




# SBArchOpt: Surrogate-Based Architecture Optimization

Jasper H. Bussemaker <sup>1</sup>

<sup>1</sup> Institute of System Architectures in Aeronautics, German Aerospace Center (DLR), Hamburg, Germany

DOI: [10.21105/joss.05564](https://doi.org/10.21105/joss.05564)

## Software

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

Editor: [Vissarion Fisikopoulos](#) 

## Reviewers:

- [@athulpg007](#)
- [@SichengHe](#)

Submitted: 31 May 2023

Published: 07 September 2023

## License

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

## Summary

In engineered systems, the architecture of a system describes how the components of a system work together to fulfill the system functions and meet stakeholder expectations ([Crawley et al., 2015](#)). As the architecture is developed in early conceptual design stages, decisions involving architecture often have a large influence on final system performance, for example, in terms of how well the functions are fulfilled, or at what cost and in what timeframe. However, architecture design spaces (i.e., the set of all possible architecture alternatives) can be very large due to the combinatorial nature of architecture decisions, making it infeasible to compare all alternatives to each other. Additionally, for new systems there might not be any prior experience to start from, requiring the use of (typically) computationally-expensive physics-based simulation to estimate system performance.

The field of *system architecture optimization* aims to enable the use of physics-based simulation for exploring the large combinatorial architecture design space, by formulating the system architecting process as a numerical optimization problem ([Bussemaker & Ciampa, 2022](#)). In optimization, the goal is to minimize (or maximize) one or more objective functions by modifying design variables, while ensuring that design constraints are satisfied. For example, for an aircraft propulsion system, the objectives could be to minimize energy consumption and operating costs, by changing number of propellers, mechanical power generation source, fuel type, and operating strategy (i.e., the design variable), while ensuring that thermodynamic stress and reliability constraints are satisfied. By applying architecture optimization, more architecture alternatives can be considered in the early design phase with the expected result of better understanding of the design space and more informed decision-making.

Architecture optimization problems feature several behavioral characteristics that make them a particularly challenging class of optimization problem ([Bussemaker, Bartoli, et al., 2021](#)):

- Evaluation functions are non-linear *black-box* functions that are *expensive* to evaluate: it might for example take several hours to evaluate the performance of only one architecture alternative.
- There might be *multiple conflicting objectives* (i.e., design goals) to optimize for, meaning that rather than one optimal design, there might be a Pareto-set of optimal designs.
- Simulations used in performance evaluation might fail to converge, yielding Not-a-Number as evaluation result; this phenomenon is called *hidden constraints*, because they can be seen as design constraints that are “hidden” when defining the problem.
- The design space might contain both continuous and discrete variables, making the optimization problem a *mixed-discrete* problem.
- Decisions can be conditionally active based on other decisions: there is a *hierarchy* between decisions.

Such optimization problems can be readily solved by Multi-Objective Evolutionary Algorithms (MOEAs). However, these need many function evaluations to converge ([Chugh et al., 2019](#)), which is a problem for expensive evaluation functions. Surrogate-Based Optimization (SBO)

algorithms and in particular Bayesian Optimization (BO) algorithms ([Garnett, 2023](#)) instead build a surrogate model (also known as response surface or regression function) of the design space, use that model to suggest new design points to evaluate, and repeat the process after updating the surrogate model. This approach is powerful, although existing SBO algorithms need to be extended to support all architecture optimization challenges described above.

## Statement of need

Several open-source Surrogate-Based Optimization (SBO) libraries already exist, in particular BoTorch ([Balandat et al., 2020](#)), Trieste ([Picheny et al., 2023](#)), SMAC3 ([Lindauer et al., 2022](#)), and HEBO ([Cowen-Rivers et al., 2022](#)). These libraries support multi-objective and mixed-discrete optimization; however, they do not all support hidden constraints and decision hierarchy. The latter requires the automatic correction and imputation of design vectors to ensure no duplicate design vectors are generated ([Bussemaker, Bartoli, et al., 2021](#)).

The purpose of SBArchOpt is to provide a one-stop solution for solving architecture optimization problems, by:

1. providing a common interface for implementing architecture optimization problems, ensuring that all information needed by optimization algorithms is available; and
2. providing several options for optimization algorithms that work out-of-the-box for most architecture optimization problems.

*SBArchOpt* implements experience with solving architecture optimization problems (e.g., [Bussemaker, Bartoli, et al., 2021](#); [Bussemaker et al., 2023](#)) in an open-source Python library. Target users are systems engineers and researchers in the field of (architecture) optimization.

