

# Generalized Net Model of Selection Operator of Genetic Algorithms

Olympia Roeva, Tania Pencheva, Krassimir Atanasov  
Centre of Biomedical Engineering  
Bulgarian Academy of Sciences  
Sofia, Bulgaria  
[olympia@clbme.bas.bg](mailto:olympia@clbme.bas.bg), [tania.pencheva@clbme.bas.bg](mailto:tania.pencheva@clbme.bas.bg),  
[krat@bas.bg](mailto:krat@bas.bg)

Anthony Shannon  
Warrane College  
The University of New South Wales  
Kensington, Australia  
[t.shannon@warrane.unsw.edu.au](mailto:t.shannon@warrane.unsw.edu.au)

**Abstract:** The apparatus of Generalized Nets is applied here to a description of a selection operator, which is one of the basic genetic algorithm operators. This genetic operator performs a probabilistic selection based upon the individual's fitness such that the better individuals have an increased chance of being selected for the next generation. The resulting GN model could be considered as a separate module, but can also be accumulated into a GN model to describe a whole genetic algorithm.

**Keywords:** Generalized net, Genetic algorithms, Selection operator

## I. INTRODUCTION

Genetic Algorithms (GA) are an adaptive heuristic search algorithm based on the evolutionary ideas of natural selection and genetics [7, 9]. The basic techniques of GA are designed to simulate processes in natural systems necessary for evolution, especially those follow the principles first laid down by Charles Darwin of "survival of the fittest". GA require only information concerning the quality of the solution produced by each parameter set (objective function value information). This characteristic differ GA from the optimization methods that require derivative information or, worse yet, complete knowledge of the problem structure and parameters. Since GA do not require such problem-specific information, they represent an intelligent exploitation of a random search used to solve optimization problems in a large domain of problems.

GA are implemented in a computer simulation in which a population of abstract representations (called *chromosomes* or the *genotype of the genome*) of candidate solutions (called *individuals*, *creatures*, or *phenotypes*) to an optimization problem evolves toward better solutions. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the *fitness* of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory

fitness level has been reached for the population. Once the genetic representation and the fitness function are defined, GA proceed to initialize a population of solutions *randomly*, then improve it through repetitive application of *mutation*, *crossover*, *inversion* and *selection* operators.

GA are quite popular and are applied in many domains – industrial design, scheduling, network design, routing, time series prediction, database mining, control systems, artificial life systems, as well as in many fields of science [5, 8÷10, 18]. On the other hand, until now the apparatus of Generalized Nets (GN) has been used as a tool for the description of parallel processes in several areas – economics, transport, medicine, computer technologies, etc. [1÷3, 12÷17]. That is why the idea of application of GN to GA description has intuitively appeared. Only a few GN models regarding genetic algorithm performance have been developed until now [1, 3, 12÷17]. In [1, 3] a GN model for genetic algorithms learning is proposed. The GN model in [16] describes the selection of genetic algorithm operators. The model has the possibility to test different groups of the defined genetic algorithm operators and to choose the most appropriate combination between them. The developed GN executes an GA and realizes tuning of the genetic operators, as well as of the fitness function, for the considered problem. The GN model in [17] describes the genetic algorithm search procedure. The model simultaneously evaluates several fitness functions, ranks the individuals according to their fitness and has the opportunity to choose the best fitness function regarding to specific problem domain. In [12÷15] the basic genetic algorithms operators – correspondingly *selection*, *crossover* and *mutation* are described using GN. Different types of crossover, namely *one-*, *two-point* crossover, as well as "*cut and splice*" techniques, are described in details in [14]. GN model, presented in [15], describes the mutation operator of the Breeder genetic algorithm. The GN model of a *roulette wheel selection method* as one of the widely used selection functions has been developed and presented in [12], while the GN model of a *stochastic universal sampling* is presented in [13]. The purpose of this investigation is to be developed a GN model, depicted the algorithm of *selection* of which individuals to pass on their genes to the next generation. This GN model intends to

incorporate the selection previously made by the *roulette wheel selection method* or the *stochastic universal sampling* thus appearing as an upper level in the description of a whole genetic algorithm.

## II. SELECTION OPERATOR

The selection of individuals to produce successive generations plays an extremely important role in a genetic algorithm. A probabilistic selection is performed based upon the individual's fitness such that the better individuals have an increased chance of being selected. An individual in the population can be selected more than once with all individuals in the population having a chance of being selected to reproduce into the next generation.

During each successive generation, a proportion of the existing population is selected to breed a new generation. Individual solutions are selected through a fitness-based process, where fitter solutions (as measured by a fitness function) are typically more likely to be selected. Certain selection methods rate the fitness of each solution and preferentially select the best solutions. Other methods rate only a random sample of the population, as this process may be very time-consuming.

