Finding Knees in Bayesian Multi-objective Optimization

Arash Heidari[1(✉)], Jixiang Qing[1], Sebastian Rojas Gonzalez[1,3], Jürgen Branke[2], Tom Dhaene[1], and Ivo Couckuyt[1]

[1] Faculty of Engineering and Architecture, Ghent University - imec, Ghent, Belgium
arash.heidari@ugent.be
[2] Warwick Business School, University of Warwick, Coventry, UK
[3] Data Science Institute, Hasselt University, Hasselt, Belgium

Abstract. Multi-objective optimization requires many evaluations to identify a sufficiently dense approximation of the Pareto front. Especially for a higher number of objectives, extracting the Pareto front might not be easy nor cheap. On the other hand, the *Decision-Maker* is not always interested in the entire Pareto front, and might prefer a solution where there is a desirable trade-off between different objectives. An example of an attractive solution is the knee point of the Pareto front, although the current literature differs on the definition of a knee. In this work, we propose to detect knee solutions in a data-efficient manner (i.e., with a limited number of time-consuming evaluations), according to two definitions of knees. In particular, we propose several novel acquisition functions in the Bayesian Optimization framework for detecting these knees, which allows for scaling to many objectives. The suggested acquisition functions are evaluated on various benchmarks with promising results.

Keywords: Multi-objective optimization · Knee finding · Bayesian optimization · Surrogate modeling

1 Introduction

Optimization is an important topic in many domains, from engineering design to economics and even biology. Real-world problems often involve multiple conflicting objectives. For example, in engineering, minimization of cost and maximization of efficiency are looked for simultaneously. As a result, there will be a set of solutions, each better in one or more objectives and worse in at least one objective. In the other words, they do not dominate each other. Hence, these solutions are referred to as *non-dominated* or *Pareto-optimal* and form the so-called Pareto set and Pareto front in the decision and objective spaces,

This work has been supported by the Flemish Government under the 'Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen' and the 'Fonds Wetenschappelijk Onderzoek (FWO)' programmes.

G. Rudolph et al. (Eds.): PPSN 2022, LNCS 13398, pp. 104–117, 2022.
https://doi.org/10.1007/978-3-031-14714-2_8

respectively. A multi-objective optimization, without the loss of generality, can be defined as:

$$minimize\ f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_m(\mathbf{x}) \qquad \mathbf{x} \in \Omega \subseteq \mathbb{R}^n \tag{1}$$

Finding a set of non-dominated solutions is challenging, or even infeasible, especially with an increasing number of objectives, as the number of solutions to cover the entire Pareto front usually grows exponentially with the number of objectives [5]. In practice, the *Decision-Maker (DM)* is not interested in the whole front of the solutions and might *prefer* a solution where there is a desirable trade-off between different objectives. One approach to tackle this problem is to transform the multi-objective setting into a single-objective problem [9,14], for example, by using a (non)linear utility function, but identifying the appropriate weights with no prior information is not an easy task.

One set of attractive solutions are the *knees* of the Pareto front (see Fig. 1), first defined in [11]. However, definitions of what *a knee* is differ in the literature; depending on the definition, a knee might hold different properties. For example, the ratio of gain and loss in each objective might be the same at a knee point.

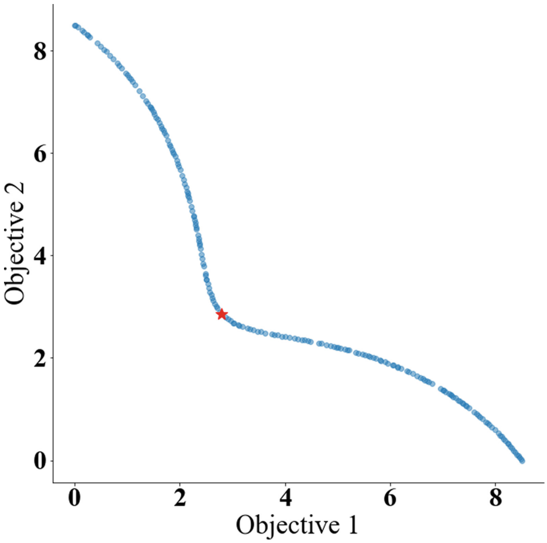


Fig. 1. Pareto front approximation of a bi-objective minimization problem. Intuitively, the knee (red star) is an attractive solution as it strikes a good trade-off between objectives. (Color figure online)

Most of the current literature on knee-oriented optimization focuses on Evolutionary Algorithms (EAs) to estimate the location of the knee. While EAs are a good solution for high-dimensional and intractable problems, they are data-hungry methods, as they evaluate the objective functions many times during optimization. However, EAs are still preferred in some situations, e.g., when the objective exhibits complex non-linear behavior.

