Parameter-less Population Pyramid with Feedback

Marcin M. Komarnicki
Department of Computational Intelligence
Wroclaw University of Science and Technology
Wroclaw, Poland
marcin.komarnicki@pwr.edu.pl

Michal W. Przewozniczek
Department of Computational Intelligence
Wroclaw University of Science and Technology
Wroclaw, Poland
michal.przewozniczek@pwr.edu.pl

ABSTRACT

The Parameter-less Population Pyramid (P3) is a recent method proposition that includes the linkage learning mechanisms based on the Dependency Structure Matrix (DSM) clustering. P3 was shown to be effective for solving various hard theoretical problems. In this paper, we show that for problems built from the bimodal deceptive functions the effectiveness of P3 is low due to low quality of linkage information gathered by P3. Therefore, we propose the feedback operation that periodically copies the current best individual to the lowest subpopulation of the pyramid. Such mechanism triggers the climb of the valuable individual from the lowest to the highest pyramid level (subpopulation). This process intensifies the operations performed on the best individual and may lead to breakthroughs. Moreover, it influences the DSM and thus causes the improvement of the linkage information quality.

CCS CONCEPTS

•Computing methodologies → Search methodologies;

KEYWORDS

Linkage Learning, Coevolution, Genetic Algorithms

ACM Reference format:

Marcin M. Komarnicki and Michal W. Przewozniczek. 2017. Parameter-less Population Pyramid with Feedback. In *Proceedings of GECCO '17 Companion, Berlin, Germany, July 15-19, 2017,* 2 pages.
DOI: http://dx.doi.org/10.1145/3067695.3076065

1 INTRODUCTION

Linkage learning techniques remain one of the promising directions of the Evolutionary Computation field development. Their combination with population diversity preservation may lead to more effective evolutionary methods propositions. One such propositions is the P3 method. P3 integrates linkage learning, a novel coevolution schema, and the typical evolutionary operators. The linkage learning mechanisms employed in P3 are based on the DSM clustering. Although they were shown effective and precise [2–4], in this paper, we show the well-known problems for which the performance of P3 is relatively low due to the difficulties in finding the proper linkage and in exchanging the building blocks between P3 subpopulations. The main objective of this paper is the proposition of the P3 modifcation that would improve its performance.

GECCO '17 Companion, Berlin, Germany

DOI: http://dx.doi.org/10.1145/3067695.3076065

2 PROPOSED METHOD

P3 [2] is a recent proposition of the linkage learning method designed to effectively solve hard optimization problems. Since P3 is parameter-less, it can be used without any prior tuning procedure. P3 uses a pyramid-like structure built from many subpopulations that are added during the method run. To discover linkage P3 uses the DSM-based procedure. The linkage information is used to improve the crossover operation.

In P3 new individuals are created randomly and optimized using the First Improvement Hill Climber [2]. After the optimization, the new individual is crossed with the rest of the population with the use of the linkage information. All individuals are stored in the hierarchical pyramid. The higher the subpopulation level is, the better individuals are expected. At the beginning, there is only one subpopulation and the number of levels increases during the method run. The new level is added when the crossover operation on the top level individual returns the new one that fulfills the following conditions. The new individual is not present in the pyramid yet and its fitness is better than the fitness of the parent individual.

The linkage information is separate for each level. The linkage learning technique used in P3 is adopted from Linkage Tree Genetic Algorithm (LTGA) [4]. The linkage information is stored in a form of clusters. At the particular level, all clusters must be different.

The original P3 method prevents the preconvergence by a proper isolation of its levels. However, this positive feature may sometimes be a drawback. The reasons are twofold. First, passing the good quality building blocks to the lower pyramid levels may be hard. Second, the building blocks contained by the new individual may be destroyed on its way up to the pyramid top. Thus, the valuable building block will never be delivered to the best individuals in the pyramid. Therefore, we introduce the feedback operation that allows higher pyramid levels for the limited communication with the lower ones.

The P3 method with the proposed mechanism will be denoted as Parameter-less Population Pyramid with Feedback (fP3). The difference between P3 and fP3 is as follows. After each iteration of fP3 the feedback operation is executed with a given probability, defined by a user. The feedback operation is similar to climbing the pyramid by a new individual. The difference is that instead of the new individual the best individual found so far is allowed to climb its way to the top of the pyramid. During the feedback operation, the best individual found so far is crossed with all individuals in the pyramid. If the crossover operator generates a better individual then it is added to the next level. Note, that it becomes a new global best. Sometimes it will be inserted to one of the lowest populations.

3 THE RESULTS

All methods were coded in C++, we have used the source codes pointed in [2] for P3 and LTGA, and in [3] for Dependency Structure Matrix Genetic Algorithm II (DSMGA-II). All source codes were joined on the problem definition level in one project. The complete results, source codes and the experiment configuration files are available at http://www.mp2.pl/download/ai/20170413_fp3.zip. All experiments were executed on the PowerEdge R430 Dell server, Intel Xeon E5-2670 2.3 GHz 64GB RAM with Windows 2012 Server 64-bit installed. Due to the time-based stop condition, the number of computation processes was always one less than a number of available CPU nodes. The HyperThreading was turned off. All experiments were executed in a single thread without any other resource consuming processes running. Each experiment was repeated 30 times.