Most functions are stochastic and designed so that a small proportion of less fit solutions are selected. This helps keep the diversity of the population large, preventing premature convergence on poor solutions. There are many methods how to select the best individuals, i.e. *roulette wheel selection*, *Boltzman selection*, *tournament selection*, *rank selection*, *steady state selection* and some others [4]. Among the most popular and well-studied selection methods are *roulette wheel selection* and *tournament selection*. The selection method is a user-defined parameter of the whole procedure of selection.

Fig. 1 presents the Matlab code from *Genetic algorithms Toolbox* [6, 11] of the function *selection* (*select.m*).

## III. GN MODELS OF SELECTION OPERATOR

The GN model depicted the algorithm of *selection*, as described in the function *select.m* [6, 11], is presented in Fig. 2.

The token  $\alpha$  enters GN in place  $l_1$  with an initial characteristic "pool of possible parents". The token  $\alpha$  is split into new tokens  $\delta$ ,  $\varepsilon$ ,  $\rho$ ,  $\beta$ ,  $\sigma$  and  $\gamma$  which obtain correspondingly following characteristics:

- $\delta$  in place  $l_2$  – "rate of individuals to be selected (*GGAP*)";
- $\varepsilon$  in place  $l_3$  – "identification of the population size (*Nind*)";
- $\rho$  in place  $l_4$  – "number of subpopulations (*SUBPOP*)";
- $\beta$  in place  $l_5$  – "fitness values of the individuals in the population (*FitnV*)";
- $\sigma$  in place  $l_6$  – "name of the selection function (*SEL\_F*)";
- $\gamma$  in place  $l_7$  – "individuals (parents) of the current population (*Chrom*)".

```
% SELECT.M (universal SELECTION)
%
% This function performs universal selection. The function handles
% multiple populations and calls the low level selection function
% for the actual selection process.
%
% Syntax: SelCh = select(SEL_F, Chrom, FitnV, GGAP, SUBPOP)
%
% Input parameters:
% SEL_F - Name of the selection function
% Chrom - Matrix containing the individuals (parents) of the
%         current population. Each row corresponds to one
%         individual.
% FitnV - Column vector containing the fitness values of the
%         individuals in the population.
% GGAP - (optional) Rate of individuals to be selected
%         if omitted 1.0 is assumed
% SUBPOP - (optional) Number of subpopulations
%          if omitted 1 subpopulation is assumed
%
% Output parameters:
% SelCh - Matrix containing the selected individuals.

% Author: Hartmut Pohlheim
% History: 10.03.94 file created

function SelCh = select(SEL_F, Chrom, FitnV, GGAP, SUBPOP);

% Check parameter consistency
if nargin < 3, error('Not enough input parameter'); end

% Identify the population size (Nind)
[NindCh, Nvar] = size(Chrom);
[NindF, VarF] = size(FitnV);
if NindCh ~= NindF, error('Chrom and FitnV disagree'); end
if VarF ~= 1, error('FitnV must be a column vector'); end

if nargin < 5, SUBPOP = 1; end
if nargin > 4,
    if isempty(SUBPOP), SUBPOP = 1;
    elseif isnan(SUBPOP), SUBPOP = 1;
    elseif length(SUBPOP) ~= 1, error('SUBPOP must be a scalar');
    end
end

if (NindCh/SUBPOP) ~= fix(NindCh/SUBPOP),
    error('Chrom and SUBPOP disagree'); end
Nind = NindCh/SUBPOP;

if nargin < 4, GGAP = 1; end
if nargin > 3,
    if isempty(GGAP), GGAP = 1;
    elseif isnan(GGAP), GGAP = 1;
    elseif length(GGAP) ~= 1, error('GGAP must be a scalar');
    elseif (GGAP < 0),
        error('GGAP must be a scalar bigger than 0'); end
end

% Compute number of new individuals (to select)
NSel = max(floor(Nind*GGAP + .5), 2);

% Select individuals from population
SelCh = [];
for irun = 1:SUBPOP,
    FitnVSub = FitnV((irun - 1)*Nind + 1:irun*Nind);
    ChrIx = feval(SEL_F, FitnVSub, NSel) + (irun - 1)*Nind;
    SelCh = [SelCh; Chrom(ChrIx, :)];
end
% End of function
```