Evaluating objective functions are often computationally expensive, severely limiting the number of function evaluations during optimization. In engineering design, for instance, high-fidelity models can take hours to days for one simulation. Thus, it is of interest to solve the problem in a data-efficient manner, i.e., finding the most interesting Pareto-optimal solutions with minimal computational budget.

In this paper, we investigate two definitions of a knee in multi-objective optimization, and propose **three novel acquisition functions** for the Bayesian Optimization (BO) framework to detect them in a **data-efficient** way. BO is an optimization technique that utilizes a surrogate model to reduce the number of time-consuming evaluations.

This paper is structured as follows. In Sect. 2, we briefly review the related work. Proposed algorithms are covered in detail in Sect. 3. Section 4 summarizes the experimental setup, while the results are discussed in Sect. 5. Finally, in the last section, we conclude with a discussion on further improvements.

2 Related Work

2.1 Bayesian Optimization

A powerful option for finding the optimum of a black-box and expensive-to-evaluate function is Bayesian Optimization (BO) [22]. BO employs an acquisition function based on a surrogate model to quantify how interesting a solution is. The point that maximizes the acquisition function will be chosen as the next candidate for evaluation (Algorithm 1). Popular choices for the acquisition function are Expected Improvement (EI) [13,18] and Probability of Improvement (PoI) [12,17]. BO can also be used to find the complete Pareto front of the solutions, using e.g., the Expected Hyper-Volume Improvement (EHVI) acquisition function [6,8].

Algorithm 1. Bayesian Optimization

```
Input Evaluated design of experiment using, e.g., Halton sampling
Input An acquisition function
while Budget left do
    Train a surrogate model
    Prediction of the surrogate model in the decision space
    $K \leftarrow$ The point that maximizes the acquisition function
    Evaluate $K$ using time consuming function
    Reduce Budget
end while
```

2.2 Gaussian Process

The Gaussian Process (GP) surrogate model [20] is a common choice in Bayesian Optimization. The GP provides the prediction for a new input as well as uncertainty. The GP is fully specified by a mean function and kernel matrix $k(\mathbf{x}_i, \mathbf{x}_j)$. Assuming a zero mean function, the posterior of the GP is given as:

$$\mu(\mathbf{x}_*) = k_* K_{xx}^{-1} y \tag{2}$$

$$\sigma^2(\mathbf{x}_*) = k_{**} - k_* K_{xx}^{-1} k_*^T \tag{3}$$

where $\mathbf{x}_*$ is a new input, $K_{xx} = k(\mathbf{x}_i, \mathbf{x}_j)$, $k_* = k(\mathbf{x}_*, \mathbf{x}_i)$, and $k_{**} = k(\mathbf{x}_*, \mathbf{x}_*)$. Different kernel functions, such as Matérn kernels [16] or an RBF kernel, can be used. Which one to choose is problem-dependent. For example, the Matérn 5/2 kernel has less strong assumptions on the smoothness of the target function and found to be more suitable for real-life problems [19].

2.3 Knee Finding Using Evolutionary Algorithms

Multi-objective Evolutionary Algorithms (MOEAs) are popular to find the Pareto front of a problem. Yu et al. [25] classify the MOEAs into four different categories, i.e., dominance relations based, decomposition based, indicator based, and secondary-criterion based methods. Interested readers can refer to [10,15,23,25] for more details.

There is no unique definition of what a knee point is. A knee point is an attractive solution of the Pareto front that will often be chosen by the DM in the absence of prior knowledge of the problem [4]. In [21], the methods to quantitatively measure a knee are classified into two different groups: (1) based on the geometric characteristics of the Pareto front, and (2) based on the trade-off information. In [25], knee-oriented MOEAs are classified into five categories, i.e., utility-based, angle-based, dominance-based, niching-based, and visualization-based approaches. Each of the algorithms has its own definition of the knee, making it difficult to compare them. For example, Branke et al. [4] defines the knee as a point in the Pareto front that has the largest angle to its neighbours, while other works take a utility-based approach for specifying a knee point [2,26]. In this work we focus on the definition of knee as described in [11] to develop the proposed acquisition functions. We also propose another definition of the knee and construct an acquisition function based on that. These are described in detail in the next section.

3 Proposed Algorithms

We investigate two definitions of a knee point: (1) based on the Hyper-Volume (i.e., the volume of objective space dominated by a given set of solutions [27]) with respect to a reference point, and (2) based on the distance to a reference line.

