Maintaining Genetic Diversity in Multimodal Evolutionary Algorithms using Population Injection

Robin Mueller-Bady,
Martin Kappes
Frankfurt University of Applied Sciences
Nibelungenplatz 1
Frankfurt am Main, Germany
mueller-bady@fb2.fra-uas.de
kappes@fb2.fra-uas.de

Inmaculada Medina-Bulo,
Francisco Palomo-Lozano
Universidad de Cádiz
Avda. de la Universidad de Cádiz 10
11519 Puerto Real, Spain
inmaculada.medina@uca.es
francisco.palomo@uca.es

ABSTRACT

In this paper, we present a computationally inexpensive method for maintaining genetic diversity in evolutionary algorithms using population injection. As opposed to other methods, e.g., cellular EAs, population injection does not require any maintenance or setup effort.

Here, we present first experimental results comparing a (μ, λ) EA with and without population injection and a cellular EA using the h1 benchmark. As can be observed in the results, population injection is worth to be considered for problems which suffer from premature convergence.

CCS Concepts

•**Mathematics of computing** → **Evolutionary algorithms;**

Keywords

Evolutionary algorithm, cellular evolutionary algorithm, population injection, premature convergence, genetic diversity

1. INTRODUCTION

A common problem evolutionary algorithms (EAs) are facing is premature convergence due to a lack of genetic diversity of the population. Especially having a complex multimodal search landscape, it can often be observed that EAs tend to converge early at an arbitrary local optimum.

Early on, premature convergence has been identified as one of the major issues in complex multimodal evolutionary algorithms. Since then, several different techniques for avoiding those problems have been developed, e.g., cellular EAs [8, 1]. However, the major disadvantage is the high computational effort in maintaining the neighborhood relation of individuals and their positions within the population.

GECCO '16 July 20-24, 2016, Denver, CO, USA

ACM ISBN 978-1-4503-4323-7/16/07.

DOI: http://dx.doi.org/10.1145/2908961.2909052

Inserting new individuals into an existing generation has already been researched in the past [4] but based on different EAs and methods. There are further techniques like crowding [5], or fitness sharing [6], all of which require effort for maintenance of (meta-)information about the current evolutionary process or the current population.

In this paper, we propose an computationally efficient and inexpensive method for counteracting before mentioned problem by adding new random individuals into the population, called population injection. By injecting a controlled amount of new random individuals to an existing population, genetic diversity of the population can be increased again, leading to an improvement in the process of exploring the search landscape. Injection is triggered right after the selection phase in case premature convergence is detected based on runtime information of the optimization process, e.g. distance of individuals to each other, which, in most cases, is captured anyway. Due to selection pressure induced by selection operations, most of the newly injected individuals will be removed from the population while others enrich the population providing genetic variation.

2. EXPERIMENTS AND RESULTS

In order to measure the performance of the population injection method, we applied it to the well known h1 benchmark function as defined in [7]. The h1 benchmark function is a bidimensional continuous maximization benchmark having exactly one global optimum at $(8.6998, 6.7665)$ with value 2 in the domain $[-100, 100]$. For comparison reasons, three different EAs were used: a synchronous cellular EA, a common (μ, λ) EA, and a (μ, λ) EA using population injection. The starting population is initialized at random.

As a measure of performance, both the distance of the result to the global optimum and the improvement per evaluation are calculated. As stepsize for the EAs, the number of fitness evaluations is used. The configuration of the different EAs are shown in Table 2. As injection threshold, i.e., the difference in fitness variability between different populations, we use $\phi = 0$, whereas for comparison between different generations, the injection width, we use $\sigma = 3$. The number of individuals to be injected (injection size) is chosen according to λ. Operations and parameters are chosen according to the definitions of Deb et al. [3]. For the synchronous cellular EA, we used a toroidal von Neumann neighborhood as defined in [2] using a Manhattan distance of 2. Experimental

Table 1: h1 Benchmark Results

Benchmark	Final Optimum	Distance to Optimum	Improvement / Evaluation
Cellular	1.9942	0.0058	9.33×10^{-6}
(μ, λ)	1.9922	0.0078	9.26×10^{-6}
Injection	2.0000	0.0000	9.28×10^{-6}
Injection vs. Cellular	+0.0058 (+0.29%)	−0.0058 (+100.00%)	-0.05×10^{-6} (−0.54%)
Injection vs. (μ, λ)	+0.0078 (+0.39%)	−0.0078 (+100.00%)	$+0.02 \times 10^{-6}$ (+0.22%)

