

# Robust Parallel Genetic Algorithms with Re-Initialisation

Ivan Sekaj

Department of Automatic Control Systems  
Faculty of Electrical Engineering and Information Technology  
Slovak University of Technology, Ilkovičova 3, 812 19 Bratislava  
Slovak Republic  
sekaj@kasr.elf.stuba.sk

**Abstract.** The influence of different parallel genetic algorithm (PGA) architectures on the GA convergence properties is analysed. Next, two proposed versions of these PGA architectures are compared – homogenous and heterogeneous. Finally the effect of re-initialisation in some partial populations on the PGA convergence has been analysed. The proposed PGA modifications are useful mainly in case of non-smooth cost function optimisation.

## 1 Introduction

Important factors, which influence the convergence properties of the genetic algorithm (GA) are the selective pressure and population diversity [10],[1],[6]. Selective pressure is a measure of preferring currently better individuals from the population of a particular GA for the currently worse ones. Algorithms with a high selective pressure are characterized by fast convergence, but generally to some local optimum. The population diversity is a measure of gene dissimilarity and with its increase it is possible to unbend the algorithm from the current local optimum and redirect it to better solutions, if possible to the global optimum. A too low diversity, but on the other hand also a too high one can slow down the convergence rate. Increase of the selective pressure can be influenced using such selection methods, which prefer highly fit individuals. Opposite to this, population diversity can be increased by such selection methods, which contain more randomness, regardless of fitness considering. Other important factor, which increases diversity, is the increasing effect of operators modifying the current individuals in the population (e.g. mutation).

In the GA's with a simple population, the selective pressure and population diversity act against each another. For a good GA convergence it is important to find equilibrium between the selective pressure and the population diversity. Note, that the equilibrium depends on the solved problem. This equilibrium can change by introducing multiple populations in parallel genetic algorithms (PGA). By this way it is possible to increase the selective pressure in some populations and simultaneously to increase the population diversity in other populations. In the presented contribution the influence of using various PGA architectures, various genetic operations and population re-initialisation on the PGA convergence are analysed.

## 2 Parallel GA architectures

In the literature, different PGA architecture types and also different viewpoints of their classification can be found [3],[5]. Some authors are using the division to fine-grained and coarse-grained PGA's [2].

For our analysis 6 representative PGA architectures have been selected, each with 9 populations (Fig.1A-F). In all cases a periodical migration of one individual from each population into other populations has been considered. That means that in each migration period (say each  $n/10$  or  $n/100$  generations,  $n$  is the generation number) the best individual of each of the 9 populations is selected and its copy is sent to the specified populations according to the defined migration structure. In the architecture types A and B a grid PGA structure with the communication of the neighbour population has been used. The type C is a hierarchical structure with 8 low-level nodes and 1 upper-level node. In the case D the ring architecture has been used. The cases E and F have three level hierarchical structures with different linkage between the low and middle levels. Additionally, the types A and B are using a bi-directional migration while C, D, E and F use only a unidirectional one.

