

Data-Efficient Interactive Multi-objective Optimization Using ParEGO

Peer-reviewed author version

Heidari, A; ROJAS GONZALEZ, Sebastian; Dhaene, Tom & Couckuyt, I (2025)
Data-Efficient Interactive Multi-objective Optimization Using ParEGO. In: Meo, R.;
Silvestri, F. (Ed.). Machine Learning and Principles and Practice of Knowledge
Discovery in Databases, Springer International Publishing AG, p. 519 -526.

DOI: 10.1007/978-3-031-74633-8_39

Handle: <http://hdl.handle.net/1942/49402>

Data-Efficient Interactive Multi-Objective Optimization Using ParEGO*

Arash Heidari¹, Sebastian Rojas Gonzalez^{1,2}, Tom Dhaene¹, and Ivo Couckuyt¹

¹ Faculty of Engineering and Architecture, Ghent University - imec, Ghent, Belgium

² Data Science Institute, Hasselt University, Belgium

Abstract. Multi-objective optimization is a widely studied problem in diverse fields, such as engineering and finance, that seeks to identify a set of non-dominated solutions that provide optimal trade-offs among competing objectives. However, the computation of the entire Pareto front can become prohibitively expensive, both in terms of computational resources and time, particularly when dealing with a large number of objectives. In practical applications, decision-makers (DMs) will select a single solution of the Pareto front that aligns with their preferences to be implemented; thus, traditional multi-objective algorithms invest a lot of budget sampling solutions that are not interesting for the DM. In this paper, we propose two novel algorithms that employ Gaussian Processes and advanced discretization methods to efficiently locate the most preferred region of the Pareto front in expensive-to-evaluate problems. Our approach involves interacting with the decision-maker to guide the optimization process towards their preferred trade-offs. Our experimental results demonstrate that our proposed algorithms are effective in finding non-dominated solutions that align with the decision-maker's preferences while maintaining computational efficiency.

Keywords: Multi-Objective Optimization · Bayesian Optimization · Interactive Optimization · Surrogate Modelling.

1 Introduction

Traditionally, optimization problems focus on a single objective, typically minimizing cost. However, often real-world decision-making requires considering multiple conflicting objectives simultaneously. This gives rise to multi-objective or multi-criteria optimization problems, where performance is assessed using a vector of outputs representing the optimal trade-offs between the conflicting objectives (referred to as the non-dominated or Pareto-optimal set).

Multi-objective optimization is a well-established field with extensive research [5,13]. Traditional *a posteriori* methods aim to approximate the entire Pareto front first, and then allow the DM to select their preferred solution.

* This work has been supported by the Flemish Government under the ‘Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen’ and the ‘Fonds Wetenschappelijk Onderzoek (FWO, grant 12AZF24N)’ programmes.

Evidently, this is data-inefficient. *A priori* methods on the other hand, utilize a weighted cost function defined before optimization to find a single solution, which requires extensive or complete knowledge of the objective space, which is unrealistic in practical scenarios.

Our focus in this study is on a potentially more efficient approach, but under-explored in the literature: to solve the problem interactively by iteratively collecting preference information from the decision maker during the optimization process. This interactive approach may lead to the convergence of the most preferred solution(s), without wasting large amounts of computational budget [19].

The literature on interactive optimization comes from different streams of research; see for instance analytical methods [2,18], evolutionary Multi-objective Optimization (EMO) algorithms [4,16], and Bayesian multi-objective optimization (BMO) techniques [3,14]. Preference-based EMO algorithms have shown efficacy in black-box optimization of complex problems; however, they suffer from data inefficiency, implying the requirement of numerous function evaluation. On the other hand, BMO offers data efficiency for expensive-to-evaluate or time-consuming problems, but incurs an additional modelling cost and error due to the surrogate model.

