A New Evolutionary Approach using Pre-Post Testing to Trigger Exploration and Exploitation in DOPs

Hajer Ben-Romdhane
LARODEC Laboratory, ISG de Tunis,
Université of Tunis, Tunisia
hajer.bn.romdhan@gmail.com

Enrique Alba
Universidad de Málaga
Málaga, Spain
eat@lcc.uma.es

Saoussen Krichen
LARODEC Laboratory, ISG de Tunis,
Université de Tunis, Tunisia
saoussen.krichen@isg.rnu.tn

ABSTRACT

Striking an effective balance between exploration and exploitation (E&E) is still one of the major concerns when using evolutionary algorithms (EAs) in dynamic environments. In this work, a new scheme for adaptively balancing E&E in EAs is proposed. Based on the results of a statistical Pre-Post analysis of the population, the next search mode can be decided (i.e., exploration or exploitation). The experimental results showed that our proposal excels versus several competing approaches from the state of the art.

CCS CONCEPTS

•**Theory of computation** → *Design and analysis of algorithms;*
•**Computing methodologies** → *Artificial intelligence;*

KEYWORDS

Evolutionary algorithms, Exploration-exploitation tradeoff, Dynamic optimization, Pre-Post testing

ACM Reference format:
Hajer Ben-Romdhane, Enrique Alba, and Saoussen Krichen. 2017. A New Evolutionary Approach using Pre-Post Testing to Trigger Exploration and Exploitation in DOPs. In *Proceedings of GECCO '17 Companion, Berlin, Germany, July 15-19, 2017,* 2 pages.
DOI: http://dx.doi.org/10.1145/3067695.3076022

1 INTRODUCTION

Evolutionary algorithms (EAs) are among the most widespread and successful optimization approaches that have addressed dynamic optimization problems (DOPs). Out of all the existing problems in DOPs, convergence is one of the major concerns with EAs in dynamic environments. This is because adapting to the new environment would be difficult after convergence occurs and specific techniques are required to escape the local optima.

In this work, we introduce PROS: a new adaptive EA based on population Pre-Post testing for DOPs. While most of the existing adaptive approaches redefine the mutation rate or the population size to quickly adapt to changes, we intend to establish an adaptive scheme to explicitly balance E&E all along the optimization process. Indeed, our proposal is to manage the transitions between the two behaviors based on the results of a Pre-Post testing of the population between two time steps. Therefore, if the exploitation activity -for instance- fails in difficult regions to make an improvement, the algorithm shifts its behavior towards exploration.

GECCO '17 Companion, Berlin, Germany
 978-1-4503-4939-0/17/07…$15.00
DOI: http://dx.doi.org/10.1145/3067695.3076022

2 THE PROPOSED APPROACH

In order to better steer the search direction, our proposal is to analyze the population behavior after the accomplishment of each search activity to determine possible future improvement or stagnation in the population. This will allow the algorithm to make its choice between E&E in an adaptive manner. To be done, we perform a Wilcoxon signed-rank test to compare the fitness means of the population between two time points, so that to measure population behavior in this time span. Therefore, if the current search mode failed to improve the population so far (which is manifested by a non-significant difference between the two samples of the population), the algorithm shifts towards the other search mode. This way, the selected activity lasts as long as it is able to enrich the population, and the search mode is switched otherwise.

In PROS, the exploration activity is handled by a Steady state genetic algorithm, while the exploitation activity is assigned to a Hill-climbing local search. Thus, the two-subpopulations are evolved independently during the E&E activities, and the migration of individuals is allowed at the end of each search activity. The migration policy used here is the "best-worst". Besides, and to better tackle the dynamics of the environment, PROS uses a re-initialization scheme based on a memory of past solutions. Therefore, the current population goes first through a reevaluation, then the best solutions retrieved from the memory are inserted in the population, and finally a small rate of random individuals are inserted.