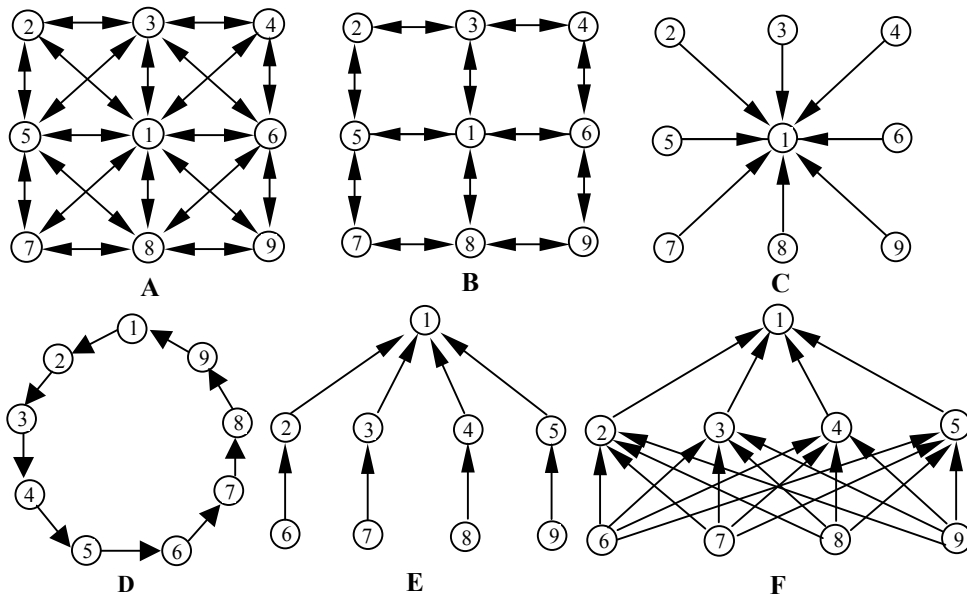


Fig.1 Different PGA architecture types

The GA scheme, which has been used in our experiments in each population of the PGA with some modifications (which will be explained later) and also in the simple population GA (SGA), is as follows:

1. Initialisation of the population (randomly) and fitness calculation.

2. Selection of best individuals (one or more), which are without any change copied into the new population – Pop1. Random selection of a group of individuals, which are copied without any change, into the new population – Pop2.
3. Selection of parents, which will be modified by genetic operations crossover and mutation – Pop3.
4. Mutation of the group Pop3.
5. Crossover of the group Pop3.
6. Completion of the new population by unification of the groups Pop1, Pop2 and Pop3.
7. New population fitness calculation. Test of terminating condition, if not fulfilled, then jump to the step 2.

In our analysis, the above scheme has been used, with following modifications. From the selection method in the step 3 the tournament selection [4],[7],[8] the linear ranking selection and the random selection (random selection of strings, without considering fitness) have been used. Next, two mutation versions have been used. In the global mutation the mutated gene is replaced by a random (real) value from the entire search space with uniform probability distribution. In the local mutation there are small random real numbers added to (or subtracted from) the original gene values. The used size of random changes was between 1% and 20 % of the entire search space, with a uniform distribution and a zero mean value. In case of crossover, a simple one-point crossover or an intermediate crossover known from Evolution strategies [9],[6] have been used. The size of each population was set to 20 strings and real coding of strings has been used. More details about the used operations will be specified in Section 2.2.

In order to demonstrate the analysis results, let us use the following test functions. The first function  $f_1$  consists of three partial functions – quadratic, Schwefel function and 3 Gauss peaks (holes)

$$f_1(x) = x_1^2 + x_2^2 + x_3^2 \quad \text{if } -500 > x_i > 500; \quad i=1,2,3$$

*otherwise*

$$\begin{aligned} f_1(x) = & \frac{1}{2} \sum_{i=1}^3 \left( -x_i \sin(\sqrt{|x_i|}) \right) - 800 \exp \left( -\frac{x_1^2}{1500} - \frac{x_2^2}{1500} - \frac{x_3^2}{1500} \right) - \\ & - 400 \exp \left( -\frac{(x_1^2 - 200)}{1500} - \frac{(x_2^2 - 200)}{1500} - \frac{(x_3^2 - 200)}{1500} \right) - \\ & - 300 \exp \left( -\frac{(x_1^2 + 300)}{1500} - \frac{(x_2^2 + 300)}{1500} - \frac{(x_3^2 + 300)}{1500} \right) \end{aligned}$$

The global minimum is  $f_1(x_1, x_2, x_3) = -800$ ;  $x_1 = x_2 = x_3 = 0$ . This function belongs to the category of “deceptive functions”, where the search for the global optimum is not an easy problem, because of the “unexpected” position of the global optimum. Graph of  $f_1$  for the two variable case is in Fig.2.

The second test function is the “Egg holder function”

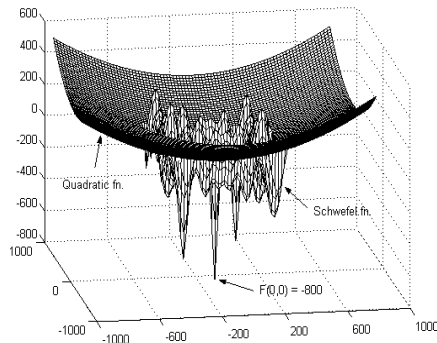
$$f_2(X) = \sum_{i=1}^{n-1} \left( -x_i \sin\left(\sqrt{|x_i - (x_{i+1} + 47)|}\right) - (x_{i+1} + 47) \sin\left(\sqrt{|x_{i+1} + 47 + \frac{x_i}{2}|}\right) \right)$$

A graph for the two variable case  $f_2(x_1, x_2)$  is in Fig.3. It is a multi-modal function. The global minimum for the used 10 variable case is near the value  $f_2(x) = -8247$ ;  $x = [440 \ 455 \ 470 \ 426 \ 441 \ 455 \ 471 \ 426 \ 442 \ 456]$ .

