

Generalized Net for Evaluation of the Genetic Algorithm Fitness Function

Olympia Roeva¹, Krassimir Atanassov^{1,2} and Anthony Shannon^{2,3}

¹ Centre of Biomedical Engineering “Prof. Ivan Daskalov”,
Bulgarian Academy of Sciences
105 Acad. G. Bonchev Str., 1113 Sofia, Bulgaria,
e-mails: *olympia@clbme.bas.bg* and *krat@bas.bg*

² KvB Institute of Technology, North Sydney, 2060, AUSTRALIA

³ Warrane College, University of New South Wales, Kensington, 1465, AUSTRALIA,
emails: *tonySHANNON@raffles.kvb.edu.au* and *tony@warrane.unsw.edu.au*

Abstract Using the apparatus of Generalized nets (GN) a GN model of a genetic algorithm is developed. The presented GN model describes the genetic algorithm search procedure based on the mechanism of natural selection. The GN model simultaneously evaluates several fitness functions, ranks the individuals according to their fitness and has the opportunity to choice the best fitness function regarding to specific problem domain.

1 Introduction

Genetic algorithms (GAs), proposed by John Holland in 1975, are global, parallel, stochastic search methods, based on Darwinian evolutionary principles [7]. Even nowadays there exists big interest in both theoretical [14] and application [11] aspects of GA.

GAs may be configured in many ways, and these different setups may have strong effects on the solutions found. Fitness functions, crossover operators, mutation operators, selection operators, and population size are just a few of the many parameters that are due to optimization. There are a lot of papers on various techniques researchers have found to set up these parameters for various problem domains [12].

GAs methods consist of a population of individuals competing on a “survival of the fittest” basis. Each individual, or chromosome, represents a trivial solution of the problem. The algorithm, transforming the chromosome population into a new population with more adapted individuals, proceeds in successive steps called generations. During each generation, the population evolves by means of a “fitness” function that selects individuals by standard crossover and mutation operators.

The key point in the performance of a genetic algorithm is the definition of the Fitness Function (FF). Selecting the best chromosomes requires defining a fitness function able to evaluate the “quality” of the solutions proposed, i.e. their ability to discriminate data pertaining to different classes. Many authors have proposed different FF according the considered problem [4, 5, 6, 8, 9, 10].

The main idea of this paper is to develop a genetic algorithm that proposed the possibility to calculate the individual fitness based on a set of different FF and to choose the

“best” function for a given problem. As an appropriate tool for developing of such genetic algorithm the apparatus of Generalized Nets (GNs, see [2]) is considered. The means of the GN theory are especially suitable for describing parallel algorithms that investigate several solutions simultaneously.

The research in [1], extended to [3] is the first implementation of GNs for description of genetic algorithms. Based on this research a generalized net model of a genetic algorithm is proposed here.

2 Genetic algorithm performance

The outline of an classical genetic algorithm could be presented as:

1. **[Start]** Generate random population of n individuals (suitable solution of the problem).
2. **[Fitness]** Evaluate the fitness function of each individuals in the population.
3. **[New population]** Create a new population by repeating the following steps:
 - 3.1. **[Selection]** Select parent chromosomes from a population according to their fitness.
 - 3.2. **[Crossover]** With a crossover probability cross over the parents to form new offspring (children).
 - 3.3. **[Mutation]** With a mutation probability mutate new offspring.
4. **[Replace]** Use new generates population for a further run of the algorithm.
5. **[Test]** If the condition is satisfied, stop, and return the best solution in current population.
6. **[Loop]** Go to step 2.

The genetic algorithm works on a coding of the parameter space. Having decided on the coding to be used initial population is created at random. The population contains a certain number of individuals. An objective function (usually referred to as fitness function in GAs terminology) serves as a measure of goodness of a string and is a functional of the function that will be optimized. In the next step individuals represented by their fitness function are ranked. Individuals from a population are selected according to their fitness. Thus solutions from one population are taken and used to form a new population. This is motivated by the hope, that the new population will be better than the old one. A certain function performing selection is applied. In the next step a set of operators is applied to the initial populations to generate successive generations. The selected individuals are then recombined in order to form new offspring (children) crossing over the parents with a crossover probability. Then, mutation takes place with mutation rate. For the new individuals of the population the objective function values are calculated. Reinsertion function is used. The new offspring is inserted in the population. The new generated population is used for a further run of the algorithm. Natural evolution of the population continues until the termination criterion is satisfied.

Based on above mentioned description a GN model of a GAs is proposed.

3 Generalized net model

The GN model of the regarded genetic algorithm consists of three contours. Tokens from type α describing individuals (solutions) move in the first of them (the l -contour). A single β -token, describing the algorithm itself, moves in the second (the m -contour). A γ -token for each β -token loops in the third (the n -contour). The γ -token controls the set of fitness functions and estimates the performance of the genetic algorithm represented by the β -token. The GN model of a genetic algorithm is shown on Fig. 1.

