Decision Maker Preferences in Surrogate-based Multi-Objective Optimization: A Survey

Sasan Amini[1], Antonio Candelieri[2], and Inneke Van Nieuwenhuyse[1]

[1] Computational Mathematics Research Group, Data Science Institute, Hasselt University, 3590 Diepenbeek, Belgium
{sasan.amini, inneke.vannieuwenhuyse}@uhasselt.be

[2] Department of Economics, Management and Statistics, University of Milano-Bicocca, 20126 Milano, Italy
antonio.candelieri@unimib.it

Abstract. Multi-objective optimization problems are highly relevant in practice, and algorithms to solve these types of problems abound in the literature. This survey focuses explicitly on surrogate-based algorithms that use the decision-maker's preference information to guide the search toward the most preferred areas of the Pareto front. Considering such preferences not only facilitates the decision-making process for the user but also helps the analyst to save expensive computational budget. The way in which user preference information is handled in the algorithms differs across publications. We classify them according to the type and timing of the preference information. We provide an overview of the state-of-the-art, highlight the most important shortcomings in the literature, and present promising directions for further research.

Keywords: Multi-objective optimization · Surrogate-based optimization · Pareto-front · Preferences

1 Introduction

There has been an increasing amount of literature on multi-objective (MO) optimization problems in recent decades. MO problems indeed occur frequently in real-life settings, for instance, in logistics and supply chain management [37], engineering [51], and machine learning [4]. In general, a MO problem can be expressed as "min $f_1(x), f_2(x), ..., f_n(x)$" where x is a d-dimensional decision vector in the decision space $D \subset R^d$, $n \geq 2$ is the number of objective functions. Typically, the objective functions are conflicting. The goal of the analyst then is to find the set of *non-dominated or Pareto-optimal solutions*, i.e., those solutions where no single objective can be improved without deteriorating any other objective(s). The set of Pareto optimal solutions in the decision space is referred to as the *Pareto set*; the evaluation of these Pareto optimal solutions in objective space yields the *Pareto front* [14]. As an illustration, Figure 1 shows three points (A, B, and C) in a bi-objective problem that are *not* Pareto optimal (however, point B dominates point C, and point A dominates both points B and C). The figure also shows the *ideal* point (which combines the optima of the individual objective functions, see [34]), and the *Nadir point* (which combines the worst objective values of the Pareto optimal solutions). MO algorithms estimate the Pareto front, known as the experimental Pareto front, and their performance is measured by how closely this estimation aligns with the true Pareto front.

Many MO algorithms do not include any decision maker (DM) preferences in the search process; consequently, they aim to yield an approximation of the whole Pareto front as output. In such algorithms, the DM selects the most preferred solution(s) *a posteriori*, based on the available trade-off information. This leads to two inefficiencies. Firstly, it puts a big burden on the DM, who is confronted with a (potentially large) set of solutions. This is particularly problematic in settings with many objectives (i.e., large n) and/or a relatively large search space (i.e., large d), as the number of Pareto optimal solutions then grows very fast, and they can no longer be visualized straightforwardly [12]. This makes it much harder for the DM to detect the preferred set. More importantly, estimating the entire Pareto front causes an inefficient use of the computational budget, as the process generates potentially many unpreferred solutions. This is particularly relevant in settings with limited computational budget, due to time and/or cost constraints [19]. Consequently, efforts have been made to include DM preferences in the search process, such that the resulting solutions focus on that area of the front that is relevant for the DM (also referred to as the *region of interest* (RoI), e.g., [11]). The resulting algorithms are of high importance in many industrial optimization problems, where function evaluations may require expensive (physical or computer) experiments: these algorithms help to reduce the number of experiments and facilitate the decision-making process. Most of the efforts to incorporate decision-maker (DM) preferences have been conducted within the

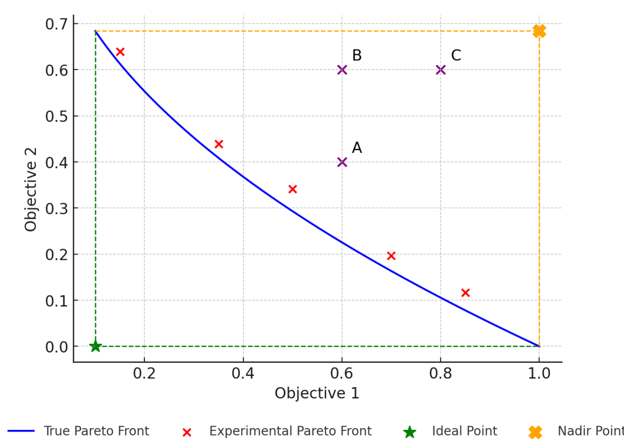

Fig. 1. Illustration of Pareto front in a bi-objective minimization problem

framework of evolutionary algorithms, as highlighted in the overviews provided by [8] and [45].

