МАТЕМАТИКА И МАТЕМАТИЧЕСКО ОБРАЗОВАНИЕ, 2001
MATHEMATICS AND EDUCATION IN MATHEMATICS, 2001
Proceedings of Thirtieth Spring Conference of
the Union of Bulgarian Mathematicians
Borovets, April 8–11, 2001

A GENERALIZED NET FOR GENETIC ALGORITHMS LEARNING

Hristo Tsvetanov Aladjov, Krassimir Todorov Atanassov

A generalized net describing the process of learning of genetic algorithm is constructed. Some extensions of the concept of genetic algorithm are discussed.

Introduction. The means of the Generalized Net (GN, see, e.g. [1]) theory are especially suitable for describing parallel algorithms that investigate several solutions simultaneously. Being such algorithms, the Genetic Algorithms (GAs, see, e.g. [2]) do not make an exception. Here we construct a GN describing the process of GA learning and the results of the functioning of this process.

Short remarks on Generalized Nets (GNs). The concept of a *Generalized Net* (GN) is described in details in [1], see also

`www.daimi.aau.dk/PetriNets/bibl/aboutpnbibl.html`

These nets are the essential extensions of the ordinary Petri nets. GNs are defined in a way that is principly different from the ways of defining the other types of Petri nets. When some of the GN-conponents of a given model are not necessary, they can be omitted and the new nets are called reduced GNs. Here a reduced GN without temporal components is used.

Formally, every transition (in the used form of a reduced GN) is described by a five-tuple:

$$Z = \langle L', L'', r, \Box \rangle,$$

where:

(a) L' and L'' are finite, non-empty sets of places (the transition's input and output places, respectively); for the above transition these are

$$L' = \{l'_1, l'_2, \dots, l'_m\}$$

and

$$L'' = \{l''_1, l''_2, \dots, l''_n\};$$

(b) 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]):

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

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

(c) $\square$ is an object having a form similar to a Boolean expression. It may contain as variables the symbols that serve as labels for transition's input places, and is an expression built up from variables and the Boolean connectives $\wedge$ and $\vee$, with semantics 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 object

$$E = \langle A, \pi_L, K, X, \Phi \rangle$$

is called a (reduced) GN, if

(a) A is a set of transitions;

(b) 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 $n \in \mathcal{N}, n \geq 1$ and $1 \leq k \leq n$ (obviously, L is the set of all GN-places), where $\mathcal{N}$ is the set of the natural numbers;

(c) K is the set of the GN's tokens;

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

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

A GN-model for GA-learning. The GN-model of a genetic algorithm is shown on Fig.1. It consists of three contours. Tokens from type α describing individuals (solutions) move in the first of them (the $l-$contour). A single $\beta-$token, for every genetic algorithm, describing the algorithm itself, moves in the second one (the $m-$contour). A $\gamma-$token for each $\beta-$tokens loops in the third one (the $n-$contour). The $\gamma-$token controls parameters and estimates the performance of the genetic algorithm represented by the $\beta-$token.

Initially, $\alpha-$tokens representing individuals in terms of genetic algorithms theory enter l_1. They have the following initial characteristic of:

$$x_0^\alpha = \text{“}\langle I, C, f \rangle\text{”},$$

where

$I \in \mathcal{N}$ is the identifier of the individual;
C is the chromosome of the individual;
$f \in \mathcal{R}$ is the individual fitness, where $\mathcal{R}$ is the set of the real numbers.