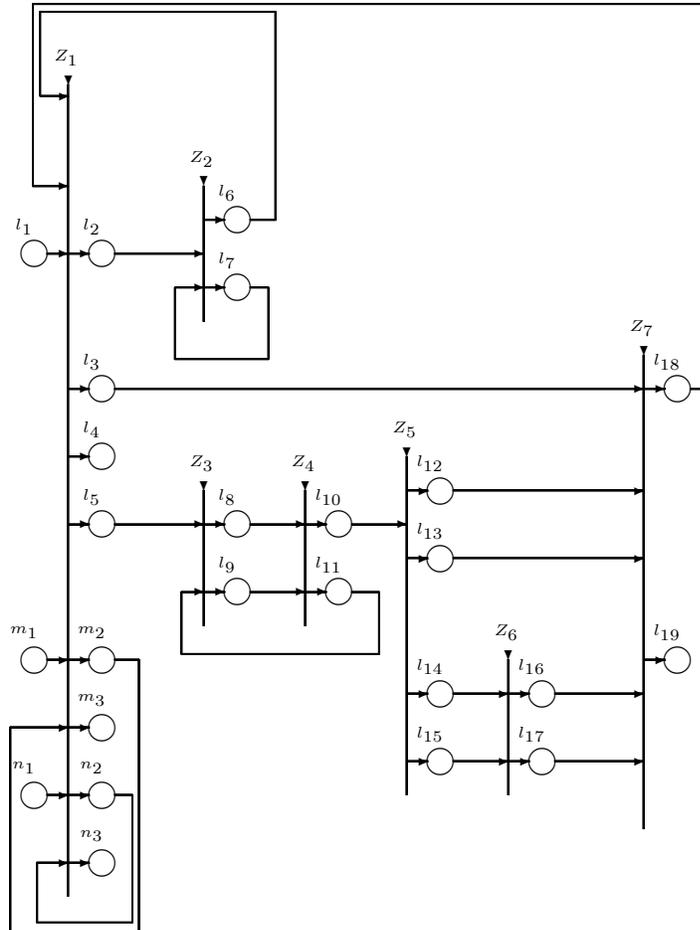


Fig. 1

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

$$x_0^\alpha = \langle I, C, f \rangle,$$

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.

Together with the α -tokens, a single β -token enters m_1 . It has as a characteristic the parameters and functions of the genetic algorithm.

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

where

$\textit{Task} \in \{ \textit{“estimate”}, \textit{“select”}, \textit{“process”}, \textit{“reinsert”} \};$

$N \in \mathcal{N}$ is the number of the individuals controlled by the algorithm;

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

is the set of operations over the individuals and their probabilities, where $U_{\textit{Op}}$ is the set of crossover and mutation functions and p is the probability of crossover and mutation.

$F : U_C \rightarrow \mathcal{R}$ is a set of fitness functions. Where U_C is the set of all chromosomes;

$S : U_C \rightarrow \{ \textit{“survive”}, \textit{“die”}, \textit{“mutate”}, \textit{“crossover”} \}$ is the selection function that determines what will happen with the individual;

$R : U_C \rightarrow \{ \textit{“false”}, \textit{“true”} \}$ is the reinsertion function that determines how the offspring inserts in the current population.

For each β -token one γ -token enters the net. This token describes the process that controls the execution of the genetic algorithm. The γ -token characteristic has the following form:

$$x_i^\gamma = \langle \textit{T}, \textit{M}, \textit{E} \rangle,$$

where

\textit{T} is an estimation of the total efficiency of the algorithm;

\textit{M} is the modification function that changes the available set of fitness functions;

\textit{E} is the end-condition function that determines whether the algorithm will terminate its execution.

For each step of the algorithm the β -and γ -tokens loop in m_2 and n_2 , respectively.

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

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

After evaluation of the definite set of fitness functions the individuals are selected for survival or death, crossover and mutation. The probability for a given action is determined by the fitness value of the individual and the probability of the operations.