In optimization problems with expensive-to-evaluate functions, it is even more vital to focus efforts on finding solutions preferred by the DM [21]. Yet, in spite of the increased interest in surrogate-based algorithms to solve MO problems in recent years, algorithms incorporating DM preferences remain very scarce. Surrogate-based optimization algorithms are specifically designed to reduce the computational cost of optimization by approximating the objective function [18], and the inclusion of DM preferences naturally aligns with this goal by narrowing the search space to the most relevant regions. Despite their conceptual alignment, surprisingly, considerably fewer studies have focused on incorporating DM preferences within the surrogate-based optimization literature. This gap is especially striking given the potential of surrogate-based approaches to significantly enhance data efficiency while facilitating targeted exploration of the decision space.

This article surveys *surrogate-based* algorithms for MO optimization with DM preferences; It explicitly focuses on algorithms that take into account DM preferences *during the search*; hence, we do not consider *a posteriori* selection. The classification proposed in this article is in line with the classification proposed by [48] and [2], two papers that focus solely on interactive multi-objective optimization, mostly discussing evolutionary algorithms and just a few surrogate-based algorithms. To the best of our knowledge, our work presents the first comprehensive survey that particularly focuses on the surrogate-based optimization algorithms and discusses how the incorporation of the decision-maker preferences (either a priori or interactively) can contribute to the main goal of increasing data efficiency in these algorithms. We focused on articles indexed in the Scopus and Web of Science databases, book chapters, and conference proceedings to identify the most recent research directions. The search terms such as "multi-objective optimization", "surrogate-based optimization", "meta model-based optimization", and "user preference" were used to select a primary set of papers, which were then checked for relevance. By applying the ancestry approach [6], we selected a set of 13 articles (Table 1) that are relevant for this review.

2 Surrogate-based multi-objective optimization

In many optimization problems, the objectives are evaluated using computationally expensive simulations or costly physical experiments [18, 9]. This limits the number of evaluations that may be performed [23]. Evolutionary approaches are ill-suited in such cases, as they require many function evaluations to solve the problem. Surrogate-based algorithms, particularly those using Gaussian Process Regression (GPR) and Radial Basis Functions (RBF), have been widely used for solving single-objective expensive optimization problems [44, 23]. In the context of multi-objective optimization, research on surrogate-based methods has gained increasing attention in recent years, with various approaches proposed in the literature [23].

The main idea in surrogate-based optimization is to approximate the expensive objectives by means of one or multiple surrogates (also referred to as meta-models), and then gain more information about these unknown objective functions *efficiently* (i.e., with a limited number of additional function evaluations), by exploiting the knowledge present in the surrogate(s) [31]. A surrogate model can be considered as a simplified and computationally controllable model of the true objective function(s). Contrary to the evolutionary approaches, surrogate-based approaches can predict the unknown function outputs at unobserved points in the design space, by exploiting the model information.

Figure 2 shows the general steps in a surrogate-based MO optimization algorithm with DM preferences. The first step is the generation of an initial design. The purpose of this step is to create a space-filling set of design points chosen throughout the domain of the problem; Latin Hypercube Sampling is the most common choice in the literature for its space-filling and non-collapsing features. After evaluating the objective functions on the initial design, the surrogate model(s) are fit to the observed input/output data. Next, an *infill criterion* (also referred to as *acquisition function* in Bayesian optimization [20], and *sampling strategy* in Radial Basis Functions literature), is used to determine which input combination to evaluate next: this criterion uses the surrogate model information in such a way that it typically balances exploitation (i.e., sampling in areas with promising predictor values) and exploration (sampling in areas with high uncertainty for the objective function(s)). The point where the acquisition function is maximized is chosen as the next one to sample and is referred to as the so-called *infill point*.

Depending on the properties of the problem, different infill criteria can be used; common choices in the literature are Expected Improvement (EI), Probability of Improvement (PoI), and (relatively recently) Entropy Search (ES) [20]. To integrate the DM's preference information in the algorithms, the acquisition function can be modified such that the search of the infill point is guided to the most interesting areas of the Pareto front. Section 4 discusses such acquisition functions. After evaluating the infill point using the expensive function, the surrogate model is updated, and the algorithm continues until a stopping criterion (e.g., computational budget) is met. Finally, the algorithm identifies the Pareto front and the corresponding Pareto set.

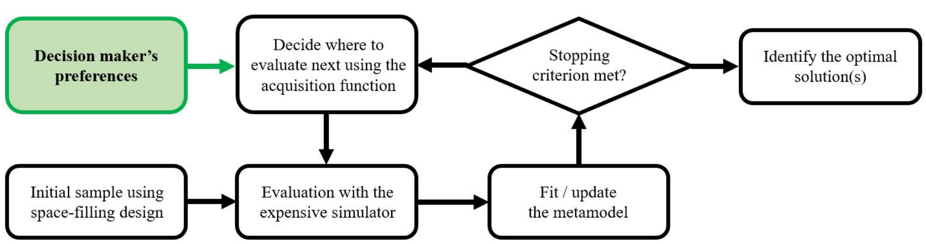

Fig. 2. Main steps in surrogate-based MO optimization algorithms with DM preferences

3 Preference information: type and timing

The way in which DM preference information shows up in the algorithms differs in two respects (see Figure 3): (1) *when* the preference information is collected (see Section 3.1), and (2) which *type of preference information* is collected (see Section 3.2). The collected preference information then is considered in an overall utility function. A *utility function* quantitatively evaluates solutions for their conformity with the DM's preferences. Often, the utility function is not explicitly known; several methods have been developed in the literature to estimate the DM's utility function based on the collected preference information (see Section 3.2).

