GENERALIZED NETS MODEL OF OFFSPRING REINSERTION IN GENETIC ALGORITHMS

Tania Pencheva¹, Krassimir Atanassov¹, Anthony Shannon²

¹ Institute of Biophysics and Biomedical Engineering – Bulgarian Academy of Sciences
105 Acad. G. Bonchev Str., Sofia 1113, Bulgaria
E-mails: tania.pencheva@biomed.bas.bg, krat@bas.bg

² Faculty of Engineering & IT, University of Technology
Sydney, NSW 2007, Australia
E-mail: tshannon38@gmail.com

Abstract: The apparatus of Generalized Nets is applied here to describe one of the basic functions in genetic algorithms, namely reinsertion. This function is responsible for an insertion of offspring into the current population, replacing parents with offspring and returning the resulting population. The resulting generalized net model could be considered as a separate module, but it can also be assembled into a generalized net model to describe a whole genetic algorithm.

Keywords: …

1. Introduction

Genetic Algorithms (GA) are an adaptive heuristic search algorithm [7]. GA simulate processes in natural systems necessary for evolution, following the principles of “survival of the fittest” formulated for first time by Charles Darwin. As a search technique, GA are implemented in a computer simulation in which a population of abstract representations (chromosomes) of candidate solutions (individuals) to an optimization problem evolves toward better solutions. Once the genetic representation and the fitness function are defined, GA proceed to initialize a population of solutions randomly. Once the offspring have been produced by selection, recombination and mutation of individuals from the old population, the fitness of the offspring is determined. If less offspring are produced than the size of the original population then to maintain the size of the original population, the offspring have to be reinserted into the old population. Similarly, if not all offspring are to be used at each generation or if more offspring are generated than the size of the old population then a reinsertion scheme must be used to determine which individuals are to exist in the 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.
Due to a variety of successive implementations of Generalized Nets (GN) theory for description of parallel processes in several areas [1-3], the idea of using GN for the description of GA has intuitively appeared. Up to now, a few GN models regarding genetic operators’ description have been developed [8-12]. There GN models are used to describe the basic genetic algorithms operators, namely selection, crossover and mutation. The GN model of a roulette wheel selection method, which is one of the widely used selection functions, has been developed in [9], while the GN model of a stochastic universal sampling is presented in [10]. A GN model allowing the user to choose between different selection functions have been elaborated in [8]. Different types of crossover, namely one-, two-point crossover, as well as “cut and splice” techniques, are described in details in [11]. The GN model, presented in [12], describes the mutation operator of the Breeder GA.

The purpose of the present investigation is to develop a GN model, which describes reinsertion of offspring in the population. Different schemes of global reinsertion exist [5], namely pure, uniform, elitist and fitness-based reinsertion. Since the developed here GN model is based on the Matlab code, user can choose between uniform and fitness-based reinsertion.

2. Reinsertion schemes

Reinsertion scheme is determined by the selection algorithm used [5, 6]. Thus, global reinsertion is used for all population based selection algorithm (roulette-wheel selection, stochastic universal sampling, truncation selection), while local reinsertion – for local selection. Here the interest is pointed to global reinsertion, which can occur in the following different schemes [5]:

- pure reinsertion – produce as many offspring as parents, and replace all parents by the offspring;
- uniform reinsertion – produce less offspring than parents, and replace parents uniformly at random;
- elitist reinsertion – produce less offspring than parents, and replace the worst parents;
- fitness-based reinsertion – produce more offspring than needed for reinsertion, and reinsert only the best offspring.

Pure reinsertion is the simplest reinsertion scheme. Every individual lives one generation only. This scheme is used in the simple genetic algorithm. However, it is very likely, that very good individuals are replaced without producing better offspring and thus, good information is lost. The elitist combined with fitness-based reinsertion prevents this losing of information and is the recommended method. At each generation, a given number of the least fit parents is replaced by the same number of the most fit offspring (Fig. 1, [5]).