Recently, researchers have proposed several variants of BMO interactive methods to guide the search towards the most preferred solution, including scalarization [17], acquisition functions that capture the DM’s preferences and provide theoretical guarantees, but require significant computational resources [3,14], and using reference points as preferences [15,7]. In [10], BO is used to identify a desirable solution that is interesting but not necessarily the most preferred solution. This solution is referred to as the knee of the Pareto front, which offers a good trade-off between different objectives.

In this work we introduce two novel scalarization-based BMO approaches to efficiently locate the preferred region of the Pareto front in an interactive and data-efficient manner, relying on Gaussian Processes to build metamodels of the objectives and advanced discretization techniques to explore interesting regions of the search space. Due to space constraints we don’t discuss Bayesian Optimization, but please refer to [6] for a comprehensive overview.

2 Proposed Methods

Throughout this paper, an interaction with the DM involves presenting the extracted Pareto front and soliciting their preference for the most desirable solution so far. The first method employs a Delaunay triangulation technique [8] to partition the input space and subsequently explores the preferred region through DM feedback. The second approach involves sampling and adjusting the scalarization weights to encourage exploration of the preferred region. Both methods consist of an initialization phase and an exploration phase; the initialization phase is based on the well-known ParEGO algorithm [11], and it is the same for both approaches. In what follows we first introduce the Initialization phase, followed by the description of both methods proposed.

2.1 Initialization Phase

The initialization phase itself comprises two steps. First, P_{space} candidates are evaluated using a space-filling design such as Halton sampling with the expensive-to-evaluate function $F(\cdot)$. In the second step, the augmented Tchebycheff scalarization function is employed with P_{init} weights sampled uniformly from a d_{out} dimensional probability simplex, $\Delta^{d_{out}}$ (defined in equation 1), where d_{out} is the number of objectives. Since the weights are sampled i.i.d. uniformly, the resulting evaluations of the proposed candidates are expected to be on or close to the Pareto front with high diversity [11]. The non-dominated solutions obtained after executing the initialization phase are subsequently presented to the DM to select their desired trade-off (i.e., their preferred solution thus far). Algorithm 1 outlines the specific steps involved in the initialization phase. It should be noted that in Step 4 of the algorithm, a small value for ρ (e.g., 0.05) is used.

$$\Delta^{d_{out}} = \left\{ \mathbf{w} \in \mathbb{R}^{d_{out}} \mid \sum_{i=1}^{d_{out}} w_i = 1, w_i \geq 0 \text{ for all } i \right\} \quad (1)$$

Algorithm 1 Initialization Phase

Input d_{out} : number of objectives
Input Δ^n : n -dimensional simplex
Input $\alpha(\cdot)$: EI Acquisition function ▷ Eq. 2
1: $D = \{x_i, F(x_i) \mid x_i \sim \text{HaltonSampling}\}_{i=1}^{P_{space}}$
2: $\mathbf{W}_{init} = \{\mathbf{w}_j \mid \mathbf{w}_j \sim \Delta^{d_{out}}\}_{j=1}^{P_{init}}$
3: **for** $\mathbf{w}$ in $\mathbf{W}_{init}$ **do**
4: $U = \max_i [w_i \cdot F_i] + \rho \sum_{j=1}^k [w_j \cdot F_j]$ ▷ Tchebycheff scalarization
5: $\tilde{M} \leftarrow \text{Surrogate model for } U$ ▷ Employing GP
6: $c = \arg\max_x \alpha(x; \tilde{M}, D)$
7: $D = D \cup \{c, F(c)\}$
8: **end for**
9: Interact with the DM for feedback
10: $P \leftarrow$ The preferred solution on the extracted Pareto front
11: $\mathbf{w}_P \leftarrow$ The corresponding weight for P
12: $\mathbf{x}_P \leftarrow$ The corresponding input for P

Algorithm 1 incorporates the Expected Improvement (EI) acquisition function, which is defined as follows:

$$EI(\mathbf{x}_*) = (f_{min} - \mu(\mathbf{x}_*))\Phi\left(\frac{f_{min} - \mu(\mathbf{x}_*)}{s(\mathbf{x}_*)}\right) + s(\mathbf{x}_*)\phi\left(\frac{f_{min} - \mu(\mathbf{x}_*)}{s(\mathbf{x}_*)}\right) \quad (2)$$

Here, $\mu(\mathbf{x}_*)$ and $s(\mathbf{x}_*)$ represent the prediction and uncertainty of the surrogate model at the new point $\mathbf{x}_*$, respectively. f_{min} denotes the minimum value extracted thus far, and Φ and ϕ represent the cumulative distribution function and the probability density function, respectively.