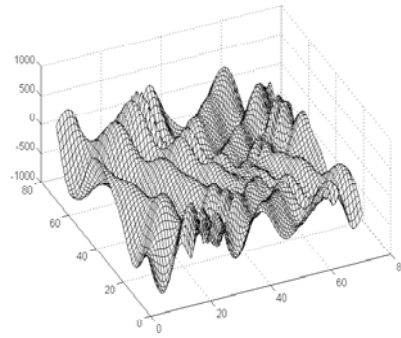
The last used test function is the Griewangk function

$$f_3(x) = \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 ; \quad -600 > x_i > 600 ; \quad i=1,2,\dots,7$$

with the global minimum  $x_i^* = 0 ; i=1,2,\dots,n$  and the value  $f_3(x^*) = 0$ .



**Fig.2** Graph of function  $f_1$



**Fig.3** Graph of function  $f_2$  - "Egg holder"

Let us consider four different situations, which will be analysed in more detail:

1. The GA's in all 9 partial populations are identical, identical GA operations and parameters are used – “homogenous PGA”.
2. In different populations different GA operations and parameters are applied – “heterogeneous PGA”.
3. Like in case 1, but some populations are periodically re-initialised.
4. Like in case 2, but some populations are periodically re-initialised.

## 2.1 Homogenous PGA

In each of the 9 populations the identical GA with the above mentioned structure has been used with the following specification: selection of 1 best string into Pop1, tournament selection for Pop3, probability of local and global mutation of a gene is 0.05, the step size of additive mutation is limited to 5% from the entire space, conventional one point crossover. In Fig.4 convergence graphs for all 6 types of PGA

architectures with homogenous structure are compared on the  $f_1(x)$  minimization example. Next, the convergence of a single population GA is depicted. The size of this population is equal to the number of all populations of the PGA  $9 \times 20 = 180$ , the specification is the same as mentioned above for all partial populations in the PGA. Note, that the convergence graph is the dependence of the best fitness value of all populations of the PGA, on the generation number. Each graph (also in next experiments) represents the average of 25 PGA runs. The graphs are marked A1, B1, C1, D1, E1 and F1 according the Fig.1A-F, index 1 belongs to homogenous PGA. SGA is the single population GA. Due to the use of parallel structures the convergence rate is better than for the SGA, which is characterized by a premature convergence. The first advantage of PGA's is, that in more relatively independent populations there is a better chance to find the direction to the global optimum, without any influence of some currently better local optima. Note, if the migration structure has a "high density" (too many migration connections) or/and the migration period is too short, the PGA behaviour can be similar to the SGA one and it can lead to a premature convergence.

## 2.2 Heterogeneous PGA

In the heterogeneous PGA, opposite to the homogenous case, the partial GA's in populations 1 to 9 are not identical. The aim is to introduce a diversity of search properties into the PGA. Let us consider following GA modifications in particular populations:

Population 1 and 5: the same GA as in the Section 2.1.

Population 2-4: The local (additive) mutation uses different mutation ranges (mutation step limitation) of 1%, 5%, 10% and 20% respectively from the entire search space.

Population 6: "Super elitist" algorithm, where 4 copies of the best string, 3 copies of the second best string, 2 copies of the third best string and one copy of the fourth best string are selected for the "reproduction group" (Pop3). The remaining strings are selected randomly. The rest is similar as in the Section 2.1.

Population 7: The strings for the reproduction group are selected only randomly.

Population 8: Similar to the population 1, but the mutation probability (global and local) is 0.2 (0.05 in population 1).

Population 9: Similar to the population 1, but instead of a conventional crossover the intermediate crossover is used.

