

Now, we shall represent some new modifications of ACO algorithms, ideas for which come from GN description of the method.

1. If over the path of some ant in the previous simulation there is a linear part, i.e., a part without branches, and if some ant treads on a node that is the beginning of such path region, in the present simulation all this path can be gone on one step.
2. We shall discuss introducing of elements of intuitionistic fuzzy estimations of the pheromone quantity. So we can obtain the degree of existing of the pheromone for a fixed node and the degree of the evaporated pheromone.
3. A GN-model of multiplicity ants can be constructed. In this case we can simulate the process of joint ant transfer to a node, where the paths of the different ants will be divided. The GN-model can use some restrictions for this situation, because if the ants multiply on each step, their number will increase exponentially. For example, we can allow multiplication only in the case when the ants are in a near neighborhood of a node, where there was a big value on the previous simulation. Another version of this idea is: only the ant that is the nearest to such node, can multiply, but for each time-moment this is possible only for one ant.
4. Including "mortal" ants in the algorithm. In this case some of the ants can die, e.g., when some of them tread on a node through which another ant already had went.
5. A modification of this model will be a model in which some (or all) ants have limited life-time. In this case we must find the maximal values for a limited time.

4 A GN-Model of ACO for Multiple Knapsack Problem

In this section we construct GN for Multiple Knapsack Problem (MKP). Description of Multiple Knapsack Problem you can see in [3]. The Generalized Net (GN, see [2,4]) have 4 transitions, 11 places, and three types (α, ε, and $\beta_1, \beta_2, \ldots$ — in advance determined number) of tokens (see Fig. 2). These tokens enter, respectively, places l_1 with the initial characteristic

"volumes of knapsacks"

and l_2 with the initial characteristics for the i-th β-token

"$\langle i$-th thing, volume, price$\rangle$".

$$Z_1 = < \{l_1\}, \{l_3\}, \begin{array}{c|c} & l_3 \\ \hline l_1 & W_{1,3} \end{array} >,$$

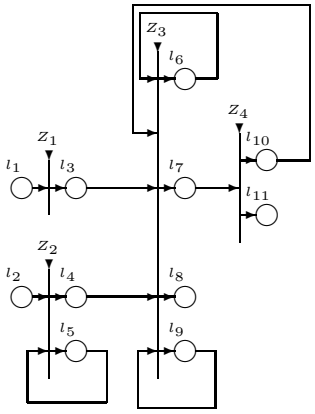

Fig. 2. GN net model for ACO

where
$W_{1,3}$ ="transition function".

The aim of this construction is: token α to go to place l_3 when all β-tokens are already in place l_5 and therefore we can calculate the common price of the things (let they are N in number)

$$CP = \sum_{i=1}^{N} pr_3 x_0^{\beta_i},$$

where x_s^{ω} is s-th characteristic ($s \geq 0$; 0-th characteristic is the initial one) of token ω.

Now, token α can obtain as a new characteristic

$$\text{"middle price for unit volume, i.e.} \frac{CP}{x_0^{\alpha}}\text{"}.$$

Token α from places l_1 enters place l_3 with a characteristic

"vector of current transition function results $\langle \varphi_{1,cu}, \varphi_{2,cu}, ..., \varphi_{n,cu} \rangle$",

while token ε stays only in place l_4 obtaining the characteristic

"new m-dimensional vector of heuristics with elements – the graph vertices or,

new l-dimensional vector of heuristics with elements – the graph arcs".

$$Z_2 = < \{l_2\}, \{l_4, l_5\},$$

$$\begin{array}{c|cc} & l_4 & l_5 \\ \hline l_2 & W_{2,4} & W_{2,5} \end{array} >$$

where:
$W_{2,4}$ = new m-dimensional vector of heuristics,
$W_{2,5}$ = all β-tokens are collected already in place l_5.