3.1 Timing of the preference information

MO algorithms that take into account DM preferences during the search can do so *a priori* (meaning that preferences are known explicitly *upfront*), or *interactively* (meaning that preference information is collected, often iteratively, *during* the optimization). A priori methods are efficient and straightforward in theory; yet, in practice, it is often difficult (and in cases with a large number of objectives, even almost impossible) for the decision-maker to provide specific preference information upfront. In an interactive algorithm, contrarily, the DM progressively specifies preference information during the search, as more insight about the problem is gained [39]. This is particularly relevant in black-box problems, where a priori knowledge is unavailable. Most interactive methods also provide the opportunity for the DM to change her preferences during the search and, as a result, the algorithm's parameters; this is evidently not the case in a priori algorithms. In practice, the number of interactions with the DM is often limited. A disadvantage of interactive methods is that they may collect inconsistent user preferences; especially when the number of interactions with the decision maker increases, the preferences provided by the DM may contain contradictory information due to the cognitive burden of the DM [27]. Inconsistent user preferences emerging over the optimization process of black-box objectives could be due to the fact that the DM learns more about the problem after every query.

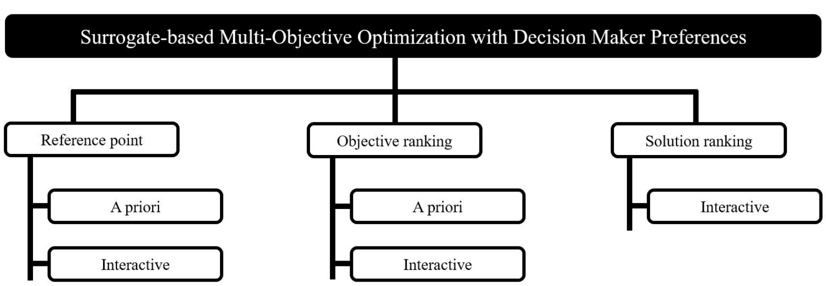

Fig. 3. DM preference information: type and timing

3.2 Types of preference information

We classify the preference information of the DM into three categories: (*i*) reference point information (i.e., the DM provides a goal to meet, or defines the worst acceptable values), (*ii*) objective ranking (i.e., the DM assigns weights to the objectives), or (*iii*) solution ranking (i.e., the DM either compares the proposed solutions pairwise or ranks multiple solutions). This section discusses the main features of these information types, with their pros and cons. Furthermore, we discuss how this information is integrated into utility functions to guide the optimization algorithms through the search.

Reference points: As expressing preferences through objective function values is cognitively easy for the DM [24], reference point methods are the most straightforward and common approach in the literature. Two types of reference points (also known as *goals* or *target values*) are used: (*i*) those defining an *aspiration level*, for instance, when trying to keep costs within a certain budget, or (*ii*) those defining a *reservation level*, i.e., setting a *minimum acceptable level*; for instance, when trying to achieve at least a certain amount of profit. Reference points can be varied during the search via interaction with the DM. After each interaction, a set of solutions is then presented to the DM, and she can modify the reference point(s). Reference points can also be defined by the analyst (in case of a non-expert DM; see, e.g., [22].

With reference point information, an *achievement function* is commonly used as an objective to be minimized or maximized. If the reference point defines an aspiration level, the algorithm will try to minimize the deviation between this point and the proposed solutions; in the case of a reservation level, the deviation should be maximized. Equation (1) represents a widely used achievement function [47] from mathematical optimization literature, where r_i stands for the reference point for objective i (aspiration level, provided by the DM) and the λ_i are normalising parameters. The solution found by solving problem (1) is Pareto optimal if all the deviations from the reference point, $f_i(x) - r_i$, are strictly positive [40]. One can easily adapt the conditions for a maximization problem. The same concept has been utilized to develop surrogate-based optimization algorithms where $f_i(x)$ is replaced by its expectation as provided by the associated surrogate model.

$$\begin{aligned} \min_{x \in D \subset \mathbb{R}^d} \; & \sum_{i=1}^{n} \lambda_i \big(f_i(x) - r_i\big) \\ s.t.\; & f_i(x) - r_i \geq 0 \quad \forall\, i = 1, ..., n \end{aligned} \tag{1}$$

The objective function in (1) is a weighted L1 norm between the vector-valued function at x, that is $\mathbf{f}(x)$, and the n-dimensional reference point $\mathbf{r}$. However, other norm could be considered, such as L2 norm (i.e., Euclidean distance).