In some populations, the heterogeneous PGA enables to increase the gene diversity, for instance with a high mutation probability, using random selection etc. and in other populations to increase the selection pressure with high elitism, without violating the overall PGA operation. For example, in the hierarchical PGA architectures E and F it is advantageous to keep a high diversity measure in populations in the lower hierarchical level (populations 6-9), to increase the local search performance using high local mutation probability in combination with high selective pressure in the middle level (populations 2-5) and to concentrate the best solutions and to finalize the evolution at the upper level (population 1). In Fig.5 convergence of all 6 PGA architecture types has been compared using the example of

$f_1(x)$  minimization (the index 2 belongs to heterogeneous PGA). The difference between the homogenous and heterogeneous version is more transparent, when the GA in the homogenous version is not well adjusted (parameterised) for the particular solved problem. The heterogeneous PGA structures can be more robust and adaptable to various practical problems.

### 2.3 PGA with re-initialisation

Last two modifications will extend the PGA with a periodic re-initialisation of just some populations. In our case, this re-initialisation is implemented after each migration period, which appeared in our experiments after each 100 generations and represents the exchange of the current population by a completely new, randomly generated population. The re-initialisation has been applied in the 6 architecture types in following populations:

A (Fig. 1A): in population 3,5,6,8

B (Fig. 1B): in population 3,5,6,8

C (Fig. 1C): in populations 2-9

D (Fig. 1D): in all populations, but before the re-initialisation the best individual of the PGA is saved for the next generation

E (Fig. 1E): in populations 6-9

F (Fig. 1F): in populations 6-9

In Fig. 6 and 7 there are the results of a similar experiments as in the last two cases, but with the re-initialisation. Here, the index 3 denotes the homogenous PGA type and the index 4 the heterogeneous one. Next, the architecture types C and E for both homogenous and heterogeneous PGA's, without and with re-initialisation for each test function have been compared (C1-homogenous, C2-heterogeneous, C3-homogenous with re-initialisation, C4-heterogeneous with re-initialisation and the same for the type E). Results of the  $f_1(x)$  minimization are depicted in Fig. 8, Fig. 9 is for the  $f_2(x)$  and Fig. 10 for the  $f_3(x)$  case ( $f_3$  is a relatively smooth cost function).

From the analysis it is evident, that the re-initialisation has a positive influence on the GA convergence properties. This effect is most visible in case of such functions or optimisation problems, which cost functions are non-smooth, non-ordinary, with "unexpected" global optimum position (deceptive functions, like  $f_1$ ). Sometimes the re-initialisation is able to remove differences between homogenous and heterogeneous PGA's.

The re-initialisation is able to increase the population diversity, because it is producing new perspective search directions and in connection with a sufficiently high selective pressure (in other populations of the PGA) it is an effective way for the global optimum search.

## 3 Re-initialisation types

Finally, different re-initialisation mechanisms have been proposed and tested. The simplest way was already mentioned in Section 2.3 - the periodical use, without any other conditions. Additionally two other methods have been proposed. In the first, the

algorithm after some number of generations compares the best individuals of each population. If