The fitness-based reinsertion scheme implements a truncation selection between offspring before inserting them into the population (i.e. before they can participate in the reproduction process). On the other hand, the best individuals can live for many generations. However, with every generation some new individuals are inserted. It is not checked whether the parents are replaced by better or worse offspring.
Because parents may be replaced by offspring with a lower fitness, the average fitness of the population can decrease. However, if the inserted offspring are extremely bad, they will be replaced with new offspring in the next generation.

3. **GN model for reinsertion function**

The GN model standing for the *reinsertion function*, as described by function *reins.m* [4] in Matlab, is presented in Fig. 2.

The token $\alpha$ enters GN in place $l_1$ with an initial characteristic "parameters of GA".
Some of the most common considered parameters of GA are: number of individuals (NIND), maximal number of generations (MAXGEN), number of variables (NVAR), number of offspring to insert (NIns), number of offspring per subpopulation (NSel), objective values of the individuals (ObjVCh), etc.

The token \( \beta \) enters GN in place \( l_2 \) with an initial characteristic

“pool of possible parents”.

Tokens \( \alpha \) and \( \beta \) are combined and appear as a token \( \gamma \), which splits into four new tokens, respectively \( \gamma_1, \gamma_2, \gamma_3 \) and \( \gamma_4 \), respectively, in places \( l_3 \) to \( l_6 \) with the following characteristics:

- token \( \gamma_1 \) in place \( l_3 \) – “number of individuals (NIND)”;
- token \( \gamma_2 \) in place \( l_4 \) – “number of offspring to insert (NIns)”;
- token \( \gamma_3 \) in place \( l_5 \) – “number of offspring per subpopulation (NSel)”;
- token \( \gamma_4 \) in place \( l_6 \) – “objective values of the individuals (ObjVCh)”.

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

\[
Z_1 = \langle \langle l_1, l_2 \rangle, \langle l_3, l_4, l_5, l_6 \rangle, r_1, \wedge (l_1, l_2) \rangle,
\]

\[
\eta = \begin{bmatrix}
\text{True} & \text{True} & \text{True} & \text{True} \\
\text{True} & \text{True} & \text{True} & \text{True}
\end{bmatrix}.
\]

A new token \( \nu \) with a characteristic

“choice of reinsertion scheme – fitness-based or uniform reinsertion”

enters GN in place \( l_7 \). The token \( \gamma_1 \) splits into two new tokens \( \gamma_{11} \) and \( \gamma_{12} \), which obtain in place \( l_8 \) and \( l_{11} \) new characteristics, respectively:

- “number of individuals (NIND)”;
- “[Dummy, ChIx] = sort(rand(NIND, 1))”.

Token \( \gamma_2 \) splits into two new tokens \( \gamma_{21} \) and \( \gamma_{22} \). In place \( l_9 \), token \( \gamma_{21} \) keeps the characteristic of \( \gamma_2 \), namely

“number of offspring to insert (NIns)”.

In place \( l_{14} \), token \( \gamma_{22} \) obtains a new characteristic, namely

“OffIx = (1:NIns)”.

Token \( \gamma_3 \) keeps its characteristic in place \( l_{10} \), namely

“number of offspring per subpopulation (NSel)”.

Tokens \( \gamma_1 \) and \( \gamma_4 \) are combined and appear as a token \( \delta \) in place \( l_{12} \) with a characteristic

“[Dummy, ChIx] = sort(–ObjVCh((irun – 1)\*NIND + 1:irun\*NIND))”.

Tokens \( \gamma_3 \) and \( \gamma_4 \) are combined and appear as a token \( \varepsilon \) in place \( l_{13} \) with a characteristic

“[Dummy, OffIx] = sort(ObjVSel((irun – 1)\*NSEL + 1:irun\*NSEL))”.