Objective ranking: Preference information can also be articulated by assigning *weights* to the objectives. The weights then reflect the importance of each objective for the DM, and are used to aggregate the multiple objectives into a single objective (this presumes that the objective functions are commensurable; the resulting function is also referred to as the *scalarization function*). The weighted sum scalarization function in equation (2) is one of the most widely used scalarization functions; other common scalarization functions are the weighted Tchebycheff function, see equation (3), and the augmented Tchebycheff, see equation (4). In these equations, n is the number of objective functions, the weights λ_i are in range $[0,1]$ and $\sum_{i=1}^{n} \lambda_i = 1$. In equations (3) and (4), ρ is a small positive value, and z is the reference point.

$$f(x) = \sum_{i=1}^{n} \lambda_i f_i(x) \tag{2}$$

$$f(x) = \{\max_i \quad \lambda_i(f_i(x) - z_i)\} \tag{3}$$

$$f(x) = \{\max_i \quad \lambda_i(f_i(x) - z_i) + \rho \sum_{i=1}^{n} \lambda_i f_i(x)\} \tag{4}$$

The weights can be defined a priori, either by choosing a single weight vector (yielding a single Pareto optimal solution), or multiple vectors (in view of finding a set of desirable solutions, though there is no theoretical guarantee that multiple vectors will indeed result in multiple different solutions [40]). Alternatively, the weights can be varied during the search via interaction with the DM. After each iteration, a set of solutions is then presented to the DM, and she can modify the weights. The main drawback of scalarization is that it only generates supported solutions; i.e., algorithms using scalarization functions can only investigate the convex parts of the Pareto front, while solutions in non-convex areas cannot be generated by this method (except when using the augmented Tchebycheff function (equation 4); some authors tried to mitigate this problem in their research, see for instance [13] and [46]). Moreover, when facing a large number of objectives, weight-based methods typically become ineffective as it becomes too challenging for the DM to steer the search process towards the region of interest through a well-chosen weighting scheme [33]. Finally, scalarization is not always possible as the different objective functions may not be commensurable [38].

Solution ranking: In this approach, the DM compares sets of objective values belonging to different solutions, resulting in a ranking or classification. Usually, in the literature, *pairwise* comparison of solutions is applied, which may result in (*i*) one solution being preferred to the other, (*ii*) the DM being indifferent between both solutions. Solution ranking needs not necessarily be pairwise: the DM can also be asked to reveal her preference information on a set of multiple solutions, e.g., to classify them into preferred versus not-preferred solutions or to determine the best (or the worst) solution. Based on such ranking information, the algorithm tries to *learn* an overall utility function. As the number of objectives increases, it typically becomes impossible for the DM to compare the solutions in a consistent way.

4 Results and discussion

In the literature, all possible combinations of preference information and timing occur (apart from *a priori* timing and *solution ranking*, for obvious reasons), within different algorithmic approaches. In this section, we discuss these links between algorithmic approaches and preference information types in more detail. Table 1 summarizes the algorithms surveyed in this section.

Table 1. Overview of Surrogate-Based Multi-Objective Optimization Algorithms Incorporating Decision Maker Preferences

Algorithm	Surrogate	Timing	Information Type
R/C-mEI [21]	GPR	a priori	reference point
EAPLI [25]	GPR	a priori	reference point
TEHVI-EGO [50]	GPR	interactive	reference point
iParEGO [24]	GPR	interactive	reference point
SURROGATE-ASF [39]	RBF	interactive	reference point
EWHI [16]	GPR	a priori	objective ranking
MOBO-PC [1]	GPR	a priori	objective ranking
SAMOO-RBF [29]	RBF	a priori	objective ranking
MOBO-RS [36]	GPR	interactive	objective ranking
EI-UU/TS-UU [5]	GPR	interactive	solution ranking
Modified EI-UU[43]	GPR	interactive	solution ranking
Interactive MOBO [42]	GPR	interactive	solution ranking
TRIPE [26]	GPR	interactive	solution ranking

4.1 Algorithms with reference point

Both interactive and a priori algorithms exist that work with reference point information. The algorithm proposed in [21] uses the multiplied Expected Improvements (mEI) infill criterion to focus the search on the preferred region of the Pareto front (which is determined by an aspiration point, which can be explicitly provided by the DM, or implicitly defined by the analyst). This infill criterion proposed by the authors can be considered as a very particular instance of the one developed in [50], and is equivalent to a truncated EHI with infinite lower bounds (where the Expected Hypervolume Improvement (EHI) is an acquisition function that quantifies the volume of objective space dominated by the Pareto front—resulting from the evaluation of a new candidate solution). Yet, the mEI is simpler to tune and cheaper to compute, since it has a closed-form expression for any number of objectives. Moreover, its complexity grows only linearly in the number of objectives and is independent of the number of non-dominated solutions. The EAPLI algorithm proposed in [25] uses reference vectors defined based on the provided a priori reference point information. These reference vectors are used to decompose the original MO problem into a number of single-objective subproblems: the acute angle between the candidate solutions and the reference vectors indicates the degree of satisfaction of the user preferences.