3.1 Hyper-Volume-Based Knee (HV-Knee)

The Hyper-Volume can be used to define knees in the Pareto front. A trade-off between various objectives can be observed by calculating the Hyper-Volume between different solutions and a reference point. Solutions with a high Hyper-Volume are intuitively more interesting for the DM. Accordingly, the knee point is the point on the Pareto front that has the maximum Hyper-Volume with respect to a fixed reference point. The corresponding regret function is calculated as follows:

$$Regret_{HV} = HV(\mathbf{y}^*{}_{HV}, N^*) - HV(\mathbf{y}_{best}, N^*) \tag{4}$$

where N^* is the true Nadir point, $\mathbf{y}^*{}_{HV}$ is the point in the Pareto front that has the maximum hyper-volume with respect to N^* (ground truth), and $\mathbf{y}_{best}$ is the point that the algorithm found and has the maximum hyper-volume with respect to N^*.

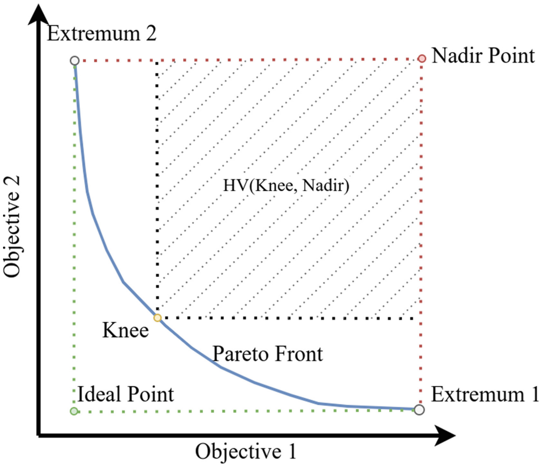


Fig. 2. Illustration of a knee point. Based on the HV-Knee definition, the point on the Pareto front that has the maximum Hyper-Volume with respect to the reference point is the best knee (yellow point). The striped region represents the Hyper-Volume (HV) between the Knee and the Nadir Point. (Color figure online)

The identified knee depends on the reference point. Due to this sensitivity, selecting the reference point is a critical part of the proposed algorithm. It can be defined upfront by the DM (informed), which might be unrealistic for many problems. Hence, we set the reference point the same as the nadir point as a sensible default (see Fig. 2) which in turn depends on an accurate estimation of the extrema. However, locating the extrema is not an easy task. We propose an interleaved approach to find the extrema. Algorithm 2 shows how two steps iterate.

Algorithm 2. The HV-Knee Algorithm

```
Input Evaluated design of experiment using, e.g., Halton sampling
while $Budget$ left do
    Train a surrogate model for each objective
    $E \leftarrow ExtremaSampler()$
    Evaluate $E$.
    Update surrogate models
    $K \leftarrow HVKneeSampler()$
    Evaluate $K$.
    Reduce $Budget$
end while
```

Both *ExtremaSampler* and *HVKneeSampler* are acquisition functions. To find the extrema, first, the ideal point, which is the minimum of each objective (see Fig. 2) is extracted from the current dataset. A large reference point will be chosen to focus more on the extrema. Finally, a derivation of the standard EHVI [6] is used. EHVI tries to evaluate the point that contributes the most to the expected Hyper-volume of the Pareto front given a fixed reference point and the extracted Pareto front so far. We modify EHVI with the ideal point as the only point in the Pareto front and a sufficiently large vector as the reference point. Reference point should have large values in a way that is dominated by all of the extremum points. For example, a vector such as $(1e6, \ldots, 1e6)$ can be used as the reference point. Algorithm 3 shows the implementation of *ExtremaSampler*.

Algorithm 3. *ExtremaSampler*: Optimizing the Extrema acquisition function

```
Input $R$ : A sufficiently large vector
$I \leftarrow$ Ideal Point extracted from the current Pareto front
$E \leftarrow$ Maximize $EHVI$ with $R$ and $I$ as the reference point and the Pareto front,
    respectively.
Return $E$ as the next candidate point
```

HVKneeSampler modifies the standard EHVI to estimate the location of the knee as well. EHVI is evaluated with the nadir point as both the reference point and the only point in the Pareto front. Algorithm 4 shows the implementation of *HVKneeSampler*.