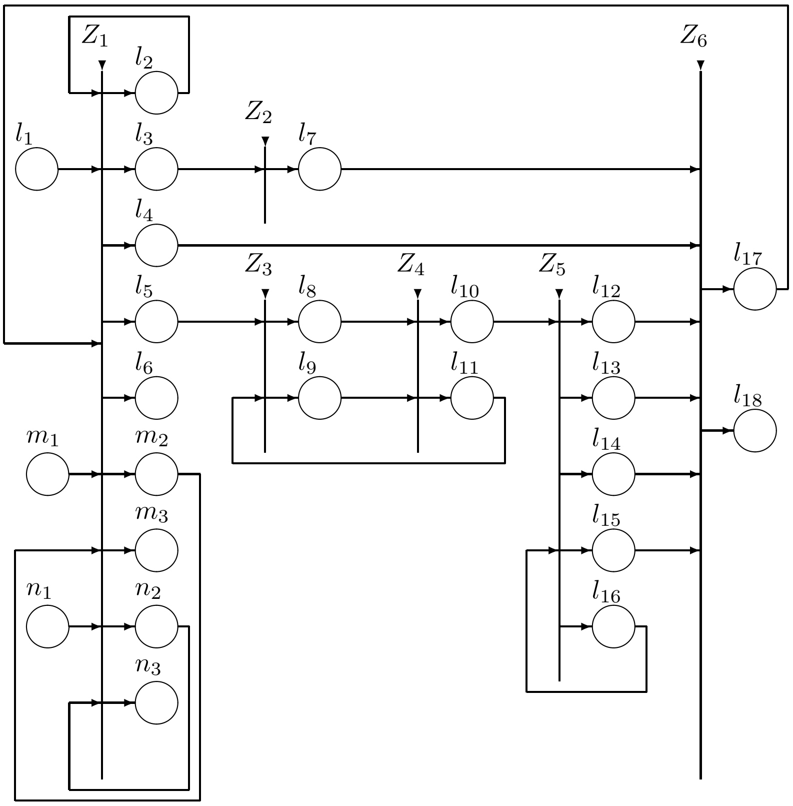
Together with the $\alpha-$tokens, a single $\beta-$ token enters in m_1. It has the parameters and functions of the genetic algorithm as a characteristic.

$$x_0^{\beta} = \text{“}\langle Task, N, O, F, S, R\rangle\text{”},$$

where
$Task \in \{$“*estimate*”, “*select*”, “*process*”, “*reinsert*”$\}$;
$N \in \mathcal{N}$ is the number of the individuals being controlled by the algorithm;

$$O = \{\langle Op, p\rangle | Op \in U_{Op}, p \in [0, 1]\}$$

is the set of operations over the individuals and their probabilities and U_{Op} is the set of all operations. Such operations are: one, two, ..., $n-$point crossover; and one, two,... $n-$ point mutation;
$FF : U_C \rightarrow \mathcal{R}$ is the fitness function. Where U_C is the set of all chromosomes;
$S : U_C \rightarrow \{$“*survive*”, “*die*”, “*mutate*”, “*crossover*”$\}$ is the selection function that determines what will happen with the individual;
$R : U_C \rightarrow \{$“*false*”, “*true*”$\}$ is the reinsertion function that determines how the offspring insertes in the current population.



For each $\beta-$token one γ token enters the net. This token describes the process that controls the execution of the genetic algorithm and adjusts its parameters and functions. In the real genetic algorithms if this process is present, the algorithm itself usually implements it. Therefore, the $\gamma-$ token characteristic has the following form:

$$x_i^\gamma = \text{“}\langle T, M, E\rangle\text{”},$$

where
T is an estimation of the total efficiency of the algorithm;
M is the modification function that changes functions F, S, R, operations and their rates;
E is the end-condition function that determines whether the algorithm terminates its execution or not.

All $\alpha-$tokens gather in l_2 where the value of the fitness function for the individual is updated, i.e. field f from their characteristic:

$$pr_3(x_i^\alpha) = pr_4(x_{cu}^\beta)(pr_2(x_i^\alpha))$$

is updated.

If the selection function requires total fitness (fitness of all members of the population) then all $\alpha-$tokens must receive their new value of f. Otherwise, no awaiting is needed and individuals can be selected for mutation, survival, crossover or death. These four states are represented by places l_5, l_6, l_7 and l_8, respectively. The probability for a given action is determined by the probability of the operation and the fitness value of the individual.

For each step of the algorithm $\beta-$ and $\gamma-$tokens loop in m_2 and n_2, respectively.

