Tenth Int. Workshop on Generalized Nets
Sofia, 5 December 2009, 1-7

Modelling of a Stochastic Universal Sampling Selection Operator in Genetic Algorithms Using Generalized Nets

Tania Pencheva[1], Krassimir Atanassov[1], Anthony Shannon[2]
[1] – Centre of Biomedical Engineering, Bulgarian Academy of Sciences
105 Acad. G. Bonchev Str., 1113 Sofia, Bulgaria
e-mails: *tania.pencheva@clbme.bas.bg*, *krat@bas.bg*
[2] – Warrane College, University of New South Wales
Kensington, 1465, Australia
e-mail: *t.shannon@warrane.unsw.edu.au*

Abstract: The apparatus of Generalized Nets (GNs) is applied here to a description of a selection operator, which is one of the basic genetic algorithm operators. The GN model presented here describes one of the most widely used selection algorithms in current GA, namely stochastic universal sampling. 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: GNs, Genetic algorithms, Selection, Stochastic universal sampling

1 Introduction

Genetic Algorithms (GA) are an adaptive heuristic search algorithm based on the evolutionary ideas of natural selection and genetics. They represent an intelligent exploitation of a random search used to solve optimization problems. 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 is based on an analogy with the genetic structure and behaviour of chromosomes within a population of individuals.

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 proceeds 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, 7, 8]. 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, 13÷16]. That is why the idea of application of GN to GA description has intuitively appeared. Until now only a few GN models regarding genetic algorithm performance have been developed [1, 3, 13÷16]. A GN model for genetic algorithms learning was proposed in [1, 3]. The GN model in [15] 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 execute an GA and realize tuning of the genetic operators, as well as of the fitness function, for the considered problem. The genetic algorithm search procedure is described with the GN model in [16]. The 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. In [12, 13, 14] 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 [13]. GN model, presented in [14], describes the mutation operator of the Breeder genetic algorithm. 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 fittness 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. There are several schemes for the selection process: *roulette wheel selection* and its extensions, *scaling techniques*, *tournament*, *elitist models*, and *ranking methods* [9, 10, 17]. Widely the used Matlab Toolbox for Genetic algorithms [6, 11] contains two functions for the selection function, namely the *roulette wheel selection method* and the *stochastic universal sampling*. Since in [12] a GN model of a *roulette wheel selection method* is developed, the goal of this investigation is to develop a GN model of a *stochastic universal sampling*.

2 Stochastic Universal Sampling

Stochastic Universal Sampling (*SUS*) developed by Baker [4] is a single-phase sampling algorithm with minimum spread and zero bias. Instead of a single selection pointer employed in *roulette wheel* methods, *SUS* uses N equally spaced pointers, where N is the number of selections required. The population is shuffled randomly and a single random number *pointer1* in the range [0, $1/N$] is generated. The N individuals are then chosen by generating the N pointers, starting with *pointer1* and spaced by $1/N$, and selecting the individuals whose fitness spans the positions of the pointers. If $et(i)$ is the expected number of trials of individual i, $\lfloor et(i) \rfloor$ is the floor of $et(i)$ and $\lceil et(i) \rceil$ is the ceiling, an individual is thus guaranteed to be selected a minimum of times $\lfloor et(i) \rfloor$ and no more than $\lceil et(i) \rceil$, thus achieving minimum spread. In addition, as individuals are selected entirely on their positions in the population, *SUS* has zero bias. For these reasons, *SUS* has become one of the most widely used selection algorithms in current GA.

Figure 1 demonstrates the *stochastic universal sampling*. The individuals are mapped to contiguous segments of a line, such that each individual's segment is equal in size to its fitness exactly as in *roulette wheel selection*. Equally spaced pointers are placed over the line as many as there are individuals to be selected (N). For 6 individuals ($N = 6$) to be selected, the distance between the pointers is 1/6 = 0.167. Figure 1 shows the selection for the sample of the random number 0.1 in the range [0, 0.167].