Three interactive algorithms using reference point information are included in this survey. By asking the DM to determine both the aspiration and reservation, reference point information can define a preferred range for each objective function (as in [50] and [24]). The preference region in [50] is represented by truncating the objective space; the infill criterion used in the algorithm is the truncated expected hypervolume improvement (TEHVI) [49], which combines information on the predictors for the objective functions, the variance of the predictors, and the preferred region. The authors of [24] extend the work by [30], developing an interactive version of the ParEGO algorithm (iParEGO) that replaces the weighted Tchebycheff scalarization function with an achievement function, and uses reference points instead of weights to include DM preferences. The DM can change her preferences, so as to visit areas of the solution space that previously were not considered. The DM iteratively provides feedback on a subset of non-dominated solutions by specifying the preferred range for each objective (by determining aspiration and reservation points). The algorithm samples reference points within these ranges, and uses a GPR model to approximate an achievement function (single-objective) to solve the problem.

The algorithm proposed by [39] is the only algorithm in this part of the literature that uses RBF instead of GPR as a surrogate model. This algorithm consists of an initialisation and decision-making phase. In the initialisation phase, the decision space is decomposed into a finite number of hyper-boxes. For each hyper-box, a single RBF surrogate of the achievement function is built using a set of predetermined reference points. These surrogate functions are used in the decision-making phase, where the algorithm interacts with the DM and collects a reference point at each iteration. The algorithm uses this reference point and the surrogate model to find an optimal solution, which is then evaluated with the original functions and shown to the DM. This loop (asking for a new reference point, solving the surrogate optimization problem, and evaluating the resulting solution with the original objective functions) iterates until the DM is satisfied.

4.2 Algorithms with objective ranking

All surrogate-based algorithms with objective ranking surveyed in this article use a priori preference information. [16] propose the expected weighted hypervolume improvement (EWHI): the weight functions are defined by the DM, and the surrogate model exploits these functions to focus on the most preferred regions. The authors use examples of weight functions suggested in [52] to demonstrate the effectiveness of the proposed algorithm on a bi-objective optimization problem.

The authors highlight the relation between EWHI and the *expected hypervolume improvement* (EHVI) criterion proposed in [15]. The other algorithm in this group, the MOBO-PC algorithm proposed in [1], formulates a new infill criterion based on expected improvement in dominated hypervolume (EHI). The algorithm formulates preference-order constraints on the objectives, by asking DM to compare objectives w.r.t. their importance. In the infill criterion, the contribution of each point to the hypervolume is weighted by the probability that it satisfies the constraints.

Paria et al. [36] propose an algorithm that uses random scalarizations of the objectives to flexibly explore the Pareto front. The method requires the user to specify a class of scalarization functions and a prior distribution over weights, which guide the optimization process. Weights are sampled iteratively, and the posterior distributions of the objectives are computed. The resulting scalarized acquisition function is used to select the next evaluation point. While the authors report promising results, particularly in terms of flexibility and computational efficiency, the work does not provide a systematic method for estimating or updating the distribution of weights. This omission leaves the responsibility of weight specification to the user, which could limit the algorithm's applicability in settings where domain expertise is lacking or preferences are dynamic.

The only algorithm using RBF as a surrogate model in this part of the literature is [29]. This algorithm consists of three phases. In the first phase, the algorithm looks for a Pareto optimal solution for the surrogate model using an initial set of points, and the weights provided by the DM. This solution is then sampled and evaluated via the original function. The second phase aims at an exploration of the Pareto front, using the *density function* (unlimited repetition of this step may result in exploring the whole Pareto front). Finally, the *Pareto-fitness function* is used to add new samples around the Pareto optimal solution found, without solving the multi-objective optimization problem. The loop of these three phases iterates until the termination criterion (i.e., computational budget) is met.

4.3 Algorithms with solution ranking

Astudillo and Frazier [5] propose a framework for multi-attribute Bayesian optimization with interactive preference learning. As previously discussed, the solution ranking approach is inherently suited to interactive settings and cannot be effectively implemented in a non-interactive manner. The algorithm learns a Bayesian posterior over the decision maker's utility function based on pairwise comparisons, collected iteratively during the optimization process. The objectives are modeled using a multi-output Gaussian Process, and two acquisition functions, Expected Improvement under Utility Uncertainty (EI-UU) and Thompson Sampling under Utility Uncertainty (TS-UU), are introduced to guide the search. While EI-UU demonstrates superior performance in numerical experiments, its computational complexity presents a challenge.