3 EXPERIMENTAL STUDY

In order to evaluate the performance of our proposal, we compare it to several EAs from the state of the art: EBEE [2], DPGA [4], MIGA [6], and a random search algorithm. The dynamic knapsack problem (DKP) is used as a dynamic test environment, for which we generate static KPs using the generator proposed in [5]. Afterwards, the dynamic instances were generated as proposed in [3]. Besides, we test under several dynamic environments with different base states k: cyclic environment with $k = 2$ (**CE(2)**) and $k = 5$ (**CE(5)**), cyclic environment with noise and $k = 5$ (**CE+N(5)**), and random environment (**RE(25)**). In our experiments, all the considered algorithms were parameterized as in [2], and the Best-of-generation Fitness (BOG) and Fitness Degradation [1] measures are used to allow meaningful comparison between the competing algorithms.

The experiments presented in this paper provide important insights into the strength of our proposal. The main aspects underlying the efficiency of PROS for DOPs can be summarized as follows. First, it outperformed all the considered algorithms in terms of

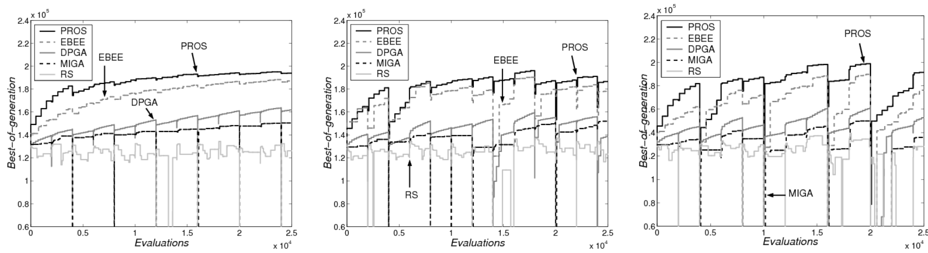

Figure 1: Comparison of algorithms average running performance on $\overline{BOG}$ for CE(2), CE+N(5), and RE(25) (from left to right)

Table 1: Results of the $\beta_{degradation}$ measure

Environment	n	PROS	EBEE	DPGA	MIGA	RS
CE(2)	100	3.72E-03	6.30E-03	9.28E-03	**1.01E-02**	-1.00E-03
	300	7.41E-03	**7.94E-03**	6.88E-03	7.11E-03	2.17E-03
	500	**7.67E-03**	5.74E-03	4.42E-03	4.99E-03	-6.18E-04
	700	**6.02E-03**	4.12E-03	2.74E-03	3.92E-03	1.02E-04
	1000	**4.41E-03**	2.10E-03	2.00E-03	3.93E-03	-4.37E-04
CE(5)	100	3.76E-03	6.26E-03	**8.65E-03**	7.94E-03	-5.97E-03
	300	**8.16E-03**	6.65E-03	6.62E-0	5.19E-033	1.01E-03
	500	**6.54E-03**	5.38E-03	3.13E-03	3.32E-03	-4.53E-03
	700	**5.62E-03**	3.89E-03	2.84E-03	3.62E-03	1.67E-03
	1000	**5.40E-03**	4.32E-04	-3.49E-03	-3.06E-04	-1.35E-04
CE+N(5)	100	1.08E-02	**4.39E-02**	5.26E-03	5.17E-03	3.63E-03
	300	**7.42E-03**	6.51E-03	4.58E-03	5.89E-03	-6.47E-03
	500	3.07E-03	**3.52E-03**	1.86E-03	2.38E-03	-1.70E-02
	700	**4.69E-03**	4.23E-03	-1.38E-03	2.49E-03	-4.37E-03
	1000	**5.69E-03**	4.98E-03	2.56E-03	-2.83E-03	-4.96E-03
RE(25)	100	1.96E-02	**2.17E-02**	9.24E-03	1.57E-02	7.31E-03
	300	**7.45E-03**	6.79E-03	4.38E-03	4.43E-03	6.63E-03
	500	**4.95E-03**	3.61E-03	2.69E-03	2.27E-03	-4.55E-03
	700	**5.04E-03**	3.26E-03	2.10E-03	1.93E-03	3.72E-03
	1000	4.71E-03	**6.87E-03**	1.72E-03	-3.24E-05	-3.07E-03
	Rank	**1**	2	3	3	5

BOG for all the considered environment configurations and problem instances. Second, according to the results obtained by the $\beta_{degradation}$ measure, PROS showed to have a desirable horizon of future high performance compared to the rest of algorithms. And finally, although EBEE is a well performing algorithm, the adaptive mechanism used in PROS showed to be more effective in steering the balance between E&E.