Algorithm 4. *HVKneeSampler*: Optimizing the HVKnee acquisition function

```
$N \leftarrow$ Nadir Point extracted from the current Pareto front
$K \leftarrow$ Maximize $EHVI$ with $N$ as the reference point and the Pareto front.
Return $K$ as the next candidate point
```

3.2 Distance to Line-Based Knee

Another intuitive definition of a knee is based on the distance to an imaginary line connecting the extrema of the front in a bi-objective setting (see Fig. 3) first proposed by [11]. It is in the interest of DM to maximize the gap between a solution and the reference line. The regret function is calculated as follows:

$$Regret_{DL} = Distance(\mathbf{y}^*{}_{DL}, L^*) - Distance(\mathbf{y}_{best}, L^*) \tag{5}$$

where L^* is the true reference line, $\mathbf{y}^*{}_{DL}$ is the point in the Pareto front that has the maximum distance to the L^* (ground truth), and $\mathbf{y}_{best}$ is the current best solution.

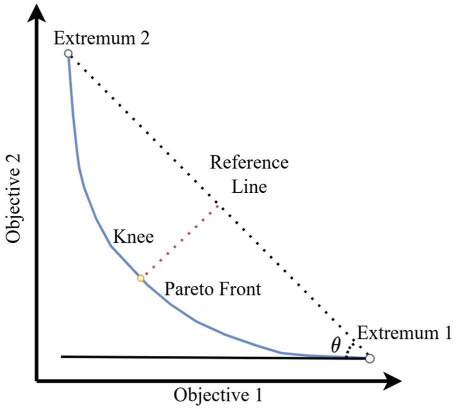


Fig. 3. Illustration of a knee based on the distance to the reference line between the two extrema. The point on the Pareto front that has the maximum distance to the line constructed by connecting the two extrema is considered the knee (yellow point). (Color figure online)

Similarly to the HV-Knee approach, the location of the extrema is unknown beforehand, and estimating them is a vital part. The two-step approach from the previous section is reused, replacing $HVKneeSampler$ with $D2LKneeSampler$.

Algorithm 5 shows the implementation of $D2LKneeSampler$. First, the current Pareto front and extrema are extracted, and the reference line will be constructed. The point in the current Pareto front that has the largest distance to the reference line is designated as the current best knee. To calculate the probability of improving over the current best knee, a naive approach is to solve a double integration requiring Monte Carlo integration. Instead, we propose to transform to a new coordinate system based on the reference line. In particular, we consider a line parallel to the reference line that passes through the current knee. The system is rotated so reference line is aligned with the horizontal axis. As a result, it is much easier to analytically integrate the (transformed) multivariate Gaussian distribution of the GPs. Now, the equation can be simplified to a single variable probability of improvement (or expected improvement) acquisition function (Fig. 4). Keep in mind that, the point should be below the reference line (assuming minimization). If the point is above the line, we use the negative of the distance.

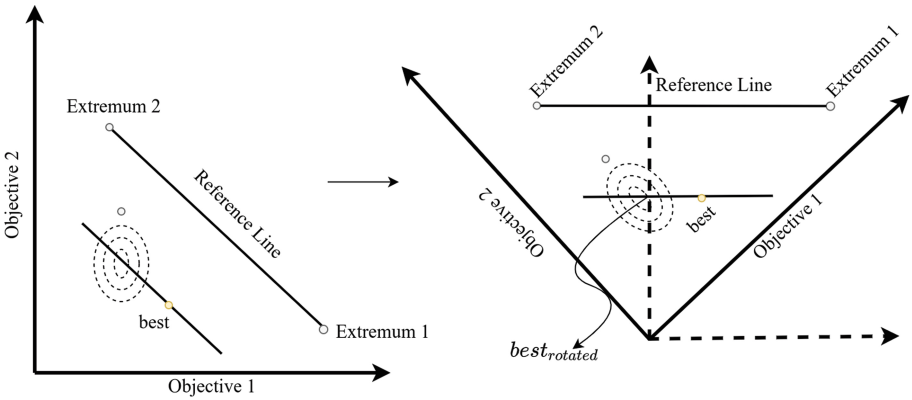


Fig. 4. The probability of improvement (or expected improvement) is calculated by transforming the problem to a new coordinate system, in such a way that the reference line becomes horizontal. A one-dimensional integration similar to the standard probability of improvement and expected improvement can be applied to the rotated Gaussian distribution.