Ungredda et al. [43] propose a single-interaction multi-objective Bayesian optimization algorithm. Instead of presenting the DM with a discrete set of

Pareto-optimal solutions at the end of the optimization process, the algorithm involves the DM once before the end of the evaluation budget. The DM selects their most preferred solution from an approximated continuous Pareto front, constructed using surrogate models. This preference information is then used to focus the remaining evaluations on finding the DM's most preferred solution or an even better one. The authors demonstrate empirically that this approach reduces opportunity cost compared to traditional methods, though it relies on the accuracy of the approximated Pareto front and the preference elicitation step. In another paper by Ungredda and Branke [42], they propose a framework that, instead of following predefined intervals or patterns for querying preferences, the algorithm dynamically decides whether to evaluate a new solution or elicit preferences. Preferences are collected through pairwise comparisons, which are used to construct a Bayesian posterior over the decision maker's utility function. While the approach demonstrates effective reduction in opportunity cost and adaptability to problem characteristics, its reliance on accurate preference modeling and potential computational complexity raises questions about scalability in real-world, high-dimensional scenarios.

Finally, two scalarization-based Bayesian algorithms for interactive multi-objective optimization have been proposed in [26]. The TRIPE method employs a triangulation-based approach to explore regions near the decision maker's preferred solution in the input space. It is hyperparameter-free but is constrained by scalability to problems with more than five input dimensions. In contrast, WAPE, the other algorithm they proposed, is better suited for higher-dimensional problems but incurs greater computational costs due to its reliance on a secondary surrogate model and acquisition function.

5 Conclusion and further research directions

In this paper, we reviewed algorithms that explicitly consider DM preferences in the search for Pareto optimal solutions. Despite the numerous works in evolutionary algorithm literature, only very recently surrogate-based approaches have been proposed. Evolutionary approaches require a considerable amount of function evaluations to converge to the DM's region of interest; surrogate-based approaches, by contrast, are much more data-efficient, and thus more suitable in problems where objective evaluations are expensive. While the results so far are promising, the capability of surrogate-based approaches to work with different types of preference information has not yet been well studied. Many infill criteria exploit scalarization methods to account for the DM preferences; care should thus be taken to choose a scalarization method that is able to also detect non-convex parts of the front, such that no solutions are overlooked. The results show that Gaussian Process Regression is by far the most often used surrogate model type for solving MO problems with DM preferences; RBF has also been used but to a much lower extent.

While *a priori* methods are a straightforward approach to incorporate DM preferences in MOPs, the DM does not necessarily know the possibilities and limitations of the problem beforehand, resulting in expectations that are either too optimistic or too pessimistic. In an iterative approach, by contrast, the DM can learn during the solution process. The implementation of interactive algorithms generally requires more caution from the analyst, though: recent research on the role of the DM in interactive MOPs [32] has revealed that the DM's inputs have a significant effect on the search process, and consequently on the solutions obtained. Furthermore, inconsistencies in the preference updates by the DM have a clear and irreversible negative effect on the optimization process. While the DM learns more about the target problem, she may change her preferences over the sequential optimization process and, consequently, the weights in the scalarization of the multi-objective function. However, this means that the objective function might be *dynamic* because it changes over time. Research on Dynamic Bayesian Optimiziation (DBO) [35, 3] could represent a promising perspective for the research topic of this survey. Interactive algorithms also have a higher risk of converging prematurely to only a subset of the solutions that are in the DM's region of interest: indeed, the DM is likely to provide preference information that builds on the "good" solutions that have already been presented, which may limit explorations into completely novel (yet potentially interesting) areas of the search space. Real-world decision scenarios may involve a group of DMs, which may differ in terms of priorities, the perception of the problem, and the degree to which they can impact the collective outcome. Although such problems have been investigated in a separate stream of literature (i.e., the group decision making literature), a few recent papers ([17]; [41]; [7]) suggest to combine insights from group decision making and multiple objective decision making. Considering its relevance in practice, handling group preferences in MOP will likely be a growing research field in the future.

Last but not least, it is quite surprising that all algorithms that appeared so far (with the exception of [36]) assume that the objectives can be observed with perfect accuracy. In many real-life settings, these observations are noisy: e.g., noise can result from sampling variance, when the objective evaluations result from a stochastic simulation [10, 28]. In settings where preferences have to be learned based on the DM's information, uncertainty may also occur in the feedback that the algorithm receives from the DM: so far, [5] are the only authors to acknowledge this type of uncertainty in the learned preferences. While we acknowledge that accounting for these different types of uncertainties in the optimization approach is highly challenging, we strongly believe that such efforts are necessary to further increase the relevance of MO algorithms for solving practical problems. As such, we consider this to be an important avenue for future research.

Acknowledgments. This work was supported by the Flanders Artificial Intelligence Research Program (FAIR2) and the Research Foundation Flanders (FWO) (grant number: G0A4624N). The first author also received a travel grant from the Research Foundation Flanders (FWO) to participate in the conference.

