Parameter-less Population Pyramid with Automatic Feedback

Adam M. Zielinski
Department of Computational Intelligence
Wroclaw University of Science and Technology
Wroclaw, Poland
azielu@gmail.com

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

Parameter-less Population Pyramid (P3) is a parameter-less optimization method that employs the Dependency Structure Matrix (DSM) to discover dependencies between genes. The results published so far show that P3 effectiveness is low when bimodal trap problems are in use. Therefore, Parameter-less Population Pyramid with Feedback (fP3) was proposed. fP3 extends P3 by the feedback operation which improves P3's effectiveness when bimodal trap problems are used. However, fP3 is no longer a parameter-less method. In this paper, we propose an adaptive strategy that sets the feedback probability in each method's iteration. Therefore, newly proposed Parameter-less Population Pyramid with Automatic Feedback (afP3) is parameter-less and preserves fP3 advantages.

CCS CONCEPTS

• **Computing methodologies → Artificial intelligence**;

KEYWORDS

Linkage Learning, Local Search, Adaptation, Parameter-less

ACM Reference Format:

Adam M. Zielinski, Marcin M. Komarnicki, and Michal W. Przewozniczek. 2019. Parameter-less Population Pyramid with Automatic Feedback. In *Proceedings of Genetic and Evolutionary Computation Conference Companion (GECCO '19 Companion)*. ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3319619.3322052

1 INTRODUCTION

The tuning procedure is usually highly resource-consuming operation. Moreover, even if performed in a reliable way it may not be enough to reach high-quality results if the test cases that exist in the considered set require different method settings. On the other, methods development and the introduction of new, beneficial mechanisms frequently leads to an increase in the number of method settings. Such situation took place when fP3 was proposed. fP3 extends P3 by adding a new mechanism denoted as feedback. fP3 is not a parameter-less method anymore. Therefore, the objective of this paper is to propose a new P3 version that would self-adapt the feedback probability parameter required by fP3.

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6748-6/19/07…$15.00
https://doi.org/10.1145/3319619.3322052

2 PROPOSED METHOD

P3 [2] is the first optimization method that is both DSM-based and parameter-less. P3 is not a standard population-based optimizer. Instead of maintaining the fixed number of individuals during the whole method's run, P3 gradually adds new individuals to a structure which resembles a pyramid. This similarity is caused by storing all individuals in hierarchical subpopulations so-called levels. At the beginning of each P3's iteration, a new individual is randomly created. After that, the individual is locally optimized using First Improvement Hill Climber [2]. Thereafter, the individual climbs the pyramid beginning from the lowest subpopulation. While climbing, it is mixed with other individuals from consecutive levels.

fP3 [4] introduces the communication between highest and lowest parts of the pyramid. During the feedback operation, the best individual found so far is climbing the pyramid like the newly created one. This mechanism may lead to finding the new best solution near the old one which may resemble the elitism. Note that each individual inserted into the pyramid during the feedback operation becomes the new best individual. fP3 is no longer a parameter-less method because the feedback operation is executed with the probability given by the user at the end of each fP3's iteration.

The feedback mechanism does not only improve the effectiveness of P3 [4]. Thus, an adaptation strategy that sets the high feedback probability when the execution of the feedback mechanism may be beneficial and the low otherwise is needed. We propose the strategy assuming that the feedback probability $p_f = SR \cdot GR$ is calculated as the multiplication of two factors — Success Rate (*SR*) and Greed Factor (*GF*). Feedback operations can be split into two groups: successful feedbacks that result in finding new best individuals and the others which may be considered as failures. If more feedbacks are marked as successful, then the feedback probability should be higher. On the other hand, if the feedback operations do not influence the search process, then they should be repeated as rarely as possible. Thus, *SR* is defined as $\frac{\max\{\#s,1\}}{\max\{\#s,1\}+\#f}$, where $\#s$ denotes the number of successful feedbacks whereas $\#f$ is the number of feedbacks that are failures. The other factor *GF* should prevent premature convergence. If there are too many successful feedbacks that performed one by one, we can assume that subsequent new best individuals are similar to one another and the whole population will be dominated by such genotype. Moreover, when the method can approach the global optimum without the feedback mechanism, we should not interrupt this process. Therefore, *GF* is calculated as $\frac{\#i}{e^{\#cs}+\#i}$, where $\#i$ is the number of iterations that elapsed since the last best individual was found and $\#cs$ denotes the number of consecutive successful feedbacks.