Transition Z_1 has the following formal definition:

$$Z_1 = \langle \{l_1, l_2, l_{17}, m_1, m_2, n_1, n_2\}, \{l_2, l_3, l_4, l_5, l_6, m_2, m_3, n_2, n_3\}, r_1,$$

$$\wedge(\vee(l_1, l_2, l_{17}), \vee(m_1, m_2), \vee(n_1, n_2))\rangle,$$

where

$$r_1 = \begin{array}{c|ccccccccc} & l_2 & l_3 & l_4 & l_5 & l_6 & m_2 & m_3 & n_2 & n_3 \\ \hline l_1 & W_1 & W_2 & W_3 & W_4 & W_5 & false & false & false & false \\ l_2 & W_1 & W_2 & W_3 & W_4 & W_5 & false & false & false & false \\ l_{17} & W_1 & W_2 & W_3 & W_4 & W_5 & false & false & false & false \\ m_1 & false & false & false & false & false & W_6 & \neg W_6 & false & false \\ m_2 & false & false & false & false & false & W_6 & \neg W_6 & false & false \\ n_1 & false & false & false & false & false & false & false & W_7 & \neg W_7 \\ n_2 & false & false & false & false & false & false & false & W_7 & \neg W_7 \end{array},$$

and
$W_1 = \text{“}pr_4(x_{cu}^\beta) = \text{'}estimate\text{' ”}$
$W_2 = \text{“}(pr_4(x_{cu}^\beta) = \text{'}select\text{'})\text{” \& “}(pr_4(x_{cu}^\beta)(pr_2(x_i^\alpha)) = \text{'}mutate\text{'})\text{”}$
$W_3 = \text{“}(pr_4(x_{cu}^\beta) = \text{'}select\text{'})\text{” \& “}(pr_4(x_{cu}^\beta)(pr_2(x_i^\alpha)) = \text{'}survive\text{'})\text{”}$
$W_4 = \text{“}(pr_4(x_{cu}^\beta) = \text{'}select\text{'})\text{” \& “}(pr_4(x_{cu}^\beta)(pr_2(x_i^\alpha)) = \text{'}crossover\text{'})\text{”}$

$W_5 =$ “$(pr_4(x^{\beta}_{cu}) = {'select'})$” & “$(pr_4(x^{\beta}_{cu})(pr_2(x^{\alpha}_i)) = {'die'})$”
$W_6 =$ “$pr_4(x^{\beta}_{cu}) \neq \emptyset$”
$W_7 =$ “$pr_4(x^{\beta}_{cu}) \neq \emptyset$”.

The $\alpha-$tokens chosen for mutation transfer from l_3 to l_7 and obtain as a characteristic the new chromosome description

$$x^{\alpha}_i = \text{“}\langle I, C', f\rangle\text{”},$$

where C' can be the result of one, two or multi-point mutation or, in general, any operation with argument beeng a single individual chromosome. This transformation is represented by transition

$$Z_2 = \langle \{l_3\}, \{l_7\}, r_2, \wedge(l_3)\rangle,$$

where

$$r_2 = \begin{array}{c|c} & l_7 \\ \hline l_3 & W_8 \end{array},$$

where
$W_8 =$ “$pr_4(x^{\beta}_{cu}) = {'process'}$ ”.

The individuals that can “survive” without no changes of their chromosomes are represented by $\alpha-$tokens in place l_4. According to the value of the reinsertion function, they will either “survive” and transfer to l_{17} or will “die” and leave the net via l_{18}.

The individuals chosen for crossover operation are represented by a set of $\alpha-$tokens in place l_5. All of them transfer into l_8 and obtain no new characteristic. After that the corresponding tokens of chosen couples of individuals unite in l_{10} and the rest of the tokens transfer in l_{11}. Tokens from l_{11} return back to l_9 in order to participate in the choice of the next couples.

In the case of a one point crossover operation united tokens from place l_{10} will split into two “parent” individuals, represented by $\alpha-$ tokens in l_{12} and l_{13}, and two “child” individuals represented by tokens in l_{14} and l_{15}, respectively. If the crossover operation has more than one crossover point then a token from l_{10} will transfer in l_{16} and loop there until the hole the offspring is generated. After each step of that loop new “child” individual are reproduced in l_{14} and l_{15}. In the final step the token from l_{16} splits into two tokens in places l_{12} and l_{13}, respectively.

$$Z_3 = \langle \{l_5, l_{11}\}, \{l_8, l_9\}, r_3, \vee(l_5, l_{11})\rangle,$$

where

$$r_3 = \begin{array}{c|cc} & l_8 & l_9 \\ \hline l_5 & W_8 & false \\ l_{11} & false & true \end{array},$$

where W_8 is defined above.