References

1. Abdolshah, M., Shilton, A., Rana, S., Gupta, S., Venkatesh, S.: Multi-objective bayesian optimisation with preferences over objectives. Advances in neural information processing systems **32** (2019)
2. 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)
3. Aglietti, V., Dhir, N., González, J., Damoulas, T.: Dynamic causal bayesian optimization. Advances in Neural Information Processing Systems **34**, 10549–10560 (2021)
4. Alexandropoulos, S.A.N., Aridas, C.K., Kotsiantis, S.B., Vrahatis, M.N.: Multi-objective evolutionary optimization algorithms for machine learning: A recent survey. In: Approximation and Optimization, pp. 35–55. Springer (2019)
5. Astudillo, R., Frazier, P.: Multi-attribute bayesian optimization with interactive preference learning. In: International Conference on Artificial Intelligence and Statistics. pp. 4496–4507. PMLR (2020)
6. Atkinson, K.M., Koenka, A.C., Sanchez, C.E., Moshontz, H., Cooper, H.: Reporting standards for literature searches and report inclusion criteria: making research syntheses more transparent and easy to replicate. Research synthesis methods **6**(1), 87–95 (2015)
7. Balderas, F., Fernández, E., Cruz-Reyes, L., Gómez-Santillán, C., Rangel-Valdez, N.: Solving group multi-objective optimization problems by optimizing consensus through multi-criteria ordinal classification. European Journal of Operational Research **297**(3), 1014–1029 (2022)
8. Bechikh, S., Kessentini, M., Said, L.B., Ghédira, K.: Preference incorporation in evolutionary multiobjective optimization: a survey of the state-of-the-art. In: Advances in Computers, vol. 98, pp. 141–207. Elsevier (2015)
9. Bhosekar, A., Ierapetritou, M.: Advances in surrogate based modeling, feasibility analysis, and optimization: A review. Computers & Chemical Engineering **108**, 250–267 (2018)
10. Boesel, J., Nelson, B.L., Kim, S.H.: Using ranking and selection to "clean up" after simulation optimization. Operations Research **51**(5), 814–825 (2003)
11. Branke, J., Deb, K.: Integrating user preferences into evolutionary multi-objective optimization. In: Knowledge incorporation in evolutionary computation, pp. 461–477. Springer (2005)
12. Branke, J., Deb, K., Miettinen, K., Słowiński, R.: Multiobjective optimization: Interactive and evolutionary approaches, vol. 5252. Springer Science & Business Media (2008)
13. Chen, Y.L., Liu, C.C.: Multiobjective var planning using the goal-attainment method. IEE Proceedings-Generation, Transmission and Distribution **141**(3), 227–232 (1994)
14. Emmerich, M.T., Deutz, A.H.: A tutorial on multiobjective optimization: fundamentals and evolutionary methods. Natural computing **17**(3), 585–609 (2018)
15. Emmerich, M.T., Giannakoglou, K.C., Naujoks, B.: Single-and multiobjective evolutionary optimization assisted by gaussian random field metamodels. IEEE Transactions on Evolutionary Computation **10**(4), 421–439 (2006)
16. Feliot, P., Bect, J., Vazquez, E.: User preferences in bayesian multi-objective optimization: the expected weighted hypervolume improvement criterion. In: International Conference on Machine Learning, Optimization, and Data Science. pp. 533–544. Springer (2018)