The form of the second transition of the GN model is as follows:
\[ Z_2 = \langle \{ l_8, l_9, l_{10}, l_{11}, l_{12}, l_{13}, l_{14} \}, r_2, \wedge (l_5, l_6, l_7) \rangle, \]

\[
\begin{array}{cccccccc}
  r_2 &=& l_8 & l_9 & l_{10} & l_{11} & l_{12} & l_{13} & l_{14} \\
  l_3 &=& \text{True} & \text{False} & \text{False} & W_{3,11} & W_{3,12} & \text{False} & \text{False} \\
  l_4 &=& \text{False} & \text{True} & \text{False} & \text{False} & \text{False} & W_{4,14} \\
  l_5 &=& \text{False} & \text{False} & \text{True} & \text{False} & \text{False} & W_{5,13} & \text{False} \\
  l_6 &=& \text{False} & \text{False} & \text{False} & \text{False} & \text{False} & W_{6,12} & W_{6,13} & \text{False} \\
  l_7 &=& \text{False} & \text{False} & \text{False} & W_{7,11} & W_{7,12} & \text{False} & \text{False} \\
\end{array}
\]

where
- \( W_{5,11} = W_{7,11} = \text{“uniform reinsertion is chosen”} \)
- \( W_{5,12} = W_{6,12} = W_{7,12} = \text{“fitness-based reinsertion is chosen”} \)
- \( W_{5,13} = W_{6,13} = \text{“NIns < NSEL”} \)
- \( W_{4,14} = \neg W_{5,13} \)

Tokens \( \gamma_{11}, \gamma_{12}, \gamma_{21} \) and \( \delta \) are combined and appear as a token \( \pi \) in place \( l_{15} \) with a characteristic

\[ \text{“PopIx = ChIx(1:NIns)) + (irun - 1)*NIND”} \]

Tokens \( \gamma_{21}, \gamma_{5}, \epsilon \) and \( \gamma_{22} \) are combined and appear as a token \( \sigma \) in place \( l_{16} \) with a characteristic

\[ \text{“SelIx = OffIx(1:NIns)) + (irun - 1)*NSEL”} \]

Thus, the form of the third transition of the GN model is as follows:

\[ Z_3 = \langle \{ l_8, l_9, l_{10}, l_{11}, l_{12}, l_{13}, l_{14} \}, \{ l_{15}, l_{16} \}, r_3, \wedge (l_5, l_6, l_7, l_{10}, l_{11}, l_{12}, l_{13}, l_{14}) \rangle \]

\[
\begin{array}{cccc}
  r_3 &=& l_8 & l_9 \\
  l_5 &=& \text{True} & \text{False} \\
  l_9 &=& \text{True} & \text{True} \\
  l_{10} &=& \text{False} & \text{True} \\
  l_{11} &=& \text{True} & \text{False} \\
  l_{12} &=& \text{True} & \text{False} \\
  l_{13} &=& \text{False} & \text{True} \\
  l_{14} &=& \text{False} & \text{True} \\
\end{array}
\]

Tokens \( \pi \) and \( \sigma \) are then combined in a token \( \eta \) in place \( l_{17} \) with a characteristic

\[ \text{“Chrom(PopIx, :) = SelCh(SelIx, :)”} \]

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

\[ Z_4 = \langle \{ l_5, l_{10} \}, \{ l_{17} \}, r_4, \wedge (l_{15}, l_{16}) \rangle \]

\[
\begin{array}{c}
  r_4 = l_{15} \\\n  l_{15} &=& \text{True} \\
\end{array}
\]

After the insertion of offspring in subpopulation in place \( l_{17} \), the token \( \eta \) could pass to place \( l_{18} \) with a characteristic

\[ \text{“end of the genetic algorithm”} \]
or in place \( l_{19} \) with a characteristic

“new subpopulation”.

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

\[
Z_5 = \langle \{l_{17}\}, \{l_{18}, l_{19}\}, r_5 \rangle
\]

\[
r_5 = \frac{l_{18}}{l_{17}} \frac{l_{19}}{W_{17,18} \ W_{17,18}}
\]

where \( W_{17,18} \) = “end of the genetic algorithm”.

4. Analysis and conclusion

The theory of generalized nets has been here applied to describe one of the basic functions in genetic algorithms, namely \textit{reinsertion function}. The GN model affords the user to choose between \textit{fitness-based} or \textit{uniform reinsertion}. Such a GN model could be considered as a separate module, but can also be assembled into a single GN model for description of a whole genetic algorithm.

Acknowledgements

This work is supported by the National Science Fund of Bulgaria, Grant No. DID-02-29 “Modeling Processes with Fixed Development Rules (ModProFix)”.

References