In an attempt to quantitatively assess the balance between E&E in PROS, we measured the duration of E&E activities by recording the number of evaluations spent between two consecutive search mode switches (NESM). In figure 2, we plot NESM over the total number of evaluations, for problem instances of size 100 and 500. It is apparent from these plots that NESM keeps on moving up and down all along the optimization process, which implies a non-uniform duration of the E&E activities. Although E&E phases are not expected to be equally long, there is some evidence that this fluctuation is essential for allowing the algorithm to maintain good performance. Indeed, mode switches arise when no improvement is experienced between two iterations and in order to prevent future performance degradation. This is well reflected on the performance of PROS when examining the plots of BOG, and fitness degradation measure: while the curves of the rest of algorithms undergo many falls and rises due to environmental changes, the curves of PROS are clearly the most stable, which indicates a balanced search engine.

REFERENCES

[1] E. Alba and B. Sarasola. 2010. Measuring Fitness Degradation in Dynamic Optimization Problems. In *Applications of Evolutionary Computation*. Lecture Notes in Computer Science, Vol. 6024. Springer Berlin / Heidelberg, 572–581.

[2] Hajer Ben-Romdhane, Saoussen Krichen, and Enrique Alba. 2017. A bi-population based scheme for an explicit exploration/exploitation trade-off in dynamic environments. *Journal of Experimental & Theoretical Artificial Intelligence* 29, 3 (2017), 453–479.

[3] Jürgen Branke, Erdem Salihoğlu, and Şima Uyar. 2005. Towards an Analysis of Dynamic Environments. In *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation (GECCO '05)*. ACM, 1433–1440.

[4] Taejin Park and Kwang Ryel Ryu. 2010. A Dual-Population Genetic Algorithm for Adaptive Diversity Control. *IEEE Transactions on Evolutionary Computation* 14, 6 (2010), 865–884.

[5] David Pisinger. 1999. Core Problems in Knapsack Algorithms. *Operations Research* 47 (1999), 570–575.

[6] Shengxiang Yang. 2008. Genetic algorithms with memory-and elitism-based immigrants in dynamic environments. *Evol. Comput.* 16 (2008), 385–416.

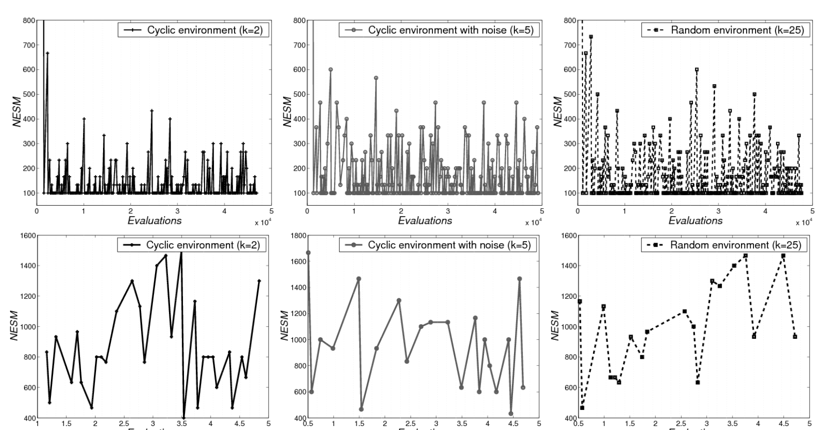

Figure 2: Average number of NESM for n=100 (first line) and n=500 (second line)