The NK landscapes, deceptive step trap and bimodal deceptive function concatenations were chosen as the test problems. The choice was based on [2, 3]. The NK landscapes and the deceptive step trap were the two test problems for which the performance of the original P3 was the lowest [2]. We have used the same problem configurations as in [2]. In this paper, we also consider the concatenations of the bimodal deceptive trap functions (BDF) [1]. The definition of the BDF is presented in formula (1).

$$\text{bimodal\_trap}(t) = \begin{cases} t/2 - |k - t/2| & , t \neq k \wedge t \neq 0 \\ k & , t = k \vee t = 0 \end{cases} \quad (1)$$

The specific feature of the BDF is that it contains $\binom{k}{k/2}$ suboptima (assuming that k is even). For such problems, it may be hard to discover the linkage on the base of the pairwise gene value frequencies. Thus, the problems built from such blocks shall be difficult to solve for the methods that use the DSM-based linkage learning techniques. Here, we have used the $k = 10$ BDF blocks. On the base of BDF, the test cases using from 10 up to 100 such blocks were constructed. In addition, we have used the noised bimodal deceptive trap functions (nBDF) of the same length. The values took by nBDF used in the experiments were: 9 (unitation 0), 0 (1), 2 (2), 1 (3), 3 (4), 2 (5), 3 (6), 1 (7), 2 (8), 0 (9), 9 (10). We have used the concatenations built from 20 up to 100 BDF and nBDF blocks. The number of blocks for each type was equal.

We use the time-based stop condition. The reason is as follows. All the competing methods use different adjustments to lower the number of FFE they use. As shown in Table 1, the FFE/time ratio was significantly different for each method-problem combination. Thus, the FFE-based computation load measure does not seem appropriate.

All methods were tuned. DSMGA-II and LTGA were using the population size of 6 000 and 50 000 individuals respectively. The fP3 feedback probability was 0.03.

In Table 2 we present the comparison of the competing methods based on the ranking. The method that was the best in solving particular problem type receives the first place, the second best method receives the second place, etc. Based on such comparison it is allowed to state that fP3 outperformed other competing methods.

Table 1: The median ratio of fitness function evaluation number and computation time for each experiment type

Problem	fP3	P3	DSMGA-II	LTGA
bdf_100	13 651	14 867	**87 189**	21 233
bdf_10_nbdf_10	76 769	**90 464**	30 920	79 750
bdf_50_nbdf_50	16 133	15 977	**18 963**	18 859
dec_step_trap	8 295	8 022	16 746	**20 277**
nk_landscapes	10 417	9 260	9 904	**11 930**

Table 2: The ranking comparison of the competing methods

Problem type	fP3	P3	DSMGA-II	LTGA
bdf	2.00	3.00	**1.00**	4.00
bdf_nbdf	2.00	3.00	4.00	**1.00**
dec_step_trap	3.00	**1.50**	**1.50**	4.00
nk_landscapes	**1.50**	**1.50**	4.00	3.00
average	**2.13**	2.25	2.63	3.00

Although the results presented in [2, 3] show that LTGA performance is lower than P3 and DSMGA-II. The conclusion based on the results presented here is that the performance of each method is highly dependent on the problem nature it was applied to solve. The proposed feedback mechanism seems to be beneficial for P3 - it improves the performance of the P3 for the two considered problem types and slightly decreases it for one.

4 CONCLUSION AND FURTHER WORK

The analysis of the fP3 performance allows to state that the proposed modifications are beneficial for problems that are hard to track for P3. The effectiveness comparison with other, up-to-date methods indicates that the competing methods effectiveness depends on the problem. However, the proposed ranking-based comparison points fP3 as the most effective one. Moreover, as shown in the results section, in some situations, the FFE may not be an appropriate way of computation load measurement for the considered competing methods. This observation seems important as the use of FFE-based stop condition is frequent in the Evolutionary Computation field.

ACKNOWLEDGMENTS

This work was supported in by the Polish National Science Centre (NCN) under Grant 2015/19/D/ST6/03115.

REFERENCES

[1] Kalyanmoy Deb, Jeffrey Horn, and David E. Goldberg. 1993. Multimodal Deceptive Functions. *Complex Systems* 7, 2 (1993), 131–154.

[2] Brian W. Goldman and William F. Punch. 2014. Parameter-less Population Pyramid. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*. ACM, 785–792.

[3] Shih-Huan Hsu and Tian-Li Yu. 2015. Optimization by Pairwise Linkage Detection, Incremental Linkage Set, and Restricted / Back Mixing: DSMGA-II. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*. ACM, 519–526.

[4] Dirk Thierens and Peter A.N. Bosman. 2013. Hierarchical problem solving with the linkage tree genetic algorithm. In *Proceeding of the 2013 Annual Conference on Genetic and Evolutionary Computation Conference*. ACM, 877–884.