Transition Z_1 has the following formal definition:

$$Z_1 = \langle \{l_1, l_6, l_{18}, m_1, m_2, n_1, n_2\}, \{l_2, l_3, l_4, l_5, l_6, m_2, m_3, n_2, n_3\}, r_1, \wedge(\vee(l_1, l_6, l_{18}), \vee(m_1, m_2), \vee(n_1, n_2)) \rangle,$$

where

	l_2	l_3	l_4	l_5	m_2	m_3	n_2	n_3
l_1	W_1	W_2	W_3	W_4	$false$	$false$	$false$	$false$
l_6	W_1	W_2	W_3	W_4	$false$	$false$	$false$	$false$
l_{18}	W_1	W_2	W_3	W_4	$false$	$false$	$false$	$false$
m_1	$false$	$false$	$false$	$false$	W_5	$\neg W_5$	$false$	$false$
m_2	$false$	$false$	$false$	$false$	W_5	$\neg W_5$	$false$	$false$
n_1	$false$	$false$	$false$	$false$	$false$	$false$	W_6	$\neg W_6$
n_2	$false$	$false$	$false$	$false$	$false$	$false$	W_6	$\neg W_6$

and

$$W_1 = \text{“}pr_4(x_{cu}^\beta) = 'estimate'\text{”},$$

$$W_2 = \text{“}(pr_4(x_{cu}^\beta) = 'select') \& \text{“}(pr_4(x_{cu}^\beta)(pr_2(x_i^\alpha)) = 'survive'\text{”},$$

$$W_3 = \text{“}(pr_4(x_{cu}^\beta) = 'select') \& \text{“}(pr_4(x_{cu}^\beta)(pr_2(x_i^\alpha)) = 'die'\text{”},$$

$$W_4 = \text{“}(pr_4(x_{cu}^\beta) = 'select') \& \text{“}(pr_4(x_{cu}^\beta)(pr_2(x_i^\alpha)) = 'crossover'\text{”},$$

$$W_5 = \text{“}pr_4(x_{cu}^\beta) \neq \emptyset\text{”},$$

$$W_6 = \text{“}pr_4(x_{cu}^\beta) \neq \emptyset\text{”}.$$

The form of the transition Z_2 is:

$$Z_2 = \langle \{l_2, l_7\}, \{l_6, l_7\}, r_2, \wedge(l_2, l_7) \rangle,$$

where

$$r_2 = \begin{array}{c|cc} & l_6 & l_7 \\ \hline l_2 & false & W_7 \\ l_7 & \neg W_7 & W_7 \end{array},$$

where

$$W_7 = \text{“}pr_4(x_{cu}^\beta) = 'process'\text{”}.$$

In position l_6 the individuals are ranked in groups according to their fitness. Depending on the accepted strategy the generalized net will be operate over different groups, for instance: most of the last groups will be die; some of the first groups will be survive for further generation and all groups, except these individuals that leave the generalized net, will be cross over and mutate.

The individuals that can “survive” with no changes in their chromosomes are represented by α -tokens in place l_3 . According to the value of the reinsertion function they will either “survive” and transfer to l_{18} or will “die” and leave the net via l_{19} .

The individuals that can “died” are represented by α -tokens in place l_4 .

The individuals chosen for the crossover operation are represented by a set of α -tokens in place l_5 . All of them transfer into l_8 and obtain no new characteristics. 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 α -tokens in l_{12} and l_{13} , and two “child” individuals represented by tokens in l_{14} and l_{15} , respectively. After each step of that loop new “child” individuals are reproduced in l_{14} and l_{15} . The “child” individuals are mutated and the results is represented by tokens in l_{16} and l_{17} .

The forms of the transitions Z_3 , Z_4 and Z_5 are:

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

where

$$r_3 = \frac{\quad \mid \quad l_8 \quad l_9}{\begin{array}{l} l_5 \mid W_7 \quad false \\ l_{11} \mid false \quad true \end{array}},$$

where W_7 is defined above.

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

where

$$r_4 = \frac{\quad \mid \quad l_{10} \quad l_{11}}{\begin{array}{l} l_8 \mid W_7 \& W_8 \quad W_8 \& \neg W_8 \\ l_9 \mid W_7 \& W_8 \quad W_8 \& \neg W_8 \end{array}},$$

where

W_8 = “the individuals are in suitable pairs”.

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

where

$$r_5 = \frac{\quad \mid \quad l_{12} \quad l_{13} \quad l_{14} \quad l_{15}}{l_{10} \mid W_9 \quad W_9 \quad W_9 \quad W_9},$$

where

W_9 = “the operation is one point crossover”.

The α -tokens (new “child” individuals) chosen for mutation transfer from positions l_{14} and l_{15} to positions l_{16} and l_{17} and obtain as a characteristic the new chromosome description:

$$x_i^\alpha = \langle I, C', f \rangle,$$

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

$$Z_6 = \langle \{l_{14}, l_{15}\}, \{l_{16}, l_{17}\}, r_6, \vee(l_{14}, l_{15}) \rangle,$$

where

$$r_6 = \frac{\quad \mid \quad l_{16} \quad l_{17}}{\begin{array}{l} l_{14} \mid W_{10} \quad false \\ l_{15} \mid false \quad W_{10} \end{array}},$$

where W_{10} = “the operation is mutation”.