2.2 Triangulation-based Preferred Region Exploration (TRIPE)

The initialization phase yields a set of initial candidates and weights, which are subsequently employed in the optimization process. TRIPE employs a sophisticated method to discretize the input space and explore the region near the preferred solution using Delaunay triangulation candidates [12]. A Delaunay triangulation is a set of line segments that connects nearby points in a manner that maximizes the angles while ensuring that no point lies inside the circumcircle of those points. This triangulation method subdivides the interior of the smallest convex set, which contains all points, known as the convex hull. The center of each triangle serves as a potential candidate for the next evaluation, referred to as the *interior candidates* (solid crosses in Fig. 1). However, as found in [8], limiting the exploration to only interior candidates constrains the exploration unless all corner points are included in the initial design. Therefore, TRIPE generates additional candidates outside any triangle as suggested in [8], referred to as *fringe candidates* (hollow crosses in Fig. 1). Interested reader are encouraged to refer to [8] for the calculation of the interior and fringe candidates. Note that in the work by Gramacy et al. [8], this triangulation method was employed for single-objective Bayesian optimization; here, we have adapted this approach to explore the preferred regions of the input space in a multi-objective scenario. Specifically, we select the direct neighbours of the selected preferred solution for expensive evaluation, as illustrated in Fig. 1. Algorithm 2 presents an overview of the TRIPE exploration phase.

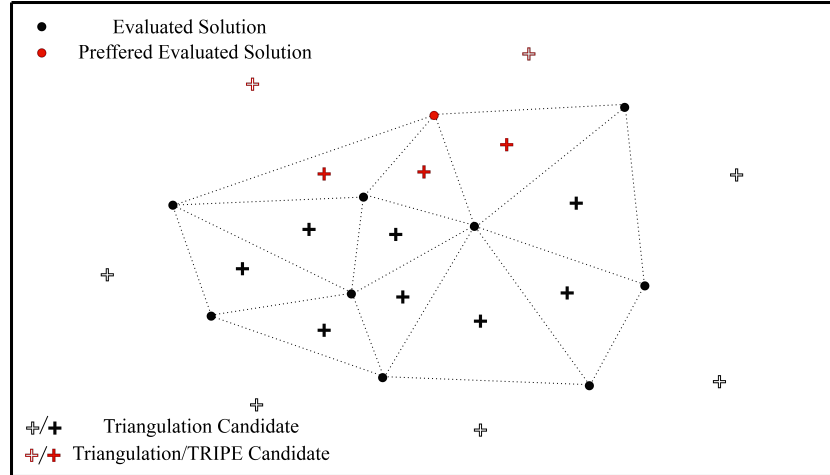


Fig. 1. Triangulation in a 2-dimensional input space. Circles represent the current solutions, with the red circle indicating the solution chosen by the DM. Crosses represent the candidates generated by triangulation. The red-colored crosses, which are neighbors of the preferred solution, will be evaluated in the next iteration.