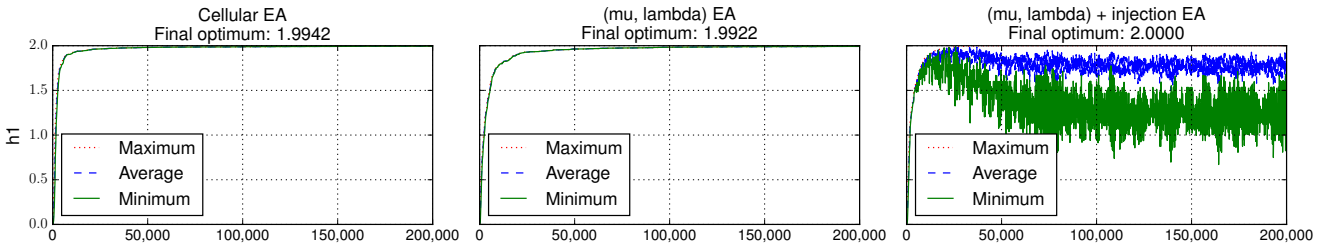

Figure 1: Results of the h1 Benchmark

Table 2: Benchmark Properties

Initial Population size	100
(μ, λ)	$(100, 100)$
Number of evaluations	200 000
Experiment repetitions	30
Selection operation	Tournament selection
Tournament size	5
Crossover operation	Sim. binary bounded
Crossover probability	0.9
Mutation operation	Polynomially bounded
Mutation probability	0.05
Mutation and crossover (η)	20

results are shown in Figure 1 and, in numerical representation, in Table 1. For the result, the minimum, average, and maximum of fitness values in relation to the number of evaluations are depicted.

As shown in Table 1, the benchmark is highly optimized by all EAs, as can be seen at the distance to the global minimum. However, the EA using population injection has slight benefits regarding the final optimum (+0.29% compared to the cellular EA and +0.39% to the (μ, λ) EA) but has slightly worse results regarding the improvement per evaluation compared to the cellular EA (-0.54%) but is 0.22% better than the (μ, λ) EA.

3. CONCLUSIONS AND FURTHER WORK

As our first results show, population injection seems to be a promising technique for management of genetic diversity while being efficient and inexpensive. Even though the cellular EA shows better performance compared to the injection-based EA regarding the final optimum, it is worth considering our proposed technique especially when computation time is expensive. In the future, we plan to investigate the influence of the methods parameters, and comparison to other techniques. Furthermore, further research using multiple and different kinds of problems, e.g. multi-objective optimization problems, needs to be done.

4. REFERENCES

[1] E. Alba and B. Dorronsoro. *Cellular Genetic Algorithms*. Springer US, 2008.

[2] R. Breukelaar and T. Bäck. Using a Genetic Algorithm to Evolve Behavior in Multi Dimensional Cellular Automata. In *Proceedings of the 2005 conference on Genetic and evolutionary computation - GECCO '05*, page 107, New York, New York, USA, jun 2005. ACM Press.

[3] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, 2001.

[4] S. M. Elsayed and R. A. Sarker. Differential Evolution with automatic population injection scheme for constrained problems. In *Differential Evolution (SDE), 2013 IEEE Symposium on*, pages 112–118, apr 2013.

[5] S. Mahfoud. Crowding and Preselection Revisited. *Urbana*, 1992.

[6] W. Spears. Simple Subpopulation Schemes. *Proceedings of the Evolutionary Programming Conference*, 3:296–307, 1994.

[7] A. J. K. van Soest and L. J. R. R. Casius. The Merits of a Parallel Genetic Algorithm in Solving Hard Optimization Problems. *Journal of Biomechanical Engineering*, 125(1):141, 2003.

[8] L. D. Whitley. Cellular Genetic Algorithms. In S. Forrest, editor, *Proceedings of the 5th International Conference on Genetic Algorithms*, page 658. Morgan Kaufmann Publishers Inc., jun 1993.

Acknowledgments

This work was supported in the framework of Hessen Modell-Projekte, financed with funds of the European Union (European Regional Development Fund - ERDF) and the State of Hessen in the context of the research project "Reactive network Optimization by Using SDN-Technology" (ROBUST) (HA project no. 473/15-15), and the Spanish Ministry of Economy and Competitiveness (National Program for Research, Development and Innovation), project DArDOS TIN2015-65845-C3-3-R and Excellence Network SEBASENet TIN2015-71841-REDT. Responsible for the content are the authors.