Figure 1. Matlab function *select.m*

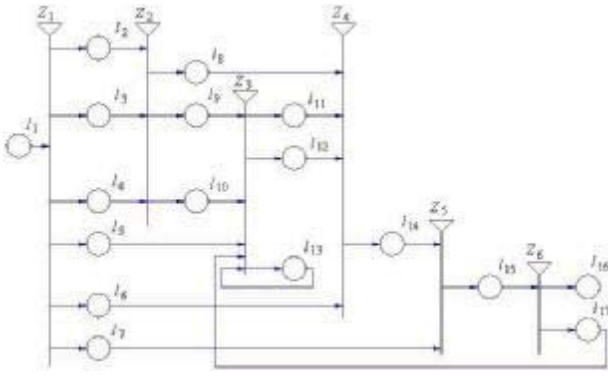


Figure 2. GN model of selection operator

The form of the first transition of the GN model is as follows:

$$Z_1 = \langle \{l_1\}, \{l_2, l_3, l_4, l_5, l_6, l_7\}, r_1, \wedge(l_1) \rangle$$

	$l_2$	$l_3$	$l_4$	$l_5$	$l_6$	$l_7$
$r_1 = l_1$	true	true	true	true	true	true

The tokens  $\delta$  and  $\varepsilon$  are combined in a new token  $\eta$  in place  $l_8$  with a characteristic “number of individuals to be selected ( $Nsel$ )”. The token  $\varepsilon$  keeps its characteristic “identification of the population size ( $Nind$ )” in place  $l_9$ , while the token  $\rho$  obtains a new characteristic “cycle index  $irun = 1:SUBPOP$ ” in place  $l_{10}$ .

The form of the second transition of the GN model is as follows:

$$Z_2 = \langle \{l_2, l_3, l_4\}, \{l_8, l_9, l_{10}\}, r_2, \wedge(l_2, l_3, l_4) \rangle,$$

	$l_8$	$l_9$	$l_{10}$
$r_2 = l_2$	true	false	false
$l_3$	true	true	false
$l_4$	false	false	true

Further, the tokens  $\varepsilon$ ,  $\rho$  and  $\beta$  are combined in a new token  $\varphi$  in place  $l_{12}$  with a characteristic “calculation of the function  $FitnVSub = FitnV((irun - 1)*Nind + 1:irun*Nind)$ ”. The token  $\varepsilon$  keeps its characteristic “identification of the population size ( $Nind$ )” in place  $l_{11}$ . The token  $\rho$  obtains a new characteristic “increment of the cycle index  $irun$ ” in place  $l_{13}$ , until  $irun$  becomes bigger than  $SUBPOP$ . The form of the third transition of the GN model is as follows:

$$Z_3 = \langle \{l_5, l_9, l_{10}, l_{13}, l_{17}\}, \{l_{11}, l_{12}\}, r_3, \wedge(l_5, l_9, l_{10}, l_{13}, l_{17}) \rangle$$

	$l_{11}$	$l_{12}$
$r_3 = l_5$	false	true
$l_9$	true	true
$l_{10}$	false	true
$l_{13}$	false	$W_{13,12}$
$l_{17}$	false	$W_{17,12}$

where  $W_{13,12} = \text{“}irun \leq SUBPOP\text{”}$ ,

$W_{17,12} = \text{“}irun < SUBPOP\text{”}$ .

Further, the tokens  $\varphi$ ,  $\sigma$ ,  $\eta$ ,  $\varepsilon$  and  $\rho$  are combined in a new token  $\theta$  in place  $l_{14}$  with a characteristic “calculation of the function

$$ChrIx = feval(SEL\_F, FitnVSub, Nsel) + (irun - 1)*Nind\text{”}.$$

The form of the fourth transition of the GN model is as follows:

$$Z_4 = \langle \{l_6, l_8, l_{11}, l_{12}\}, \{l_{14}\}, r_4, \wedge(l_6, l_8, l_{11}, l_{12}) \rangle$$

	$l_{14}$
$r_4 = l_6$	true
$l_8$	true
$l_{11}$	true
$l_{12}$	true

The tokens  $\theta$  and  $\gamma$  are combined in a new token  $\omega$  in place  $l_{15}$  with a characteristic “calculation of the function  $SelCh$ ”. The form of the fifth transition of the GN model is as follows:

$$Z_5 = \langle \{l_7, l_{14}\}, \{l_{15}\}, r_5, \wedge(l_7, l_{14}) \rangle$$

	$l_{15}$
$r_5 = l_7$	true
$l_{14}$	true

The token  $\omega$  obtains characteristics “selected individuals for next generation” in place  $l_{16}$  and “selection of individuals from population” in place  $l_{17}$ . The form of the sixth transition of the GN model is as follows:

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

$$r_6 = l_{15} \left| \begin{array}{cc} l_{16} & l_{17} \\ \hline W_{15,16} & W_{15,17} \end{array} \right.$$

where  $W_{15,16} = \text{“}i_{run} = SUBPOP\text{”}$ ;  
 $W_{15,17} = \neg W_{15,16}$ .

