

# Moead-framework : a modular MOEA/D python framework

Geoffrey Pruvost<sup>1</sup>, Bilel Derbel<sup>1</sup>, and Arnaud Liefooghe<sup>1</sup>

<sup>1</sup> Univ. Lille, CNRS, Inria, Centrale Lille, UMR 9189 CRISTAL, F-59000 Lille, France

DOI: [10.21105/joss.02974](https://doi.org/10.21105/joss.02974)

## Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Editor: [Melissa Weber Mendonça](#) ↗

## Reviewers:

- [@sjvrijn](#)
- [@chkoar](#)

Submitted: 07 December 2020

Published: 13 July 2021

## License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

## Summary

The Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) is a general-purpose algorithm for approximating the Pareto set of multi-objective optimization problems (Zhang & Li, 2007). It decomposes the original multi-objective problem into a number of single-objective optimization sub-problems and then uses an evolutionary process to optimize these sub-problems simultaneously and cooperatively. MOEA/D is a state-of-the-art algorithm in aggregation-based approaches for multi-objective optimization.

The goal of the *moead-framework* python package is to provide a modular framework for scientists and researchers interested in experimenting with MOEA/D and its numerous variants.

## Statement of Need

The MOEA/D algorithm is now considered as a framework. MOEA/D is the basis of many variants that improve or add new components to improve MOEA/D performance. The first version of MOEA/D and its most famous variants (Li & Zhang, 2009; Zhang et al., 2009) are implemented in recent multi-objective optimization software such as pymoo (Blank & Deb, 2020), pygmo (Biscani & Izzo, 2020) and jMetal (Nebro et al., 2015). These implementations offer many state-of-the-art algorithms, visualization tools or parallelization abstraction, but they do not enable detailed testing and analysis of the various algorithm's components' behavior. The modular R package MOEADr (Campelo et al., 2020) focuses on MOEA/D and allows the definition of different variants for each component of MOEA/D. While some modular frameworks already exist in Python for evolutionary algorithms such as DEAP (Fortin et al., 2012) or ModEA (van Rijn et al., 2016), these do not (easily) support implementing MOEA/D variants. Instead, they focus mostly on single-objective optimization and CMA-ES variants respectively.

With the *moead-framework* package, we aim to provide the modularity of the MOEADr package by using the flexibility of Python. Indeed, we want to allow the user to update the behavior of MOEA/D components in their research works without being limited by the software. The package is focused on a modular architecture for easily adding, updating or testing the components of MOEA/D and for customizing how components interact with each other. Indeed, in contrast with other existing implementations, *moead-framework* does not limit the users with a limited number of components available as parameters (8 components are available in MOEADr). Users can easily restructure the 10 existing components of the *moead-framework* and include new ones to easily add new features without altering existing components. Components are not only customizable with parameters as with MOEADr, but in fact they can be added with the inheritance mechanism on the main `run()` method of each algorithm.

40 For example, the *moead-framework* package was used for creating novel sub-problem selection  
41 strategies and analyzing them (Pruvost, Derbel, Liefooghe, Li, et al., 2020), and for rewriting  
42 the component used to generate new candidate (offspring) solutions with a variant based on  
43 Walsh surrogates (Pruvost, Derbel, Liefooghe, Verel, et al., 2020).

## 44 Documentation

45 The documentation is available at the following URL: [moead-framework.github.io/framework/](https://moead-framework.github.io/framework/).  
46 A [complete example](#) and [all components](#) are described in details. [Two tutorials](#) are made  
47 available for the user to experiment with their own multi-objective optimization problem and  
48 to implement their own MOEA/D variants.

## 49 Acknowledgements

50 This work was supported by the French national research agency (ANR-16-CE23-0013-01) and  
51 the Research Grants Council of HongKong (RGC Project No. A-CityU101/16).

## 52 References

- 53 Biscani, F., & Izzo, D. (2020). A parallel global multiobjective framework for optimization:  
54 pagmo. *Journal of Open Source Software*, 5(53), 2338. [https://doi.org/10.21105/joss.](https://doi.org/10.21105/joss.02338)  
55 [02338](https://doi.org/10.21105/joss.02338)
- 56 Blank, J., & Deb, K. (2020). Pymoo: Multi-objective optimization in python. *IEEE Access*,  
57 8, 89497–89509. <https://doi.org/10.1109/ACCESS.2020.2990567>
- 58 Campelo, F., Batista, L. S., & Aranha, C. (2020). The MOEADr package: A component-  
59 based framework for multiobjective evolutionary algorithms based on decomposition. *Journal of Statistical Software*, 92(6). <https://doi.org/10.18637/jss.v092.i06>
- 61 Fortin, F.-A., De Rainville, F.-M., Gardner, M.-A., Parizeau, M., & Gagné, C. (2012). DEAP:  
62 Evolutionary algorithms made easy. *Journal of Machine Learning Research*, 13, 2171–2175.
- 63 Li, H., & Zhang, Q. (2009). MOEA/D-DE : Multiobjective Optimization Problems With  
64 Complicated Pareto Sets, MOEA/D and NSGA-II. *IEEE Transactions on Evolutionary*  
65 *Computation*, 13(2), 284–302. <https://doi.org/10.1109/TEVC.2008.925798>
- 66 Nebro, A. J., Durillo, J. J., & Vergne, M. (2015). Redesigning the JMetal multi-objective  
67 optimization framework. *Proceedings of the Companion Publication of the 2015 Annual*  
68 *Conference on Genetic and Evolutionary Computation*, 1093–1100. [https://doi.org/10.](https://doi.org/10.1145/2739482.2768462)  
69 [1145/2739482.2768462](https://doi.org/10.1145/2739482.2768462)
- 70 Pruvost, G., Derbel, B., Liefooghe, A., Li, K., & Zhang, Q. (2020). On the combined impact  
71 of population size and sub-problem selection in MOEA/d. In L. Paquete & C. Zarges  
72 (Eds.), *Evolutionary computation in combinatorial optimization* (pp. 131–147). Springer  
73 International Publishing. [https://doi.org/10.1007/978-3-030-43680-3\\_9](https://doi.org/10.1007/978-3-030-43680-3_9)
- 74 Pruvost, G., Derbel, B., Liefooghe, A., Verel, S., & Zhang, Q. (2020). Surrogate-assisted  
75 multi-objective combinatorial optimization based on decomposition and walsh basis. *Pro-*  
76 *ceedings of the 2020 Genetic and Evolutionary Computation Conference*, 542–550. <https://doi.org/10.1145/3377930.3390149>  
77 <https://doi.org/10.1145/3377930.3390149>

- 78 van Rijn, S., Wang, H., van Leeuwen, M., & Bäck, T. (2016). Evolving the structure of  
79 evolution strategies. *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*.  
80 <https://doi.org/10.1109/SSCI.2016.7850138>
- 81 Zhang, Q., & Li, H. (2007). MOEA/d: A multiobjective evolutionary algorithm based on  
82 decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6), 712–731. <https://doi.org/10.1109/TEVC.2007.892759>  
83
- 84 Zhang, Q., Liu, W., & Li, H. (2009). The performance of a new version of MOEA/d on CEC09  
85 unconstrained MOP test instances. *2009 IEEE Congress on Evolutionary Computation*,  
86 203–208. <https://doi.org/10.1109/CEC.2009.4982949>

DRAFT