If μ_1, σ_1^2, μ_2, and σ_2^2 are the predicted mean and variance of a candidate point using the GPs for objective 1 and objective 2, respectively, then Eqs. 6–11 can be used to calculate lines 4–9 of Algorithm 5.

$$Cov = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix} \tag{6}$$

$$Means = \begin{bmatrix} \mu_1 & \mu_2 \end{bmatrix} \tag{7}$$

$$mean_{rotated} = \mu_2 \times cos(\theta) + \mu_1 \times sin(\theta) \tag{8}$$

$$Cov_{rotated} = \begin{bmatrix} cov_{11} & cov_{12} \\ cov_{21} & cov_{22} \end{bmatrix} = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix} \times Cov \times \begin{bmatrix} cos(\theta) & sin(\theta) \\ -sin(\theta) & cos(\theta) \end{bmatrix} \tag{9}$$

$$\sigma_{rotated}^2 = cov_{22} \tag{10}$$

If the coordinates of the current knee before the rotation is $best_1$ and $best_2$, then:

$$best_{rotated} = (best_2 - tan(\theta) \times best_1) \times cos(\theta) \tag{11}$$

Algorithm 5. *D2LKneeSampler*: Optimizing the D2LKnee acquisition function

```
1: $alpha$ ← Acquisition function (probability of improvement or expected improvement).
2: $E_1, E_2$ ← Extract extrema from the current Pareto front.
3: $L$ ← Reference line constructed using $E_1$ and $E_2$.
4: $\theta$ ← The angle between the constructed reference line and the horizontal axis.
5: $Cov$ ← The diagonal covariance matrix composed of the predicted variances of the GPs.
6: $Means$ ← The predicted mean vector of the GPs.
7: $\mu_{rotated}$ ← Rotated mean vector by $\theta$ degrees
8: $Cov_{rotated}$ ← Rotated $Cov$ by $\theta$ degrees.
9: $\sigma^2_{rotated}$ ← $cov_{22}$ element of $Cov_{rotated}$
10: $best_{rotated}$ ← Vertical coordinate of the rotated best knee.
11: $K$ ← Maximize $alpha$ with $best_{rotated}$, $\mu_{rotated}$ and $\sigma^2_{rotated}$.
12: Return $K$ as the next candidate point.
```

It is possible to use either Probability of Improvement (PoI) or Expected Improvement (EI) in the 11th line of Algorithm 5, leading to the Probability of Improving with respect to the Distance to Line (PID2L), and the Expected Improvement with respect to the Distance to Line (EID2L).

4 Experimental Setup

Experiments have been conducted with three various benchmark functions, namely DO2DK, DEB2DK, and DEB3DK [4]. We configure the functions to have input dimensions 9, 5, and 7, respectively. DO2DK has an additional parameter, s, that skews the front, which is set to 1 in the experiments.

The Pareto fronts of various benchmark functions have been approximated using the NSGA-II algorithm, and the knee(s) are calculated based on the Hyper-Volume and Distance to Line knee definitions as shown in Table 1. The extracted knee(s) using NSGA-II are designated as ground truth knee(s) and used for regret calculation. The nadir point N^*, and the reference line, L^*, are constructed using extracted extrema from NSGA-II results as well. For each benchmark function, these two definitions might end up choosing the same point as the knee, but generally this is not true, however, they are often remarkably close to each other.

For extracting the Pareto front, we configure NSGA-II with population size 200 and 200 generations (DO2DK, DEB2DK), and population size 500 and 1000 generations (DEB3DK).

We compare the proposed acquisition functions for knee detection against the standard Expected Hyper-Volume Improvement (EHVI), which extracts the whole Pareto front. We use the RBF kernel for the GPs, and the number of initialization points is ten times the input dimension. To optimize the acquisition function, a Monte Carlo approach with one thousand times input dimension samples is used and L-BFGS-B optimizer is utilized to fine-tune the best point.

Table 1. Summary of the benchmark functions to validate the proposed approaches and their truth ground knee extracted using the NSGA-II algorithm.

Name	Input dimension	Output dimension	Ground truth
DEB2DK [4]	9	2	(2.83, 2.83)*
DO2DK [4]	5	2	(1.07, 1.02)*
DEB3DK [4]	7	3	(2.85, 2.83, 3.52)*

*Rounded to two decimal places

Due to ambiguous definitions of a knee, as well as the data-hungry nature of EAs, no other knee-oriented methods could be included in the comparison (evaluating the initial population would exceed the computation budget). $Regret_{HV}$ and $Regret_{DL}$ are used to measure the performance during the optimization process. Each experiment was repeated 15 times for DEB2DK and DO2DK, and 10 times for DEB3DK, and the 50^{th}, 20^{th}, and 80^{th} percentiles were calculated.

The Pymoo python package [3], and Trieste framework [1] have been used for NSGA-II and the BO methods, respectively.