$$Z_4 = \langle \{l_8, l_9\}, \{l_{10}, l_{11}\}, r_4, \vee(l_8, l_9)\rangle,$$

where

$$r_4 = \begin{array}{c|cc} & l_{10} & l_{11} \\ \hline l_8 & W_8 \& W_9 & W_8 \& \neg W_9 \\ l_9 & W_8 \& W_9 & W_8 \& \neg W_9 \end{array},$$

where
$W_9 =$ "the individuals are in suitable pairs".

$$Z_5 = \langle \{l_{10}, l_{16}\}, \{l_{12}, l_{13}, l_{14}, l_{15}, l_{16}\}, r_5, \vee(l_{10}, l_{16})\rangle,$$

where

$$r_5 = \begin{array}{c|ccccc} & l_{12} & l_{13} & l_{14} & l_{15} & l_{16} \\ \hline l_{10} & W_{10} & W_{10} & W_{10} & W_{10} & \neg W_{10} \\ l_{16} & W_{11} & W_{11} & true & true & \neg W_{11} \end{array}$$

where
$W_{10} =$ "the operation is one point crossover",
$W_{11} =$ "$2n$ children are generated",
where n is the number of crossover points.

Now all $\alpha-$tokens that represent individuals including the new offsprings are the input places of transition Z_6. Reinsertion function will determine which of them will remain in the population and which of them not. Tokens that represent survived individuals enter l_{17}; the rest of the tokens leave the net via place l_{18}.

$$Z_6 = \langle \{l_4, l_7, l_{12}, l_{13}, l_{14}, l_{15}\}, \{l_{17}, l_{18}\}, r_6, \vee(l_4, l_7, l_{12}, l_{13}, l_{14}, l_{15})\rangle,$$

where

$$r_6 = \begin{array}{c|cc} & l_{17} & l_{18} \\ \hline l_i & W_{12} & \neg W_{12} \end{array}$$

where $(i = 4, 7, 12, 13, 14, 15)$
$W_{12} =$ "$(pr_4(x^{\beta}_{cu}) = \textit{'reinsert'})$" & "$(pr_6(x^{\beta}_{cu})(pr_2(x^{\alpha}_{i})))$".

Conclusion. In the model described above no restrictions of intrinsic parallelism of genetic algorithms are imposed. So the algorithms can be studied and compared without the limitations forced by their implementations on sequential architectures. Many extensions of the classical concept of genetic algorithms, such as the Global GA [3], the Migration GA, the Diffusion GA, the Multi objective GA and Hierarchical GA [2] can be easily represented in the model.

The definition of the model provokes some additional possibilities, such as operations with more then two arguments, modification of selection, reinsertion or fitness functions during the execution of the algorithm, and other. The constructed model is a new illustration of the idea discussed in [4] that all areas of the artificial intelligence can be described by the means of the GNs. This will help the future research of complex processes, separate parts of which are objects of studying of different areas of the artificial intelligence. The apparatus of the GNs give a possibility to unite the models of the separate processes to a global one.

REFERENCES

[1] K. Atanassov. Generalized Nets. Singapore, New Jersey, London, World Scientific, 1991.
[2] K. Man, K. Tang, S. Kwong. Genetic Algorithms. Concepts and Designs. London, Springer-Verlag, 1999.
[3] D. E. Goldberg. Genetic algorithms in search, optimization and machine learning. New York, Addison-Wesley, 1989.

[4] K. Atanassov. Generalized Nets in Artificial Intelligence. Vol. 1 Generalized nets and Expert Systems. Sofia, "Prof. M. Drinov" Academic Publishing House, 1998.

CLBME - Bulgarian Academy of Sciences
P.O. Box 12, Sofia-1113, Bulgaria
e-mails: {`aladjov, krat`}`@bgcict.acad.bg`

ОБОБЩЕНА МРЕЖА ЗА ОБУЧЕНИЕ НА ГЕНЕТИЧНИ АЛГОРИТМИ

Христо Цветанов Аладжов, Красимир Тодоров Атанасов

Построена е обобщена мрежа, описваща процеса на обучение на генетичен алгоритъм. Обсъждат се някои възможни разширения на понятието генетичен алгоритъм.