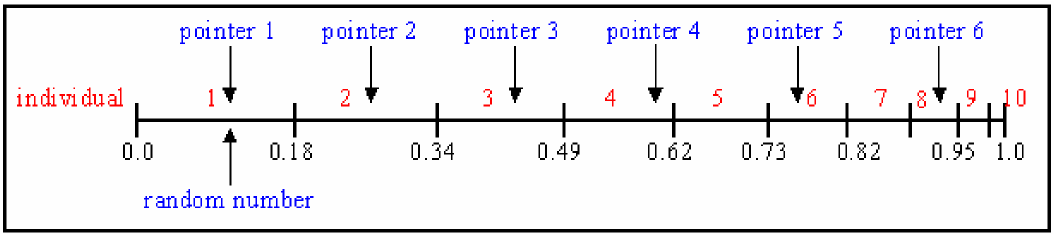

Figure 1: Stochastic universal sampling

After selection the mating population consists of the individuals 1, 2, 3, 4, 6 and 8. *Stochastic universal sampling* ensures a selection of offspring which is closer to what is deserved than *roulette wheel selection*.

3 GN Models of Stochastic Universal Sampling

Figure 2 presents the code of implemented *stochastic universal sampling* (*sus.m*) selection function in Matlab Toolbox for Genetic algorithms [6, 11]:

```
% SUS.M          (Stochastic Universal Sampling)
%
% This function performs selection with STOCHASTIC UNIVERSAL SAMPLING.
%
% Syntax:  NewChrIx = sus(FitnV, Nsel)
%
% Input parameters:
%    FitnV     - Column vector containing the fitness values of the
%                individuals in the population.
%    Nsel      - number of individuals to be selected
%
% Output parameters:
%    NewChrIx  - column vector containing the indexes of the selected
%                individuals relative to the original population, shuffled.
%                The new population, ready for mating, can be obtained
%                by calculating OldChrom(NewChrIx,:).

% Author:     Hartmut Pohlheim (Carlos Fonseca)
% History:    12.12.93     file created
%             22.02.94     clean up, comments

function NewChrIx = sus(FitnV,Nsel);

% Identify the population size (Nind)
   [Nind,ans] = size(FitnV);

% Perform stochastic universal sampling
   cumfit = cumsum(FitnV);
   trials = cumfit(Nind) / Nsel * (rand + (0:Nsel-1)');
   Mf = cumfit(:, ones(1, Nsel));
   Mt = trials(:, ones(1, Nind))';
  [NewChrIx, ans] = find(Mt < Mf & [zeros(1, Nsel); Mf(1:Nind-1, :)] <= Mt);

% Shuffle new population
   [ans, shuf] = sort(rand(Nsel, 1));
   NewChrIx = NewChrIx(shuf);

% End of function
```

Figure 2: Matlab function sus.m

The GN model, described the *stochastic universal sampling*, as described in the function *sus.m*, is presented in Figure 3:

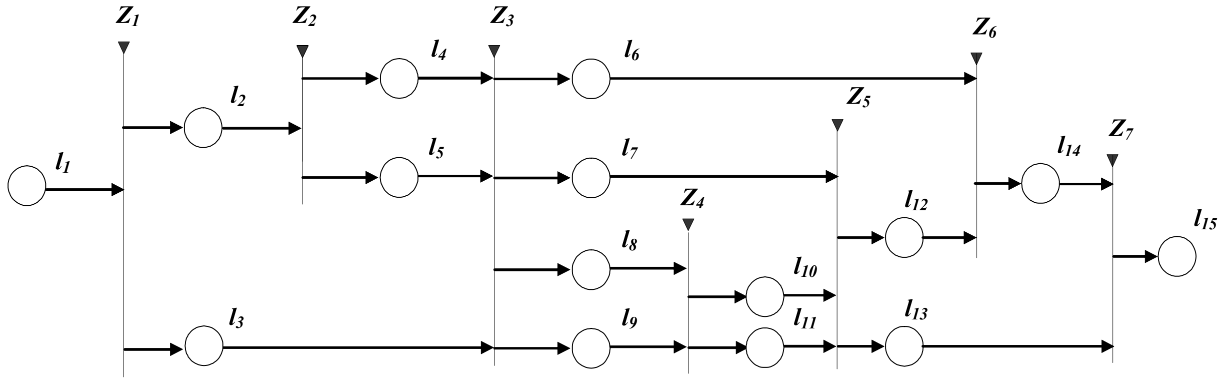

Figure 3: GN model of stochastic universal sampling

The token α enters GN in place l_1 with an initial characteristic "pool of possible parents". The token α is split into new tokens β and γ, which obtain correspondingly characteristics "fitness values of the individuals in the population (*FitnV*)" in place l_2 and "number of individuals to be selected (*Nsel*)" in place l_3. The form of the first transition of the GN model is as follows:

$$Z_1 = <\{l_1\}, \{l_2, l_3\}, r_1 = \begin{array}{c|cc} & l_2 & l_3 \\ \hline l_1 & true & true \end{array}, \wedge(l_1)>$$

The token β is split into new tokens δ and ε, which obtain correspondingly characteristics "calculation of the function *cumfit* = *cumsum(FitnV)*" in place l_4 and "identify the population size (*Nind*)" in place l_5. The form of the second transition of the GN model is as follows:

$$Z_2 = <\{l_2\}, \{l_4, l_5\}, r_2 = \begin{array}{c|cc} & l_4 & l_5 \\ \hline l_2 & true & true \end{array}, \wedge(l_2)>$$

Further, the tokens δ and γ are combined in a new token φ in place l_6 with a characteristic "calculation of the function *Mf* = *cumfit(:, ones(1, Nsel))*". The token ε keeps its characteristic "identify the population size (*Nind*)" in place l_7. The tokens δ and ε are combined in a new token η in place l_8 with a characteristic "calculation of the function *cumfit(Nind)"*. The token γ keeps its characteristic "number of individuals to be selected (*Nsel*)" in place l_9. The form of the third transition of the GN model is as follows:

$$Z_3 = <\{l_3, l_4, l_5\}, \{l_6, l_7, l_8, l_9\}, r_3, \wedge(l_3, l_4, l_5)>$$

$$r_3 = \begin{array}{c|cccc} & l_6 & l_7 & l_8 & l_9 \\ \hline l_3 & true & false & false & true \\ l_4 & true & false & true & false \\ l_5 & false & true & true & false \end{array}$$

The tokens η and γ are combined in a new token λ in place l_{10} with a characteristic "calculation of the function *trials = cumfit(Nind) / Nsel * (rand + (0:Nsel-1)')"*. The token γ obtains a new characteristic "*rand*(*Nsel*)" in place l_{11}. The form of the fourth transition of the GN model is as follows:

$$Z_4 = <\{l_8, l_9\},\ \{l_{10}, l_{11}\}, r_4, \wedge(l_8, l_9) >$$

$$r_4 = \begin{array}{c|cc} & l_{10} & l_{11} \\ \hline l_8 & true & false \\ l_9 & true & true \end{array}$$

The tokens ε and λ are combined in a new token θ in place l_{12} with a characteristic "calculation of the function *Mt = trials(:, ones(1, Nind))*". The token γ obtains a new characteristic "*sort*(*rand*(*Nsel*))" in place l_{13}. The form of the fifth transition of the GN model is as follows:

$$Z_5 = <\{l_7, l_{10}, l_{11}\},\ \{l_{12}, l_{13}\}, r_5, \wedge(l_7, l_{10}, l_{11}) >$$

$$r_5 = \begin{array}{c|cc} & l_{12} & l_{13} \\ \hline l_7 & true & false \\ l_{10} & true & false \\ l_{11} & false & true \end{array}$$

