[J2C] PyPop7: A Pure-Python Library for Population-Based Black-Box Optimization

1st Yajing Tan
Southern University of Science and Technology
Shenzhen, China
12332416@mail.sustech.edu.cn

2nd Qiqi Duan
Southern University of Science and Technology
Shenzhen, China
11749325@mail.sustech.edu.cn

3rd Guochen Zhou
Southern University of Science and Technology
Shenzhen, China
12132378@mail.sustech.edu.cn

4th Yuhui Shi
Southern University of Science and Technology
Shenzhen, China
shiyh@sustech.edu.cn

***Abstract*—In this J2C Track, we will introduce our paper on one of top-tier Machine Learning journals: *Duan et al., PyPop7: A Pure-Python Library for Population-Based Black-Box Optimization, Journal of Machine Learning Research, 25(296):1-28, 2024*. Specifically, we will present a well-designed Python library called PyPop7 for black-box optimization (BBO). As population-based stochastic search methods have become popular for BBO, the goal of PyPop7 is to provide light-weighted implementations and an extensible platform for them. Since they suffer easily from the notorious curse of dimensionality from randomization, till now various extensions and improvements have been proposed to alleviate this challenge via exploiting problem and algorithm structures: such as approximation and decomposition often with low computational overheads. PyPop7 has included many of these advances from various algorithm families and has been used and/or tested in more than 10 open-source GitHub projects. Its source code and online documents can be easily accessible via https://github.com/Evolutionary-Intelligence/pypop and https://pypop.rtfd.io, respectively.**

***Index Terms*—open-source library, python**

In this J2C Track of IEEE CEC-2025, we will introduce our latest paper on one of top-tier Machine Learning journals: *Duan et al., PyPop7: A Pure-Python Library for Population-Based Black-Box Optimization, Journal of Machine Learning Research (JMLR), 25(296):1-28, 2024*. Please refer to [1] for more details. Specifically, in this JMLR paper we will present a well-designed pure-Python software library named as PyPop7 only for black-box optimization (BBO), especially in large-scale (aka high-dimensional) scenarios.

As a variety of population-based stochastic search methods have become popular for real-parameter BBO, the main goal of PyPop7 is to provide light-weighted yet efficient implementations for them, including but not limited to evolutionary algorithms and swarm intelligence [2]. We also wish this open-source Python library as an extensible platform to adding some of latest or missed algorithmic advances on high-dimensional BBO, as Python has become the most widely used programming language for artificial intelligence.

This J2C paper and open-source library was supported by Shenzhen Fundamental Research Program under Grant No. JCYJ20200109141235597.

Since they suffer easily from the notorious curse of dimensionality from randomization [3], till now various extensions and improvements have been proposed to alleviate this challenge via exploiting problem and algorithm structures: such as approximation and decomposition often with low computational overheads. PyPop7 has included many of these advances from various algorithm families, including genetic algorithms [4], evolution strategies [5] [6], estimation of distribution algorithms [7], particle swarm optimization [8], etc. To response to *"metaphor-based metaheuristics, a call for action"* [8], we have also provided necessary design philosophy to help us choose from the disorganized evolutionary computation bestiary [9].

Till now, PyPop7 has been used and/or tested in >10 open-source GitHub projects, including black-box fine-tuning of fundamental models. Currently, we are planning to extend it for more complex distributed computing [10] scenarios.

References

[1] Q. Duan et al., "Pypop7: A pure-python library for population-based black-box optimization," JMLR, vol. 25, no. 296, pp. 1–28, 2024.
[2] A. E. Eiben and J. Smith, "From evolutionary computation to the evolution of things," Nature, vol. 521, no. 7553, pp. 476–482, 2015.
[3] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization." JMLR, vol. 13, no. 2, 2012.
[4] S. Forrest, "Genetic algorithms: principles of natural selection applied to computation," Science, vol. 261, no. 5123, pp. 872–878, 1993.
[5] Y. Akimoto, A. Auger, T. Glasmachers, and D. Morinaga, "Global linear convergence of evolution strategies on more than smooth strongly convex functions," SIOPT, vol. 32, no. 2, pp. 1402–1429, 2022.
[6] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," ECJ, vol. 9, no. 2, pp. 159– 195, 2001.
[7] S. Baluja and R. Caruana, "Removing the genetics from the standard genetic algorithm," ICML. Elsevier, 1995, pp. 38–46.
[8] C. Aranha et al., "Metaphor-based metaheuristics, a call for action: the elephant in the room," Swarm Intelligence, vol. 16, no. 1, pp. 1–6, 2022.
[9] F. Campelo and C. Aranha, "Lessons from the evolutionary computation bestiary," Artificial Life, vol. 29, no. 4, pp. 421–432, 2023.
[10] Q. Duan et al., "Distributed evolution strategies with multi-level learning for large-scale black-box optimization," TPDS, vol. 35, no. 11, pp. 2087–2101, 2024.