The Effect of Selection from Old Populations in Genetic Algorithms

Mauro Castelli Luca Manzoni Leonardo Vanneschi

Dipartimento di Informatica, Sistemistica e Comunicazione
Università degli Studi di Milano - Bicocca
Viale Sarca 336, 20126 Milan, Italy
{mauro.castelli, luca.manzoni, vanneschi}@disco.unimib.it

ABSTRACT

In this paper a method to increase the optimization ability of genetic algorithms (GAs) is proposed. To promote population diversity, a fraction of the worst individuals of the current population is replaced by individuals from an older population. To experimentally validate the approach we have used a set of well-known benchmark problems of tunable difficulty for GAs, including trap functions and NK landscapes. The obtained results show that the proposed method performs better than standard GAs without elitism for all the studied test problems and better than GAs with elitism for the majority of them.

Categories and Subject Descriptors

I.2. [**Computing Methodologies**]: Artificial Intelligence—*Problem Solving, Control Methods, and Search*

General Terms

Algorithms

Keywords

Genetic Algorithms, Evolutionary Computation

1. INTRODUCTION

The goal of this paper is to define a simple-to-implement method to improve the optimization ability of Genetic Algorithms (GAs) [7, 5]. The idea is to re-use genetic material from older generations and it is similar to the concept of "short-term" memory, which is typical of Tabu Search [4], and that has already been employed in evolutionary computation so far. In GAs, the concept of memory has been used for instance in [14, 1, 10, 12, 13]. Similar concepts are common in the field of Artificial Immune Systems (AISs) [6] (for example see [2]). The main idea of the proposed "second chance" method is to insert genetic material from older populations into the current one, replacing the worst individuals in the current population. To accomplish this goal, every k generations (where $k \in \mathbb{N}$ and $k > 1$), the worst p_r% individuals in the population (where we call p_r the *replacement pressure*) are replaced. The individuals that replace them are extracted from the population of k generations before the current one (for this reason, we call k the *refresh rate*), and they are chosen from that population using exactly the same selection method used by the standard algorithm. The name "second chance" is inspired by the fact that an individual can participate in at least two selection phases, increasing its probability of being selected. Our motivations for introducing this method are the following: first of all we hypothesize that, generation by generation, the GA individuals become more and more specialized. In some cases this behavior can cause a stagnation of the algorithm into a local optimum. The insertion of earlier - and probably less specialized - individuals with a good fitness should, in our intention, allow the population diversity to increase and facilitate the algorithm to escape from local optima. A choice that can influence the behavior of the method is *how* individuals are inserted and removed from the population. There are two main methods for accomplishing this: (1) *No-Steady state*: in this case, a fraction of p_r individuals is removed from the current population and then replaced by the new individuals selected from the old population. (2) *Steady state*: in this case, at every removal of the worst individual from the current population, a new individual selected from the old population is inserted. This method is different from the previous one since an individual with a bad fitness, that has eventually been selected from the old population, could be removed at the subsequent step.

GECCO'11, July 12–16, 2011, Dublin, Ireland.
ACM 978-1-4503-0690-4/11/07.

2. EXPERIMENTAL STUDY

In all the experiments performed, the selection method used was tournament with size 4. We used one-point crossover [7, 5] with a crossover rate of 0.95 and standard GA mutation [7, 5] with a mutation rate of 0.05. The tests were performed using both standard GAs with and without elitism. The experiments were run using different individuals length and population size: *small runs* (individuals length was 32 and population size 64) and *medium runs* (individuals length was 64 and population size 128). The second chance method was tested with a *replacement pressure* p_r of 0.75, 0.5 and of 0.25 and with a *refresh rate* k of 2, 5, 10, 15, 20 and 25. Every test was composed by 1000 runs, each of which was executed for 100 generations. Both the *no-steady state* and the *steady state* methods were tested. The test functions used were the one-max problem [11], the trap functions [3] and the NK landscapes [8, 9]. For all these problems binary genomes were considered. For all the experiments the *average best fitness* (ABF) for all the generations and the set of the best fitnesses at generation 100 have been recorded. Since the results for all the experiments cannot be presented here for lack of space, we report only a qualitative discussion.

Experimental results.

On the one-max problem the proposed method always performed better than standard GAs for the values of the parameter k between 2 and 15. When p_r was 0.5 or 0.75 the proposed method also performed better than elitist GAs for $k = 2$ and better than standard GAs for all the values of k. These results indicate that high values of p_r and small values of k seem to be the best combination.

On the trap functions, high values of p_r and low values of k are confirmed to produce the best results, allowing second chance GAs to outperform both standard and elitist GAs. An interesting consideration is that the gain in fitness happens in early generations, allowing to obtaining good approximate solutions earlier compared to the other studied methods.