$$Z_3 =< \{l_3, l_4, l_6, l_9, l_{10}\}, \{l_6, l_7, l_8, l_9\},$$

	l_6	l_7	l_8	l_9
l_3	$W_{3,6}$	$W_{3,7}$	$false$	$W_{3,9}$
l_4	$W_{4,6}$	$W_{4,7}$	$false$	$W_{4,9}$
l_6	$W_{6,6}$	$false$	$false$	$false$
l_9	$false$	$W_{9,7}$	$W_{9,8}$	$W_{9,9}$
l_{10}	$true$	$true$	$true$	$false$

$>$

where:
$W_{3,7} = W_{3,6} = W_{3,9} = \neg W_{6,6} \vee \neg W_{9,8}$,
$W_{4,6} = W_{4,7} = W_{4,9} = \neg W_{6,6} \vee \neg W_{9,8}$,
$W_{6,6}$ = the current iteration is not finished, $W_{9,7}$ = the current best solution is better than global best solution,
$W_{9,8}$ = "truth-value of expression $C_1 \vee C_2 \vee C_3$ is $true$",
$W_{9,9} = \neg W_{9,8}$.
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 α from place l_3 enters place l_6 with a characteristic

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

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

If $W_{6,6} = true$ 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_7 – token α' — with the last α-characteristic, and place l_9 — token α'' — with a characteristic

"$\langle$the best solution for the current iteration; its number$\rangle$".

Token α'' can enter place l_8 only when $W_{9,8} = true$ and there it obtains the characteristic

"the best achieved result".

$$Z_4 =< \{l_7\}, \{l_{10}, l_{11}\},$$

	l_{10}	l_{11}
l_7	$W_{7,10}$	$W_{7,11}$

$>$

where:
$W_{7,10}$ = "truth-value of expression $C_1 \vee C_2 \vee C_3$ is $true$",
$W_{7,11} = \neg W_{7,10}$.

5 Conclusion

This paper shows description of ACO algorithm by GN. This description presents us the weakness of the algorithm and points us possible improvements. Important feature of the GNs is that they are expandable. So we can add new transitions and replace places with other GNs. This gives us ideas how to construct new modified ACO algorithms and to improve achieved results. Like a future work we shell develop new ACO algorithms and we shell apply them on various difficult optimization problems. The aim is to construct the most appropriate algorithm for given class of problems. The nature of ant algorithms are parallel with big transfer of data after every iteration. So they are very suitable for parallel applications on massively parallel architectures and supercomputers.

Acknowledgments

This work has been partially supported by the Bulgarian NSF under contract DO 02–115.

References

1. Alexieva, J., Choy, E., Koychev, E.: Review and bibloigraphy on generalized nets theory and applications. In: Choy, E., Krawczak, M., Shannon, A., Szmidt, E. (eds.) A Survey of Generalized Nets. Raffles KvB Monograph, vol. 10, pp. 207–301 (2007)
2. Atanassov, K.: Generalized Nets. World Scientific, Singapore (1991)
3. Fidanova, S.: Ant colony optimization and multiple knapsack problem. In: Renard, J.P. (ed.) Handbook of Research on Nature Inspired Computing for Economics and Management, pp. 498–509. Idea Grup Inc. (2006) ISBN 1-59140-984-5
4. Atanassov, K.: On Generalized Nets Theory. Prof. M. Drinov Publishing House, Sofia (2007)
5. Fidanova, S., Atanasov, K.: Generalized Net Models of the Process of Ant Colony Optimization. Issues on Intitionistic Fuzzy Sets and Generalized Nets 7, 108–114 (2008)
6. Fidanova, S., Atanassov, K.: Generalized Net Model for the Process of Hybrid Ant Colony Optimization. Computer Rendus de l'Academie Bulgare des Sciences 62(3), 315–322 (2009)
7. Fidanova, S., Atanasov, K.: Generalized Net Models of the Process of the Ant Colony Optimization with Intuitionistic Fuzzy Estimations. In: Proc. of the Ninth Int. Conf. of Generalized nets, Sofia, pp. 41–48 (2008)