*SBArchOpt* will be used as the go-to library for solving architecture optimization problems modeled using ADORE ([Bussemaker et al., 2022](#)), software developed by the German Aerospace Center (DLR) and applied in several German and European research projects. ADORE already implements the problem definition API of *SBArchOpt*.

## Library features

The problem definition API `ArchOptProblemBase` extends the `Problem` class of `pymoo` ([Blank & Deb, 2020](#)), an evolutionary optimization framework, with several additional features:

1. a unified way to define continuous, integer, and categorical design variables;
2. an interface for accepting modified design vectors from the evaluation function;
3. a function for correcting design vectors without running an evaluation;
4. a function for returning which design variables are conditionally active;
5. an interface for storing and loading problem-specific intermediate results; and
6. several functions for compiling statistics about the design space, such as the number of valid architectures, the average number of active design variables, and more.

Optionally, the hierarchical design space structure can also be specified using the `ExplicitDesignSpace` class, which then relieves the user from implementing correction, conditional-activeness and statistics-related functions. The explicit design space definition uses `ConfigSpace` ([Lindauer et al., 2022](#)) to model conditional activation and value-pair constraints.

Then, *SBArchOpt* implements several features that may be used by any optimizer, using `pymoo`'s API:

1. a sampling algorithm for hierarchical design spaces;
2. a repair operator that calls the correction function of the problem class; and
3. intermediate results storage and restart capabilities.

To solve optimization problems, *SBArchOpt* implements the following (interfaces to) optimization libraries/algorithms:

1. pymoo: *SBArchOpt* provides a pre-configured version of the NSGA2 evolutionary optimization algorithm;
2. ArchSBO: a custom implementation of a mixed-discrete, multi-objective Surrogate-Based Optimization algorithm, with support for design variable correction, hidden constraints, and restart, using state-of-the-art mixed-discrete, hierarchical Gaussian Process models (Saves et al., 2023);
3. three open-source Bayesian Optimization libraries: BoTorch (Ax) (Balandat et al., 2020), Trieste (Picheny et al., 2023), and HEBO (Cowen-Rivers et al., 2022);
4. two proprietary Bayesian Optimization libraries: SEGOMOE (Bartoli et al., 2019) and SMARTy (Bekemeyer et al., 2022); and
5. a Tree Parzen Estimator (TPE) algorithm with support for hidden constraints.

Finally, to support development of optimization algorithms, *SBArchOpt* also provides a database of test functions:

1. many analytical test problems with various combinations of characteristics: continuous vs mixed-discrete, single- or multi-objective, with or without constraints, hierarchy, and/or hidden constraints;
2. a Guidance, Navigation and Control (GNC) optimization problem from (Apaza & Selva, 2021) trading-off system mass against reliability, and a little over 79 million possible architectures; and
3. an aircraft jet engine architecture optimization problem from (Bussemaker, De Smedt, et al., 2021) that uses a realistic engine simulation framework for performance evaluation, features hidden constraints, and trades fuel consumption against engine weight and emissions.

## Acknowledgements

I would like to thank my colleagues at the DLR in Hamburg (DE) and at ONERA in Toulouse (FR) for supporting my research and for their helpful discussions and feedback. In particular, I would like to thank Nathalie Bartoli, Thierry Lefebvre, Paul Saves, and Rémi Lafage for their warm welcome in Toulouse, and Luca Boggero and Björn Nagel for supporting my secondment at ONERA.

## References

- Apaza, G., & Selva, D. (2021, August). Automatic composition of encoding scheme and search operators in system architecture optimization. *Volume 2: 41st Computers and Information in Engineering Conference (CIE)*. <https://doi.org/10.1115/detc2021-71399>
- Balandat, M., Karrer, B., Jiang, D. R., Daulton, S., Letham, B., Wilson, A. G., & Bakshy, E. (2020). BoTorch: A framework for efficient Monte-Carlo Bayesian optimization. In H. Larochelle, M. Ranzato, R. Hadsell, M.-F. Balcan, & H.-T. Lin (Eds.), *Advances in neural information processing systems 33: Annual conference on neural information processing systems 2020, NeurIPS 2020, december 6-12, 2020, virtual*. <https://proceedings.neurips.cc/paper/2020/hash/f5b1b89d98b7286673128a5fb112cb9a-Abstract.html>
- Bartoli, N., Lefebvre, T., Dubreuil, S., Olivanti, R., Priem, R., Bons, N., Martins, J. R. R. A., & Morlier, J. (2019). Adaptive modeling strategy for constrained global optimization with application to aerodynamic wing design. *Aerospace Science and Technology*, 90, 85–102. <https://doi.org/10.1016/j.ast.2019.03.041>