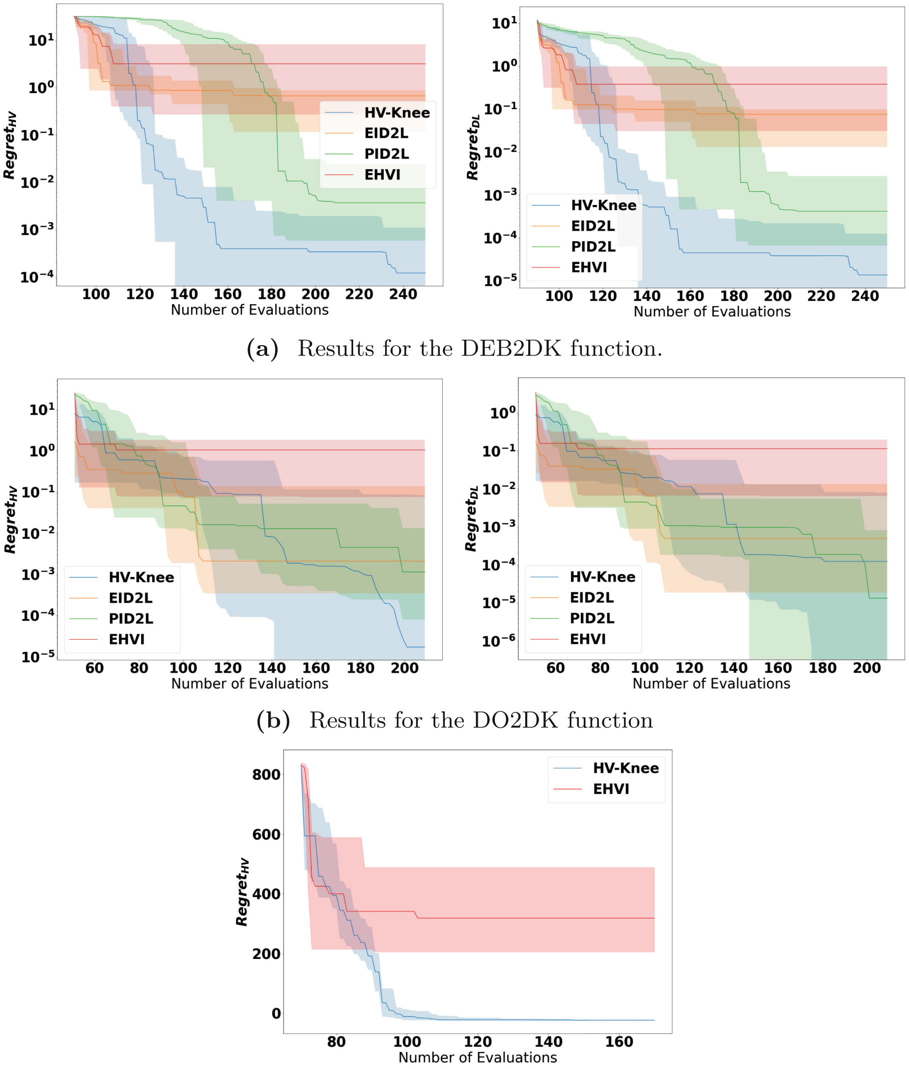
5 Results

The results are shown in Fig. 5. For DO2DK, a small value, 1.7×10^{-4} and 2×10^{-4}, is added to all Hyper-Volume and Distance Regrets, respectively, since the regret was negative for HV-knee and PID2L. This means that both PID2L and HV-Knee were successful in finding a point that performs better than the ground truth knee found by NSGA-II.

For DEB2DK the EID2L acquisition function shows a quick improvement in the early stages, but the HV-Knee and PID2L show a continuous improvement leading to better results near the end of the optimization process. All the acquisition functions exhibit the same behavior for DO2DK as well.

The last benchmark function, DEB3DK, has three objectives, and, hence, can only be used with the HV-Knee method. Note that the best regret is also negative for this case. The shaded area at the 170^{th} iteration is between -21 and -25, which means the HV-Knee acquisition function was able to find a point that performs much better than the knee point found by NSGA-II.

Figure 6 shows the extracted Pareto front for DTLZ2 [7] benchmark function. Pareto front of the DTLZ2 function is concave. In this case, the DM might prefer one of the extrema, known as the edge knee [24]. As all of the proposed acquisition functions are able to estimate the location of the extrema, and since the shape of the extracted Pareto front using the proposed acquisition functions is concave, the DM can choose one of the extrema as the final solution. PID2L (and also EID2L) return one of the extrema as the best point, but HV-Knee acquisition function prefers a point that is almost in the middle of the front.