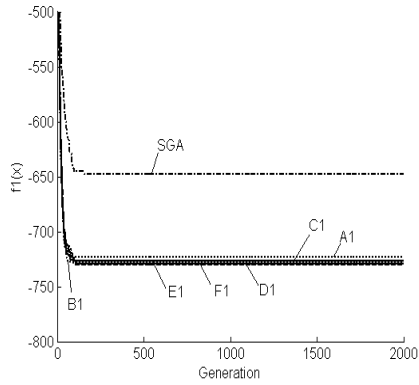


Fig.4 Homogenous PGA's,  $f_1(x)$

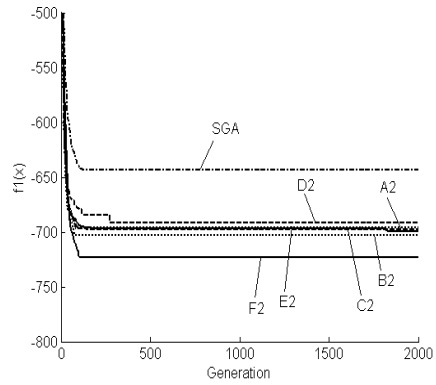


Fig.5 Heterogeneous PGA's,  $f_1(x)$

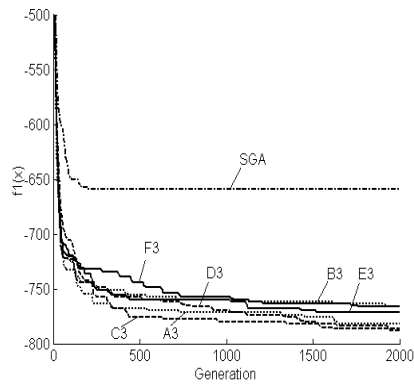


Fig.6 Homogenous PGA's with re-initialisation,  $f_1(x)$

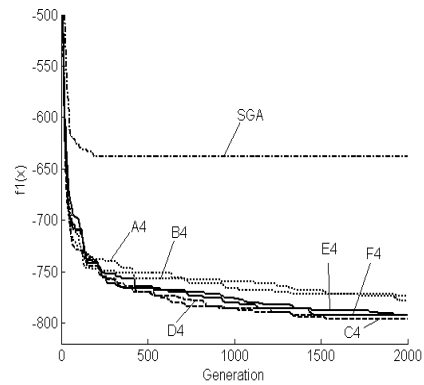


Fig.7 Heterogeneous PGA's with re-initialisation,  $f_1(x)$

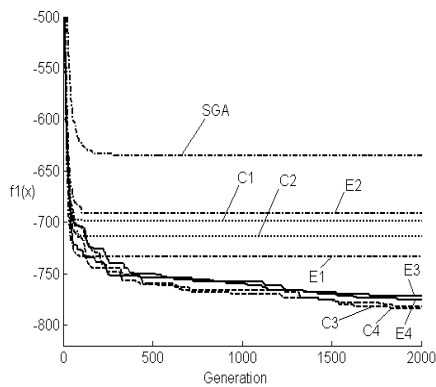


Fig.8 Comparison for  $f_1(x)$

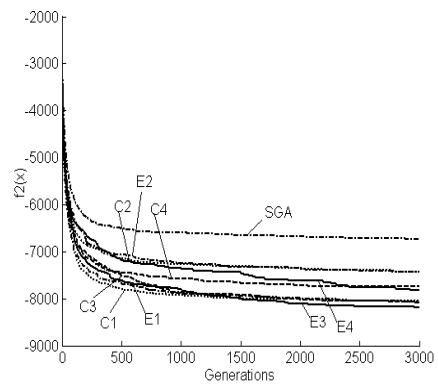


Fig.9 Comparison for  $f_2(x)$

there are some similar (or even identical) individuals (in terms of the Euclidian distance), the population which representative has the inferior fitness will be completely re-initialised. In the second method,  $n$  fittest strings of each population have been selected, where  $n$  was set to  $1/4$  of the population size. Let this subpopulation be in form of a matrix

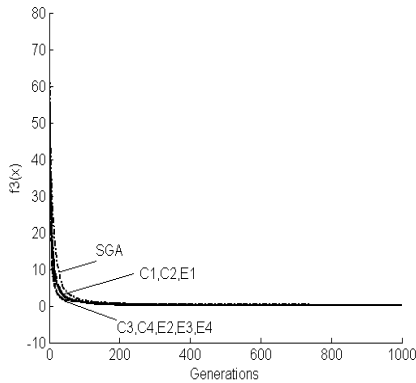
$$B = [b_{sg}]_{\substack{s=1\dots n \\ g=1\dots m}}$$

where the rows  $s$  represent strings and the columns  $g$  are their genes. Let  $\delta$  be a population diversity measure in the form

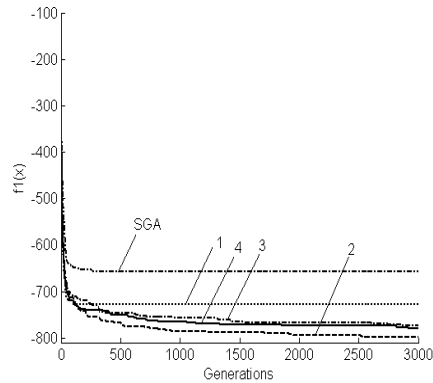
$$\delta = \sum_{g=1}^m \left( \frac{1}{n} \sum_{s=1}^n |b_{sg} - \bar{b}_g| \right) \quad (1)$$

The expression in the brackets is the mean absolute deviation,  $\bar{b}_g$  is the mean value of the  $g$ -th column. If  $\delta$  will decrease under a small defined value  $\delta < \varepsilon$ , we can assume, that the population is close to an optimum and it will no more change significantly. Such population can be re-initialised (the best individual migrates before the re-initialisation).