Algorithm 2 TRIPE Exploration Phase**Input** P : Preferred solution**Input** $\mathbf{x}_P$: Corresponding input of the preferred solution

-
- 1: **while** *Budget* left **do**
 - 2: Delaunay triangulation of the Input Space
 - 3: $C_I \leftarrow$ The interior candidates neighbouring $\mathbf{x}_P$
 - 4: $C_F \leftarrow$ The fringe candidates neighbouring $\mathbf{x}_P$
 - 5: Evaluate $C_I \cup C_F$
 - 6: Interact with the DM about her preference.
 - 7: Update P and $\mathbf{X}_P$
 - 8: **end while**
-

As pointed out in [8], the Delaunay triangulation and convex hull algorithms, which rely on existing libraries, can be computationally demanding for large candidate sets (e.g., thousands), as required for high-dimensional problems. Thus, TRIPE is best suited for low-dimensional problems, typically with no more than five input dimensions.

2.3 Weight Adjustment for Preferred Region Exploration (WAPE)

The exploration phase of WAPE follows a similar procedure as the initialization phase. During the initialization phase, the weights for the scalarized function are uniformly sampled from a d_{out} -dimensional probability simplex. However, in the exploration phase of WAPE, the N newly generated weights are constrained to be in the *proximity* of the preferred weight $\mathbf{w}_P$. The generation of a new weight utilizes Equation 3, where η denotes the *explore* parameter that determines the extent of the neighbouring region around the preferred solution to be explored. Smaller values of η result in new candidates being located closer to the preferred solution, while larger values allow for a broader search in the objective space. Additionally, N controls the number of newly generated weights in the vicinity of the preferred weight. Smaller values of N encourage more interaction with the DM, while larger values aim to minimize interactions, particularly when DM interaction is resource-intensive. Algorithm 3 provides an overview of the WAPE approach.

$$\mathbf{w}_{\text{new}} = \frac{\mathbf{w}_P * \theta}{\sum_i w_{P,i} * \theta_i}, \quad (3)$$

$$\theta \sim U(1 - \eta, 1 + \eta)^{d_{out}}$$

3 Results

Standard performance measures designed for evaluating the quality of an entire Pareto front approximation obtained by an optimizer, such as hypervolume or inverted generational distance, are not suitable for assessing preference-based

Algorithm 3 WAPE Exploration Phase

Input N : number of new candidates
Input w_P : Corresponding weight for the preferred solution
Input D : Evaluated design of experiment
Input $\alpha(\cdot)$: EI Acquisition function ▷ Eq. 2

- 1: **while** *Budget* left **do**
- 2: $\mathbf{W}_{\text{new}} = \{\mathbf{w}_i | \mathbf{w}_i = \frac{\mathbf{w}_P * \theta}{\sum_i w_{P,i} * \theta_i}\}_{i=1}^N$ ▷ Generating N new weights using Eq. 3
- 3: **for** $\mathbf{w}$ in $\mathbf{W}_{\text{new}}$ **do**
- 4: $U = \max_i [w_i \cdot F_i] + \rho \sum_{j=1}^k [w_i \cdot F_i]$ ▷ Tchebycheff scalarization
- 5: $\tilde{M} \leftarrow$ Surrogate model for U ▷ Employing GP
- 6: $c = \operatorname{argmax}_x \alpha(x; \tilde{M}, D)$
- 7: $D = D \cup \{c, F(c)\}$
- 8: **end for**
- 9: Interact with the DM about her preference
- 10: Update $\mathbf{w}_P$
- 11: **end while**

multi-objective optimizers, as noted in [1]. In the absence of a DM, it is difficult to ascertain whether the algorithm is converging towards the most preferred solution. Hence, this study employs a recently proposed metric in [17], which measures the *Opportunity Cost* (OC) between a pre-defined ground truth and any other evaluated solution (see Equation 4). Here, x^* represents the ground truth in the input space, and x can be any evaluated solution up to that point. The preference of the DM is selected as the input with the lowest OC.

$$OC = U(F(x^*)) - U(F(x)) \quad (4)$$

To determine the ground truth solution x^* in Equation 4, we first apply an EMO algorithm (NSGA-II) to find the entire Pareto front of the problem. We then use the preferred weights, randomly selected by sampling from a d_{out} dimensional probability simplex, to define a scalarization function and select the solution in the Pareto front that minimizes this function as the ground truth x^* . Our proposed algorithms are evaluated using $P_{space} = 10 \times d_{in}$ and $P_{init} = 10 \times d_{out}$, where d_{in} and d_{out} represent the input and output dimensions, respectively. We also consider the cost of interaction with the decision maker, $Cost_{DM}$, which is equivalent to the cost of one expensive evaluation. Each experiment is repeated 10 times, and we report the median as well as the 20th and 80th percentiles.