The tokens φ and θ are combined in a new token ω in place l_{14} with a characteristic "calculation of the function [*NewChrIx, ans*] = *find (Mt < Mf &* [*zeros(1, Nsel); Mf(1:Nind-1, :)*] *≤ Mt)*". The form of the sixth transition of the GN model is as follows:

$$Z_6 = <\{l_6, l_{12}\},\ \{l_{14}\}, r_6, \wedge(l_6, l_{12}) >$$

$$r_6 = \begin{array}{c|c} & l_{14} \\ \hline l_6 & true \\ l_{12} & true \end{array}$$

The tokens ω and γ are combined in a new token σ in place l_{15} with a characteristic "shuffle new population *NewChrIx = NewChrIx(shuf)*". The form of the seventh transition of the GN model is as follows:

$$Z_7 = <\{l_{13}, l_{14}\},\ \{l_{15}\}, r_7, \wedge(l_{13}, l_{14}) >$$

$$r_7 = \begin{array}{c|c} & l_{15} \\ \hline l_{13} & true \\ l_{14} & true \end{array}$$

In the place l_{15} the new chromosome is created and the selection function, performing *stochastic universal sampling*, 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.

4 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 of one of the mostly used selection functions, the *stochastic universal sampling*, has been developed in this paper. 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. In future it is planed to be constructed some GNs to represent the functioning of results of the work of whole genetic algorithms. The final aim is to produce GNs which are universal for all genetic algorithms to be constructed.

References

[1] Aladjov H., K. Atanassov, A Generalized Net for Genetic Algorithms Learning, *Proceedings of the XXX Spring Conference of the Union of Bulgarian Mathematicians*, Borovets, Bulgaria, April 8-11, 2001, 242-248.

[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] Baker J., Reducing Bias and Inefficiency in the Selection Algorithm, *Proceedings of the Second International Conference on Genetic Algorithms and their Application*, Hillsdale, New Jersey, 1987, 14-21.

[5] Bies R., M. Muldoon, B. Pollock, S. Manuck, G. Smith, M. Sale, A Genetic Algorithm-based, Hybrid Machine Learning Approach to Model Selection, *Journal of Pharmacokinetics and Pharmacodynamics*, 2006, 196-221.

[6] Chipperfield A., P. J. Fleming, H. Pohlheim, C. M. Fonseca, *Genetic Algorithm Toolbox for Use with MATLAB*, 1993.

[7] Davis L., *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, 1991.

[8] Fogel D., *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, IEEE Press, NJ, Third Edition, 2006.

[9] Goldberg D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, 1989.

[10] Houck C., J. Joines, 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] Pencheva T., K. Atanassov, A. Shannon, Modelling of a Roulette Wheel Selection Operator in Genetic Algorithms Using Generalized Nets, *Bioautomation*, 2009, 13(4), 257-264.

[13] Pencheva T., O. Roeva, A. Shannon, Generalized Net Models of Crossover Operator of Genetic Algorithm, *Proceedings of Ninth International Workshop on Generalized Nets*, Sofia, Bulgaria, July 4, 2008, 2, 64-70.

[14] Roeva O., A. Shannon, A Generalized Net Model of Mutation Operator of the Breeder Genetic Algorithm, *Proceedings of Ninth International Workshop on Generalized Nets*, Sofia, Bulgaria, July 4, 2008, 2, 59-63.

[15] Roeva O., K. Atanassov, A. Shannon, Generalized Net for Selection of Genetic Algorithm Operators, *Annual of "Informatics" Section, Union of Scientists in Bulgaria*, 2008, 1, 117-126.

[16] Roeva, O., K. Atanassov, A. Shannon, Generalized Net for Evaluation of the Genetic Algorithm Fitness Function, *Proceedings of the Eighth International Workshop on Generalized Nets*, Sofia, Bulgaria, June 26, 2007, 48-55.

[17] Zalzala A. M. S., P. J. Fleming (Eds.), *Genetic Algorithms in Engineering Systems*, Institution of Electrical Engineers, London, 1997.