- Bekemeyer, P., Bertram, A., Hines Chaves, D. A., Dias Ribeiro, M., Garbo, A., Kiener, A., Sabater, C., Stradtner, M., Wassing, S., Widhalm, M., & others. (2022). Data-driven aerodynamic modeling using the DLR SMARTy toolbox. *American Institute of Aeronautics and Astronautics (AIAA) Aviation 2022 Forum*, 3899. <https://doi.org/10.2514/6.2022-3899>
- Blank, J., & Deb, K. (2020). Pymoo: Multi-objective optimization in Python. *IEEE Access*, 8, 89497–89509. <https://doi.org/10.1109/access.2020.2990567>
- Bussemaker, J. H., Bartoli, N., Lefebvre, T., Ciampa, P. D., & Nagel, B. (2021, August). Effectiveness of surrogate-based optimization algorithms for system architecture optimization. *American Institute of Aeronautics and Astronautics (AIAA) AVIATION 2021 FORUM*. <https://doi.org/10.2514/6.2021-3095>
- Bussemaker, J. H., Boggero, L., & Ciampa, P. D. (2022, June). From system architecting to system design and optimization: A link between MBSE and MDAO. *32nd Annual International Council on Systems Engineering (INCOSE) International Symposium*. <https://doi.org/10.1002/iis2.12935>
- Bussemaker, J. H., & Ciampa, P. D. (2022). MBSE in architecture design space exploration. In A. M. Madni, N. Augustine, & M. Sievers (Eds.), *Handbook of model-based systems engineering*. Springer. [https://doi.org/10.1007/978-3-030-27486-3\\_36-1](https://doi.org/10.1007/978-3-030-27486-3_36-1)
- Bussemaker, J. H., De Smedt, T., La Rocca, G., Ciampa, P. D., & Nagel, B. (2021, August). System architecture optimization: An open source multidisciplinary aircraft jet engine architecting problem. *AIAA AVIATION 2021 FORUM*. <https://doi.org/10.2514/6.2021-3078>
- Bussemaker, J. H., García Sánchez, R., Fouda, M., Boggero, L., & Nagel, B. (2023, July). Function-based architecture optimization: An application to hybrid-electric propulsion systems. *33rd Annual International Council on Systems Engineering (INCOSE) International Symposium*. <https://doi.org/10.1002/iis2.13020>
- Chugh, T., Sindhya, K., Hakanen, J., & Miettinen, K. (2019). A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms. *Soft Computing*, 23(9), 3137–3166. <https://doi.org/10.1007/s00500-017-2965-0>
- Cowen-Rivers, A. I., Lyu, W., Tutunov, R., Wang, Z., Grosnit, A., Griffiths, R. R., Maraval, A. M., Jianye, H., Wang, J., Peters, J., & others. (2022). HEBO: Pushing the limits of sample-efficient hyper-parameter optimisation. *Journal of Artificial Intelligence Research*, 74, 1269–1349. <https://doi.org/10.1613/jair.1.13643>
- Crawley, E., Cameron, B., & Selva, D. (2015). *System architecture: Strategy and product development for complex systems* (p. 465). Pearson Education. <https://doi.org/10.1007/978-1-4020-4399-4>
- Garnett, R. (2023). *Bayesian optimization*. Cambridge University Press. <https://doi.org/10.1017/9781108348973>
- Lindauer, M., Eggensperger, K., Feurer, M., Biedenkapp, A., Deng, D., Benjamins, C., Ruhkopf, T., Sass, R., & Hutter, F. (2022). SMAC3: A versatile Bayesian optimization package for hyperparameter optimization. *Journal of Machine Learning Research*, 23(54), 1–9. <http://jmlr.org/papers/v23/21-0888.html>
- Picheny, V., Berkeley, J., Moss, H. B., Stojic, H., Granta, U., Ober, S. W., Artemev, A., Ghani, K., Goodall, A., Paleyes, A., Vakili, S., Pascual-Diaz, S., Markou, S., Qing, J., Loka, N. R. B. S., & Couckuyt, I. (2023). *Trieste: Efficiently exploring the depths of black-box functions with TensorFlow*. arXiv. <https://doi.org/10.48550/ARXIV.2302.08436>
- Saves, P., Lafage, R., Bartoli, N., Diouane, Y., Bussemaker, J. H., Lefebvre, T., Hwang, J. T., Morlier, J., & Martins, J. R. R. A. (2023). *SMT 2.0: A surrogate modeling toolbox with a focus on hierarchical and mixed variables Gaussian processes*. <https://arxiv.org/abs/2305.13998>