We investigate the performance of the proposed methods against a competing benchmark from the literature [9] on the DTLZ2 problem with two objectives and 3, 5 and 9 input dimensions. The hyperparameters of WAPE are set to $N = 5$ and $\eta = 0.05$. The results presented in Figure 2 demonstrate that the WAPE approach exhibits superior performance to TRIPE and the benchmark from the literature. Notably, we observe that the performance trend of the objective convergence for TRIPE does not improve with increasing computational budget when the input dimension is 5 or greater.

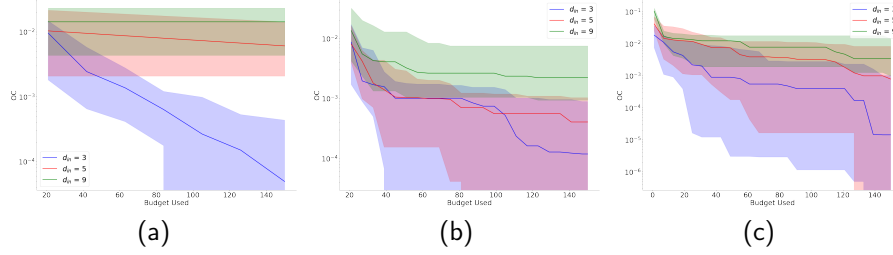


Fig. 2. The OC during optimization for (a) TRIPE and (b) WAPE (c) benchmark [9].

4 Conclusion

We have proposed two new scalarization-based Bayesian algorithms for interactive optimization of multi-objective problems with limited evaluation budgets. The TRIPE method uses a triangulation-based strategy that considers the neighbours of a given preferred solution in input space. The algorithm is hyperparameter-free and utilizes state-of-the-art computational libraries for triangulation and convex hull calculations, but is limited when scaling to more than 5 input dimensions. On the other hand, WAPE can handle more complex problems with higher input dimensions but is computationally more expensive as it uses a secondary surrogate model and acquisition function. Our preliminary results show that both proposed methods are competitive against a benchmark from the literature, and for further work we intend to evaluate more difficult test problems.

References

1. Afsar, B., Miettinen, K., Ruiz, F.: Assessing the performance of interactive multiobjective optimization methods: A survey. *ACM Computing Surveys (CSUR)* **54**(4), 1–27 (2021)
2. Armbruster, B., Delage, E.: Decision making under uncertainty when preference information is incomplete. *Management science* **61**(1), 111–128 (2015)
3. Astudillo, R., Frazier, P.: Multi-attribute bayesian optimization with interactive preference learning. In: Chiappa, S., Calandra, R. (eds.) *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research*, vol. 108, pp. 4496–4507. PMLR (26–28 Aug 2020), <https://proceedings.mlr.press/v108/astudillo20a.html>
4. Branke, J., Corrente, S., Greco, S., Słowiński, R., Zielniewicz, P.: Using choquet integral as preference model in interactive evolutionary multiobjective optimization. *European Journal of Operational Research* **250**(3), 884–901 (2016)
5. Emmerich, M., Deutz, A.H.: A tutorial on multiobjective optimization: fundamentals and evolutionary methods. *Natural computing* **17**(3), 585–609 (2018)
6. Frazier, P.I.: Bayesian optimization. In: *Recent advances in optimization and modeling of contemporary problems*, pp. 255–278. *Informatics* (2018)