Now all α -tokens that represent individuals including the new offspring is the input places of transition Z_7 . Reinsertion function will determine which of them will remain in the population and which of them will not. Tokens that represent survived individuals enter place l_{18} ; the rest of the tokens leave the net via place l_{19} .

$$Z_7 = \langle \{l_3, l_{12}, l_{13}, l_{16}, l_{17}\}, \{l_{18}, l_{19}\}, r_7, \vee(l_3, l_{12}, l_{13}, l_{16}, l_{17}) \rangle,$$

where

$$r_7 = \frac{\quad \mid \quad l_{18} \quad l_{19}}{\begin{array}{l} l_3 \mid W_{11} \quad \neg W_{11} \\ l_{12} \mid W_{11} \quad \neg W_{11} \\ l_{13} \mid W_{11} \quad \neg W_{11} \\ l_{16} \mid W_{11} \quad \neg W_{11} \\ l_{17} \mid W_{11} \quad \neg W_{11} \end{array}},$$

where

$$W_{11} = "(pr_4(x_{cu}^\beta) = 'reinsert')" \& "(pr_6(x_{cu}^\beta)(pr_2(x_i^\alpha)))".$$

At every step utilizing definite set of fitness functions the GN model evaluates fitness value of the individuals. Based on the received fitness values individuals are ranking in the groups. The “best” individuals for every FF are ranked in the first group. The “worst” individuals are ranked in the last group. Individuals from the last groups (last one, last two, etc.) died. On the one hand the individuals from the first group or from the first two groups survived for the next generation of the algorithm without changes. On the other hand these individuals together with the rest survival individuals are process applying crossover and mutation functions.

4 Conclusion

A generalized net model of a genetic algorithm is developed. Performing series of experiments the presented GN model could be defined the “best” fitness function for each considered problem domain. The proposed net tests the effect that the rate of change in the fitness function has on GAs performance and proposed such fitness function that enhances the efficiency of GAs.

References

- [1] Aladjov, H., K. Atanassov. A generalized net for genetic algorithms learning. – In: Proc. of the XXX Spring Conf. of the Union of Bulgarian Mathematicians, Borovets, 2001, 242–249.
- [2] Atanassov, K. Generalized Nets. World Scientific, Singapore, New Jersey, London, 1991.
- [3] Atanassov, K., H. Aladjov. Generalized Nets in Artificial Intelligence. Vol. 2: Generalized nets and Machine Learning. “Prof. M. Drinov” Academic Publishing House, Sofia, 2000.
- [4] Chen Q., K. Worden, P. Peng, A.Y.T. Leung, Genetic algorithm with an improved fitness function for (N)ARX modelling , Mechanical Systems and Signal Processing, 21(2), 2007, 994–1007.
- [5] Clerc F., R. Rakotomalala, D. Farrusseng, Learning fitness function in a combinatorial optimization process, – In: Proc. of International Symposium on Applied Stochastic Models and Data Analysis (ASMDA), May, 17-20, 2005, 535–542.
- [6] Diaz-Gomez P. A., D. F. Hougen, Genetic Algorithms for Hunting Snakes in Hypercubes: Fitness Function Analysis and Open Questions, – In: Proc. of Seventh ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD’06), 2006, 389–394.
- [7] Goldberg, D.: Genetic algorithms in search, optimization and machine learning. Addison-Wesley Publishing Company, Massachusetts, 1989.

- [8] Inagaki J., M. Haseyama, H. Kitajima, A New Fitness Function of a Genetic Algorithm for Routing Applications, IEICE transactions on information and systems, E84-D(2), 2001, 277–280.
- [9] Kazarlis S., V. Petridis, Varying fitness functions in genetic algorithms: Studying the rate of increase of the dynamic penalty terms, Lecture Notes in Computer Science, 1498/1998, 2006, 211–220.
- [10] Kita H., Y. Sano, Genetic Algorithms for Optimization of Noisy Fitness Functions and Adaptation to Changing Environments, 2003 Joint Workshop of Hayashibara Foundation and SMAPIP, July 11-15, 2003, Okayama, Japan.
- [11] Man, K., K. Tang, S. Kwong. Genetic Algorithms. Concepts and Designs. London, Springer-Verlag, 1999.
- [12] Mitchell M., An Introduction to Genetic Algorithms (Complex Adaptive Systems Series). Cambridge, MA: MIT Press, 1996.
- [13] Pohlheim, H.: Genetic and Evolutionary Algorithms: Principles, Methods and Algorithms, available at http://www.systemtechnik.tu-lmenau.de/~pohlheim/GA_Toolbox/algindex.html
- [14] Vise, M. The Simple Genetic Algorithm. Cambridge, MIT Press, 1999.