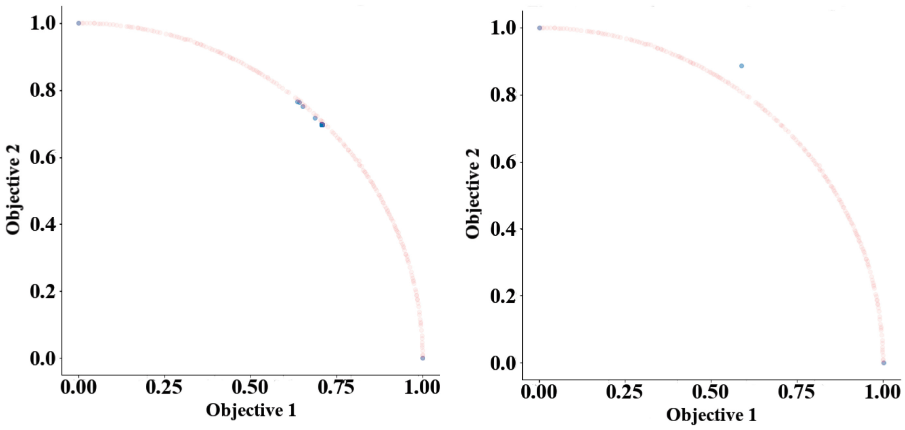


(a) Results for the DEB2DK function.

(b) Results for the DO2DK function

(c) Results for the DEB3DK function

Fig. 5. Results for the various benchmark function. Each experiment is repeated 15 times for the DO2DK and the DEB2DK functions, and 10 times for the DEB3DK function. The medians are denoted by the solid lines, while the shaded area represents the area between 20^{th} and 80^{th} percentile.



(a) HV-Knee Acquisition Function **(b)** PID2L Acquisition Function

Fig. 6. Extracted Pareto front of the DTLZ2 benchmark function using HV-Knee and PID2L acquisition function (in blue) and NSGA-II algorithm (in red). In both cases, it is clear that the Pareto front is concave, and since both acquisition functions are able to estimate the location of the extrema, one of the extrema can be chosen as the final solution. (Color figure online)

6 Conclusion and Next Steps

In this work we have proposed three acquisition functions for Bayesian Optimization to find attractive solutions (knees) on the Pareto front. These acquisition functions were able to identify the correct knee in a data-efficient manner, using about 200 evaluations (or even less) for a satisfying solution. Identifying a single solution is more efficient than the complete Pareto front, allowing Bayesian Optimization to scale up to more inputs and objectives. The proposed acquisition functions outperformed the ground truth obtained using an expensive NSGA-II approach. However, in some cases EAs are still preferred, for example, when at least one of the objective functions is hard to model with a GP (intractable function, high-input dimension), or when the evaluation of the objective functions is cheap and fast.

The developed acquisition functions alternated between two steps which is more time-consuming than it needs to be. More rigorous approaches will be developed to achieve an automatic balance between finding the extrema and identifying the knee. This will reduce the number of required evaluations further. Moreover, the proposed methods only focused on the global knee. If there is more than one knee in the Pareto front, they often remain unexplored. Current approaches will be extended, so other knees are also explored.