17. Fernández, E., Gómez-Santillán, C., Rangel-Valdez, N., Cruz-Reyes, L.: Group multi-objective optimization under imprecision and uncertainty using a novel interval outranking approach. Group Decision and Negotiation **31**(5), 945–994 (2022)
18. Forrester, A.I., Keane, A.J.: Recent advances in surrogate-based optimization. Progress in aerospace sciences **45**(1-3), 50–79 (2009)
19. Frazier, P.I.: Bayesian optimization. In: Recent advances in optimization and modeling of contemporary problems, pp. 255–278. Informs (2018)
20. Garnett, R.: Bayesian optimization. Cambridge University Press (2023)
21. 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 (2020)
22. Gaudrie, D., Riche, R.L., Picheny, V., Enaux, B., Herbert, V.: Budgeted multi-objective optimization with a focus on the central part of the pareto front–extended version. arXiv preprint arXiv:1809.10482 (2018)
23. Gramacy, R.B.: Surrogates: Gaussian process modeling, design, and optimization for the applied sciences. Chapman and Hall/CRC (2020)
24. Hakanen, J., Knowles, J.D.: On using decision maker preferences with parego. In: International Conference on Evolutionary Multi-Criterion Optimization. pp. 282–297. Springer (2017)
25. He, Y., Sun, J., Song, P., Wang, X., Usmani, A.S.: Preference-driven kriging-based multiobjective optimization method with a novel multipoint infill criterion and application to airfoil shape design. Aerospace Science and Technology **96**, 105555 (2020)
26. Heidari, A., Gonzalez, S.R., Dhaene, T., Couckuyt, I.: Data-efficient interactive multi-objective optimization using parego. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 519–526. Springer (2023)
27. Jaszkiewicz, A., Branke, J.: Interactive multiobjective evolutionary algorithms. In: Multiobjective optimization, pp. 179–193. Springer (2008)
28. Kim, S.H., Nelson, B.L.: Selecting the best system. Handbooks in operations research and management science **13**, 501–534 (2006)
29. Kitayama, S., Srirat, J., Arakawa, M., Yamazaki, K.: Sequential approximate multi-objective optimization using radial basis function network. Structural and Multidisciplinary Optimization **48**(3), 501–515 (2013)
30. 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)
31. Knowles, J., Nakayama, H.: Meta-modeling in multiobjective optimization. In: Multiobjective optimization, pp. 245–284. Springer (2008)
32. Lai, G., Liao, M., Li, K.: Empirical studies on the role of the decision maker in interactive evolutionary multi-objective optimization. In: 2021 IEEE Congress on Evolutionary Computation (CEC). pp. 185–192. IEEE (2021)
33. Li, K., Liao, M., Deb, K., Min, G., Yao, X.: Does preference always help? a holistic study on preference-based evolutionary multiobjective optimization using reference points. IEEE Transactions on Evolutionary Computation **24**(6), 1078–1096 (2020)
34. Miettinen, K.: Nonlinear multiobjective optimization, vol. 12. Springer Science & Business Media (2012)
35. Nyikosa, F.M., Osborne, M.A., Roberts, S.J.: Bayesian optimization for dynamic problems. arXiv preprint arXiv:1803.03432 (2018)

36. Paria, B., Kandasamy, K., Póczos, B.: A flexible framework for multi-objective bayesian optimization using random scalarizations. In: Uncertainty in Artificial Intelligence. pp. 766–776. PMLR (2020)
37. Pudasaini, P.: Integrated planning of downstream petroleum supply chain: A multi-objective stochastic approach. Operations Research Perspectives **8**, 100189 (2021)
38. Rachmawati, L., Srinivasan, D.: Preference incorporation in multi-objective evolutionary algorithms: A survey. In: 2006 IEEE International Conference on Evolutionary Computation. pp. 962–968. IEEE (2006)
39. Tabatabaei, M., Hartikainen, M., Sindhya, K., Hakanen, J., Miettinen, K.: An interactive surrogate-based method for computationally expensive multiobjective optimisation. Journal of the Operational Research Society **70**(6), 898–914 (2019)
40. Talbi, E.: Metaheuristics: From design to implementation. John Wiley & Sons google schola **2**, 268–308 (2009)
41. Tomczyk, M.K., Kadziński, M.: Interactive co-evolutionary multiple objective optimization algorithms for finding consensus solutions for a group of decision makers. Information Sciences **616**, 157–181 (2022)
42. Ungredda, J., Branke, J.: When to elicit preferences in multi-objective bayesian optimization. In: Proceedings of the Companion Conference on Genetic and Evolutionary Computation. pp. 1997–2003 (2023)
43. Ungredda, J., Branke, J., Marchi, M., Montrone, T.: Single interaction multi-objective bayesian optimization. In: International Conference on Parallel Problem Solving from Nature. pp. 132–145. Springer (2022)
44. Vu, K.K., d'Ambrosio, C., Hamadi, Y., Liberti, L.: Surrogate-based methods for black-box optimization. International Transactions in Operational Research **24**(3), 393–424 (2017)
45. Wang, H., Olhofer, M., Jin, Y.: A mini-review on preference modeling and articulation in multi-objective optimization: current status and challenges. Complex & Intelligent Systems **3**(4), 233–245 (2017)
46. Wang, R., Zhou, Z., Ishibuchi, H., Liao, T., Zhang, T.: Localized weighted sum method for many-objective optimization. IEEE Transactions on Evolutionary Computation **22**(1), 3–18 (2016)
47. Wierzbicki, A.P.: A mathematical basis for satisficing decision making. Mathematical modelling **3**(5), 391–405 (1982)
48. 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)
49. Yang, K., Deutz, A., Yang, Z., Back, T., Emmerich, M.: Truncated expected hypervolume improvement: Exact computation and application. In: 2016 IEEE Congress on Evolutionary Computation (CEC). pp. 4350–4357. IEEE (2016)
50. Yang, K., Li, L., Deutz, A., Back, T., Emmerich, M.: Preference-based multi-objective optimization using truncated expected hypervolume improvement. In: 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD). pp. 276–281. IEEE (2016)
51. Zhu, S., Wu, Q., Jiang, Y., Xing, W.: A novel multi-objective group teaching optimization algorithm and its application to engineering design. Computers & Industrial Engineering **155**, 107198 (2021)
52. Zitzler, E., Brockhoff, D., Thiele, L.: The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration. In: International Conference on Evolutionary Multi-Criterion Optimization. pp. 862–876. Springer (2007)