Even though five different values of K_{NK} (i.e. the parameter that allows to tune the amount of epistatic interaction in the NK landscapes) where tested, only three of them are discussed here. When $p_r = 0.25$, the same qualitative conclusions of the one-max and trap functions also hold for the NK landscapes. Elitist GAs outperforms the other studied methods in all these cases, with second chance GAs with $k = 2$ in the second position. A difference with respect to the other cases is that standard GAs this time is not always the method that returns the worst results. In fact, it performs better than second chance GAs with $k = 25$ when $K_{\text{NK}} = 2$ and when the steady replacement is used with small populations. The case $p_r = 0.25$ for the NK landscapes is particularly interesting since it is the first case where second chance GAs does not always perform better than standard GAs. The results for $K_{\text{NK}} = 2$, $K_{\text{NK}} = 6$ and $K_{\text{NK}} = 10$ are now discussed for the case when $p_r = 0.5$. Contrarily to what happens for the one-max and the trap functions for the same value of p_r, second chance GAs with $k = 2$, this time, does not always outperform all the other methods. More precisely, it is outperformed by elitist GAs for $K_{\text{NK}} = 6$ and $K_{\text{NK}} = 10$. It is interesting to note that standard GAs is not always the method that returns the worst results. The results for $K_{\text{NK}} = 2$, $K_{\text{NK}} = 6$ and $K_{\text{NK}} = 10$ are discussed for $p_r = 0.75$. In these cases, consistently with what happens for one-max and trap functions for the same value of p_r, second chance GAs with $k = 2$ always outperforms all the other studied methods. These results confirm the observations of the previously presented test problems: the best results have always been obtained by second chance GAs for high values of p_r and for $k = 2$.

The obtained results indicate that second chance GAs can perform better than both standard and elitist GAs, but both parameters p_r and k play an important role. The parameter p_r needs to be at least equal to 0.5 in order to allow the proposed method to perform better than elitist GAs. Higher values than 0.5 seem to work even better. In fact, for $p_r = 0.75$ second chance GAs performs better than both elitist and standard GAs for all the studied cases. Also, the parameter k plays a role that is of primary importance. Only small values seem to be of interest. In fact, for high values of k, second chance GAs has, in some cases, returned even worse results than standard GAs. In particular, only for the value of $k = 2$ second chance GAs always outperformed both standard and elitist GAs. It is interesting to note that the different behavior obtained by changing the values of the two parameters was consistent for all the studied benchmark problems. Thus, we hypothesize that the observed behavior may be problem independent.

3. CONCLUSIONS AND FUTURE WORK

A new method for improving the optimization ability of Genetic Algorithms (GAs) has been presented in this paper. It is based on the idea of re-using "good" but "old" individuals in the current population, giving them a second chance to survive and mate. The obtained results have indicated that second chance GAs is able to outperform standard GAs with no elitism for almost all the studied problems, while it is able to outperform GAs with elitism only if a particular setting of two important parameters (called refresh rate and replacement pressure) is used. In particular, "low" values of the refresh rate and "high" values of the replacement pressure seems to be the best choice for all the considered test problems. Future work includes the experimental validation of second chance GAs on a wider set of test problems of different nature and a comparison between it and a wider set of GA variants.

4. REFERENCES

[1] Y. Cao and W. Luo. Novel associative memory retrieving strategies for evolutionary algorithms in dynamic environments. In *Proceedings of the 4th International Symposium on Advances in Computation and Intelligence*, ISICA '09, pages 258–268, Berlin, Heidelberg, 2009. Springer-Verlag.

[2] L. N. De Castro and F. J. Von Zuben. The clonal selection algorithm with engineering applications. In *GECCO 2002 - Workshop Proceedings*, pages 36–37. Morgan Kaufmann, 2002.

[3] K. Deb and D. E. Goldberg. Analyzing deception in trap functions. In D. Whitley, editor, *Foundations of Genetic Algorithms, 2*, pages 93–108. Morgan Kaufmann, 1993.

[4] F. Glover and M. Laguna. *Tabu Search*. Kluwer Academic Publishers, Norwell, MA, USA, 1997.

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

[6] J. Greensmith, A. M. Whitbrook, and U. Aickelin. Artificial immune systems (2010). *Computing Research Repository (CoRR)*, abs/1006.4949. http://arxiv.org/abs/1006.4949, 2010.

[7] J. H. Holland. *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor, Michigan, 1975.

[8] S. Kauffman and S. Levin. Towards a general theory of adaptive walks on rugged landscapes. *J. Theoret. Biol.*, 128(1):11–45, 1987.

[9] S. A. Kauffman. Adaptation on rugged fitness landscapes. In D. Stein, editor, *Lectures in the Sciences of Complexity*, SFI Studies in the Sciences of Complexity, Lecture Volume I, Redwood City, 1989. Addison-Wesley.

[10] S. Louis and G. Li. Augmenting genetic algorithms with memory to solve traveling salesman problems. In *Proceedings of the Joint Conference on Information Sciences*, pages 108–111. Duke University Press, 1997.

[11] J. Schaffer and L. Eshelman. On crossover as an evolutionary viable strategy. In R. Belew and L. Booker, editors, *Proceedings of the 4th International Conference on Genetic Algorithms*, pages 61–68. Morgan Kaufmann, 1991.

[12] J. Sheppard and S. Salzberg. Combining genetic algorithms with memory based reasoning. In *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 452–459. Morgan Kauffman, 1995.

[13] M. Wiering. Memory-based memetic algorithms. In A. Nowe, T. Lenaerts, and K. Steenhout, editors, *Benelearn'04: Proceedings of the Thirteenth Belgian-Dutch Conference on Machine Learning*, pages 191–198, 2004.

[14] S. Yang. Genetic algorithms with memory-and elitism-based immigrants in dynamic environments. *Evol. Comput.*, 16:385–416, September 2008.