References

1. Berkeley, J., et al.: Trieste, February 2022. https://github.com/secondmind-labs/trieste
2. Bhattacharjee, K.S., Singh, H.K., Ryan, M., Ray, T.: Bridging the gap: many-objective optimization and informed decision-making. IEEE Trans. Evol. Comput. **21**(5), 813–820 (2017). https://doi.org/10.1109/TEVC.2017.2687320
3. Blank, J., Deb, K.: pymoo: multi-objective optimization in python. IEEE Access **8**, 89497–89509 (2020)
4. Branke, J., Deb, K., Dierolf, H., Osswald, M.: Finding knees in multi-objective optimization. In: Yao, X., et al. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 722–731. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30217-9_73
5. Chand, S., Wagner, M.: Evolutionary many-objective optimization: a quick-start guide. Surv. Oper. Res. Manage. Sci. **20**(2), 35–42 (2015)
6. Couckuyt, I., Deschrijver, D., Dhaene, T.: Fast calculation of multiobjective probability of improvement and expected improvement criteria for Pareto optimization. J. Global Optim. **60**(3), 575–594 (2013). https://doi.org/10.1007/s10898-013-0118-2
7. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable multi-objective optimization test problems. In: Proceedings of the 2002 Congress on Evolutionary Computation. CEC 2002 (Cat. No.02TH8600). vol. 1, pp. 825–830 (2002). https://doi.org/10.1109/CEC.2002.1007032
8. Emmerich, M., Giannakoglou, K., Naujoks, B.: Single- and multiobjective evolutionary optimization assisted by gaussian random field metamodels. IEEE Trans. Evol. Comput. **10**(4), 421–439 (2006). https://doi.org/10.1109/TEVC.2005.859463
9. Hakanen, J., Knowles, J.D.: On using decision maker preferences with ParEGO. In: Trautmann, H., et al. (eds.) EMO 2017. LNCS, vol. 10173, pp. 282–297. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-54157-0_20
10. Hua, Y., Liu, Q., Hao, K., Jin, Y.: A survey of evolutionary algorithms for multi-objective optimization problems with irregular pareto fronts. IEEE/CAA J. Automatica Sinica **8**(2), 303–318 (2021). https://doi.org/10.1109/JAS.2021.1003817
11. Indraneel, D.: On characterizing the "knee" of the Pareto curve based on normal-boundary intersection. Struct. Optim. **18**(2), 107–115 (1999). https://doi.org/10.1007/BF01195985
12. Jones, D.: A taxonomy of global optimization methods based on response surfaces. J. Global Optim. **21**, 345–383 (2001). https://doi.org/10.1023/A:1012771025575
13. Jones, D.R., Schonlau, M., Welch, W.J.: Efficient global optimization of expensive black-box functions. J. Global Optim. **13**, 455–492 (1998)
14. Knowles, J.: ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. IEEE Trans. Evol. Comput. **10**(1), 50–66 (2006)
15. Ma, X., Yu, Y., Li, X., Qi, Y., Zhu, Z.: A survey of weight vector adjustment methods for decomposition-based multiobjective evolutionary algorithms. IEEE Trans. Evol. Comput. **24**(4), 634–649 (2020). https://doi.org/10.1109/TEVC.2020.2978158
16. Minasny, B., McBratney, A.B.: The matérn function as a general model for soil variograms. Geoderma **128**(3), 192–207 (2005). https://doi.org/10.1016/j.geoderma.2005.04.003, pedometrics 2003
17. Mockus, J.: Bayesian Approach to global Optimization: Theory and Applications, vol. 37. Springer (1989). https://doi.org/10.1007/978-94-009-0909-0

18. Mockus, J., Tiesis, V., Zilinskas, A.: The application of Bayesian methods for seeking the extremum. Towards Global Optim. **2**(117–129), 2 (1978)
19. Picheny, V., Wagner, T., Ginsbourger, D.: A benchmark of kriging-based infill criteria for noisy optimization. Struct. Multidiscip. Optim. **48**, 607–626 (2013). https://doi.org/10.1007/s00158-013-0919-4
20. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning, 1st edn. The MIT Press, Cambridge (2005)
21. Ray, T., Singh, H.K., Rahi, K.H., Rodemann, T., Olhofer, M.: Towards identification of solutions of interest for multi-objective problems considering both objective and variable space information. Appl. Soft Comput. **119**, 108505 (2022). https://doi.org/10.1016/j.asoc.2022.108505
22. Rojas-Gonzalez, S., Van Nieuwenhuyse, I.: A survey on kriging-based infill algorithms for multiobjective simulation optimization. Comput. Oper. Res. **116**, 104869 (2020)
23. Trivedi, A., Srinivasan, D., Sanyal, K., Ghosh, A.: A survey of multiobjective evolutionary algorithms based on decomposition. IEEE Trans. Evol. Comput. **21**(3), 440–462 (2017). https://doi.org/10.1109/TEVC.2016.2608507
24. Yu, G., Jin, Y., Olhofer, M.: Benchmark problems and performance indicators for search of knee points in multiobjective optimization. IEEE Trans. Cybern. **50**(8), 3531–3544 (2020). https://doi.org/10.1109/TCYB.2019.2894664
25. Yu, G., Ma, L., Jin, Y., Du, W., Liu, Q., Zhang, H.: A survey on knee-oriented multi-objective evolutionary optimization. IEEE Trans. Evol. Comput. 1 (2022). https://doi.org/10.1109/TEVC.2022.3144880
26. Zhang, K., Yen, G.G., He, Z.: Evolutionary algorithm for knee-based multiple criteria decision making. IEEE Trans. Cybern. **51**(2), 722–735 (2021). https://doi.org/10.1109/TCYB.2019.2955573
27. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C., da Fonseca, V.: Performance assessment of multiobjective optimizers: an analysis and review. IEEE Trans. Evol. Comput. **7**(2), 117–132 (2003). https://doi.org/10.1109/TEVC.2003.810758