Table 1: Main results - median computation time necessary for reaching the optimum

Problem	n	P3 Solved [%]	P3 Time median [s]	fP3 Solved [%]	fP3 Time median [s]	afP3 Solved [%]	afP3 Time median [s]	psLTGA Solved [%]	psLTGA Time median [s]	DSMGA-II Solved [%]	DSMGA-II Time median [s]
Bm. trap	800	100	11356.5	100	3827.0	100	1327.0	100	1786.0	100	410.0
Step trap	805	100	1441.0	100	1843.0	100	1999.5	100	266.0	100	776.5
HIFF	2048	100	553.5	100	1859.0	100	524.5	100	192.5	10	34063.5
Spin glass	784	100	57.5	100	56.0	100	48.5	100	53.0	100	2055
Max3Sat	60	100	0.0	100	0.0	100	0.0	100	0.0	100	8.5
Bm. noisy	800	0	25682.5	37	27577.5	43	27484.5	100	3730.5	100	5939.5
NK land.	600	100	908.5	100	1227.5	100	1373.0	100	709.5	17	23179.5
Rastr.	800	100	14.0	100	15.5	100	18.0	100	11.0	100	1289.0
Trap	805	100	26.0	100	26.0	100	25.0	100	22.0	100	222.0

3 RESULTS

The test problem set was the same as in [2] but it also contained bimodal trap problems from [4]. We compared afP3 with other up-to-date evolutionary methods — P3 [2], fP3 [4], Linkage Tree Genetic Algorithm [5] extended by the population sizing scheme [1] (psLTGA), and Dependency Structure Matrix Genetic Algorithm II [3] (DSMGA-II). fP3 and DSMGA-II were tuned. The feedback probability was set to 0.03. DSMGA-II were using the population of size 32 000. All experiments were executed on PowerEdge R430 Dell Server Intel Xeon E5-2670 2.3 GHz 64GB RAM with Windows 2012 Server 64-bit installed. We used time-based stop condition with 8-hour limit. To make comparisons reliable, the source codes of all methods were joined in one project, all were programmed in C++, and whenever it was possible methods shared the same possible pieces of code. All executions were single-threaded and executed in the clean environment, where no other resource-consuming processes were running. 30 runs were performed for each experiment. The full source codes, detailed experiment results, and the settings files is available at https://github.com/kommar/afP3.

In Table 1 we report the median computation time necessary to find the optimal solution for the largest considered test cases for each considered problem type. Based on the results presented in Table 1 afP3 performs equally well to fP3 for all the problems except bimodal trap and HIFF. For these two last problems, afP3 is significantly faster. This conclusion was confirmed by Wilcoxon statistical test. As shown in Table 2, the corresponding p-values are equal to 0, for the bimodal trap and HIFF problems. Note that afP3 is statistically slower than fP3 in finding the optimum for discretized rastrigin problem. However, the difference is lower than 10%. The comparison between afP3 and P3 shows that in afP3 is significantly faster in solving bimodal trap concatenations. For the largest considered case (800 genes) afP3 was faster by a factor of ten. This difference is statistically significant, which was confirmed by the Wilcoxon test. For the rest of the considered problems the computational resources necessary to find the optimum is similar for P3 and afP3 (no matter if the difference is statistically significant or not). Note, that P3 was unable to find the optimal solution in any run executed for bimodal noised trap function concatenation. afP3 reports 43% success rate for this problem which significantly outperforms P3. Based on the results presented in Table 1, psLTGA was faster than afP3 for four considered problems. It is interesting to compare afP3 with psLTGA and DSMGA-II. DSMGA-II is the fastest to solve bimodal trap problem from all considered methods. DSMGA-II also outperforms afP3 for step trap problem and bimodal noised problem. However, DSMGA-II is significantly outperformed by afP3 for all other considered problems.

Table 2: Wilcoxon test p-values referring to Table 1

afP3 is equal to	P3	fP3	psLTGA	DSMGA-II
Bm. trap (800)	0.000	0.000	0.000	0.000
HIFF (2048)	0.068	0.000	0.000	N/A
Rastr. (800)	0.000	0.001	0.000	0.000

4 CONCLUSION

Based on the obtained results, we can state that the proposed afP3 (in contrast to fP3) is a parameter-less method which significantly improves its usefulness. Compared to other considered up-to-date evolutionary methods afP3 seem to outperform DSMGA-II and is worse than psLTGA. psLTGA was able to solve bimodal noised trap problem in all of the runs, while afP3 only in 43%, and psLTGA usually requires slightly less time to reach the optimum.

ACKNOWLEDGMENTS

This work was supported in by the Polish National Science Centre (NCN) under Grant 2015/19/D/ST6/03115 and the statutory funds of the Department of Computational Intelligence.

REFERENCES

[1] Peter A.N. Bosman, Ngoc Hoang Luong, and Dirk Thierens. 2016. Expanding from Discrete Cartesian to Permutation Gene-pool Optimal Mixing Evolutionary Algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016 (GECCO '16)*. 637–644.

[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*. 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*. 519–526.

[4] Marcin M. Komarnicki and Michal W. Przewozniczek. 2017. Parameter-less Population Pyramid with Feedback. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO '17)*. 109–110.

[5] 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*. 877–884.