In the place  $l_{16}$  the selection of individuals from population is completely fulfilled. The GN model of the *selection* operator presented here could be considered as a separate module, but can also be collected into a GN model to describe a whole genetic algorithm.

#### IV. ANALYSIS AND CONCLUSIONS

The theory of Generalized Nets has been applied here to a description of one of the basic operators of genetic algorithms, namely the *selection* operator. A GN model developed in this paper depicts the algorithm of *selection* of which individuals to pass on their genes to the next generation. Such a GN model could be considered as a separate module, but also can be accumulated into one GN model for a description of a whole genetic algorithm, together with GN models of two other basic genetic operators *crossover* and *mutation*. The final aim is to produce GNs which are universal for all genetic algorithms to be constructed.

#### ACKNOWLEDGMENT

This work is supported by National Science Fund of Bulgaria by grant DID-02-29 “Modeling Processes with Fixed Development Rules (ModProFix)”.

#### REFERENCES

- [1] H. Aladjov, and K. Atanassov, “A generalized net for genetic algorithms learning”, Proc. of the XXX Spring Conference of the Union of Bulgarian Mathematicians, Borovets, pp. 242-248, April 2001.
- [2] K. Atanassov, Generalized nets, World Scientific, Singapore, New Jersey, London, 1991.
- [3] K. Atanassov, and H. Aladjov, Generalized nets in artificial intelligence, Vol. 2: Generalized Nets and Machine Learning, Sofia: Prof. M. Drinov Academic Publishing House, 2000.
- [4] J. Baker, “Reducing bias and inefficiency in the selection algorithm”, Proceedings of the Second International Conference on Genetic Algorithms and their Application, Hillsdale, New Jersey, pp. 14-21, 1987.
- [5] R. Bies, M. Muldoon, B. Pollock, S. Manuck, G. Smith, and M. Sale, “A genetic algorithm-based, hybrid machine learning approach to model selection”, Journal of Pharmacokinetics and Pharmacodynamics, pp. 196-221, 2006.
- [6] A. Chipperfield, P. J. Fleming, H. Pohlheim, and C. M. Fonseca, “Genetic algorithm toolbox for use with MATLAB”, 1993.
- [7] L. Davis, Handbook of genetic algorithms, Van Nostrand Reinhold, 1991.
- [8] D. Fogel, Evolutionary computation: toward a new philosophy of machine intelligence, IEEE Press, NJ, 3rd ed., 2006.
- [9] D. E. Goldberg, Genetic algorithms in search, optimization and machine learning, Addison Wesley, 1989.
- [10] C. Houck, J. Joines, and M. Kay, “A genetic algorithm for function optimization: a MATLAB implementation”, NCSU-IE TR 95-05, 1995.
- [11] MathWorks, Genetic algorithm toolbox user’s guide for MATLAB.
- [12] T. Pencheva, K. Atanassov, and A. Shannon, “Modelling of a roulette wheel selection operator in genetic algorithms using generalized nets”, Int. J. Bioautomation, vol 13(4), pp. 257-264, 2009.
- [13] T. Pencheva, K. Atanassov, and A. Shannon, “Modelling of a stochastic universal sampling selection operator in genetic algorithms using generalized nets”, Proceedings of the Tenth International Workshop on Generalized Nets, Sofia, pp. 1-7, December 2009.
- [14] T. Pencheva, O. Roeva, and A. Shannon, “Generalized net models of crossover operator of genetic algorithm”, Proceedings of Ninth International Workshop on Generalized Nets, Sofia, vol. 2, pp. 64-70, July 2008.
- [15] O. Roeva, and A. Shannon, “A generalized net model of mutation operator of the breeder genetic algorithm”, Proceedings of Ninth International Workshop on Generalized Nets, Sofia, vol. 2, pp. 59-63, July 2008.
- [16] O. Roeva, K. Atanassov, and A. Shannon, “Generalized net for selection of genetic algorithm operators”, Annual of “Informatics” Section, Union of Scientists in Bulgaria, vol. 1, pp. 117-126, 2008.
- [17] O. Roeva, K. Atanassov, and A. Shannon, “Generalized net for evaluation of the genetic algorithm fitness function”, Proceedings of the Eighth International Workshop on Generalized Nets, Sofia, pp. 48-55, June 2007.
- [18] A. M. S. Zalzala, and P. J. Fleming (Eds.), Genetic algorithms in engineering systems, Institution of Electrical Engineers, London, 1997.