7. Gaudrie, D., Le Riche, R., Picheny, V., Enaux, B., Herbert, V.: Targeting solutions in bayesian multi-objective optimization: sequential and batch versions. *Annals of Mathematics and Artificial Intelligence* **88**(1), 187–212 (Mar 2020). <https://doi.org/10.1007/s10472-019-09644-8>, <https://doi.org/10.1007/s10472-019-09644-8>
8. Gramacy, R.B., Sauer, A., Wycoff, N.: Triangulation candidates for bayesian optimization. In: Oh, A.H., Agarwal, A., Belgrave, D., Cho, K. (eds.) *Advances in Neural Information Processing Systems* (2022), <https://openreview.net/forum?id=OFJSAMwskM>
9. Hakanen, J., Knowles, J.D.: On using decision maker preferences with parego. In: Trautmann, H., Rudolph, G., Klamroth, K., Schütze, O., Wiecek, M., Jin, Y., Grimme, C. (eds.) *Evolutionary Multi-Criterion Optimization*. pp. 282–297. Springer International Publishing, Cham (2017)
10. Heidari, A., Qing, J., Rojas Gonzalez, S., Branke, J., Dhaene, T., Couckuyt, I.: Finding knees in bayesian multi-objective optimization. In: Rudolph, G., Kononova, A.V., Aguirre, H., Kerschke, P., Ochoa, G., Tušar, T. (eds.) *Parallel Problem Solving from Nature – PPSN XVII*. pp. 104–117. Springer International Publishing, Cham (2022)
11. Knowles, J.: Parego: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation* **10**(1), 50–66 (2006). <https://doi.org/10.1109/TEVC.2005.851274>
12. Lee, D.T., Schachter, B.J.: Two algorithms for constructing a delaunay triangulation. *International Journal of Computer & Information Sciences* **9**(3), 219–242 (Jun 1980). <https://doi.org/10.1007/BF00977785>, <https://doi.org/10.1007/BF00977785>
13. Li, B., Li, J., Tang, K., Yao, X.: Many-objective evolutionary algorithms: A survey. *ACM Computing Surveys (CSUR)* **48**(1), 1–35 (2015)
14. Lin, Z.J., Astudillo, R., Frazier, P., Bakshy, E.: Preference exploration for efficient bayesian optimization with multiple outcomes. In: *International Conference on Artificial Intelligence and Statistics*. pp. 4235–4258. PMLR (2022)
15. Taylor, K., Ha, H., Li, M., Chan, J., Li, X.: Bayesian preference learning for interactive multi-objective optimisation. In: *Proceedings of the Genetic and Evolutionary Computation Conference*. p. 466–475. GECCO ’21, Association for Computing Machinery, New York, NY, USA (2021). <https://doi.org/10.1145/3449639.3459299>, <https://doi.org/10.1145/3449639.3459299>
16. Tomczyk, M.K., Kadziński, M.: Decomposition-based interactive evolutionary algorithm for multiple objective optimization. *IEEE Transactions on Evolutionary Computation* **24**(2), 320–334 (2019)
17. Ungredda, J., Branke, J., Marchi, M., Montrone, T.: Single interaction multi-objective bayesian optimization. In: Rudolph, G., Kononova, A.V., Aguirre, H., Kerschke, P., Ochoa, G., Tušar, T. (eds.) *Parallel Problem Solving from Nature – PPSN XVII*. pp. 132–145. Springer International Publishing, Cham (2022)
18. Wallenius, J., Dyer, J.S., Fishburn, P.C., Steuer, R.E., Zionts, S., Deb, K.: Multiple criteria decision making, multiattribute utility theory: Recent accomplishments and what lies ahead. *Management science* **54**(7), 1336–1349 (2008)
19. Xin, B., Chen, L., Chen, J., Ishibuchi, H., Hirota, K., Liu, B.: Interactive multiobjective optimization: A review of the state-of-the-art. *IEEE Access* **6**, 41256–41279 (2018). <https://doi.org/10.1109/ACCESS.2018.2856832>