The comparison of the re-initialisation methods is demonstrated in Fig.11 on the example of the function  $f_1(x)$  minimization. The used PGA architecture type was C. The case without re-initialisation is marked 1, the periodic re-initialisation is marked 2, the method based on re-initialisation of similar populations is marked 3 and the method based on the diversity measure according (1) is marked 4 (remark, that each graph is the average of 25 PGA runs).



**Fig.10** Comparison for  $f_3(x)$



**Fig.11** Different re-initialisation methods,  $f_1(x)$



## 4 Conclusion

Our aim was to show some proposed parallel GA architectures and their advantages to simple GA's, which are working with a single population. First of all, the PGA's bring about the benefit of multiple independent search directions. This form of parallelism is slightly different from the parallelism in a single-population GA. In this paper different PGA architectures with different migration structures are presented. The migration linkage should be numerous (with high density of links) and/or the migration period not very large if the optimised cost function is smooth. For non-smooth (or deceptive) cost functions it is recommended to use a scarce linkage with unidirectional bindings and larger migration periods, so as to preserve a higher degree of population independency.

PGA's with heterogeneous structure, where the partial populations are using different search strategies, can effect a robustness increase i.e. an increase of independency from the solved problems or from the objective function changes, which can occur during the time. When the cost function character is not well known it is advantageous to use heterogeneous structures. In our experiments different types of PGA architecture have been used. According to our experience also from other practical applications, the best seems to be the hierarchical (two-level, three-level) architecture types C, E and F (Fig.1C, 1E, 1F).

An additional improvement in case of optimisation problems, which have "highly non-smooth" cost function can be caused by the re-initialisation of some populations of the PGA. Another observation is, that the re-initialisation is able to remove differences between homogenous and heterogeneous PGA's or between different PGA architecture types respectively. However, all the presented PGA modifications can speed up the search process and prevent the search algorithm from a premature convergence.

## References

1. Bäck T.: Selective pressure in evolutionary algorithms: A characterization of selection mechanisms. ICEC-94, 57-62, (1994)
2. Cantú-Paz, E.: A summary of research on parallel genetic algorithms. IlliGAL Report No. 95007, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, (1995)
3. Chipperfield, A.J., Fleming, P.J.: Parallel genetic algorithms: A survey. ACSE Research Report No.518, University of Sheffield, (1994)
4. Goldberg, D.E., Deb, K.: A comparative analysis of selection schemes used in genetic algorithms. In G.Rawlins, ed., *Foundation of Genetic Algorithms*, Morgan Kaufmann, (1991)
5. Man, K.F., Tang K.S, Kwong, S.: *Genetic Algorithms, Concepts and Deigns*. Springer (2001)
6. Michalewicz, Z.: *Genetic Algorithms + Data Structures = Evolution Programs*. 3rd edn. Springer-Verlag, Berlin Heidelberg New York (1996)
7. Mitchell, M.: *An introduction to Genetic Algorithms*, MIT Press, (1996)

8. Oei C.K., Goldberg D.E., Chang S.J.: Tournament selection, niching, and the preservation of diversity. Technical Report 91011, University of Illinois Genetic algorithm laboratory, (1991)
9. Schwefel, H.P.: Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie. Birkhäuser, Basel, (1977)
10. Whitley, D.: The GENITOR Algorithm and Selection Pressure.: Why Rank-based Allocation of Reproductive Trials is Best. In Proceedings of the Conf. of Genetic Algorithms, Morgan Kaufmann Publ., San Mateo, CA, (1989), 116-121

### **Acknowledgement**

This work has been supported from the grants of the Slovak grant agency VEGA 1/0155/03 and 1/0158/03.