Journal of Machine Learning Research 21 (2020) 1-42 Submitted 4/18; Revised 5/20; Published 7/20

The Kalai-Smorodinsky solution for many-objective Bayesian optimization

M. Binois * MICKAEL.BINOIS@INRIA.FR
Mathematics and Computer Science Division, Argonne National Laboratory, Lemont, USA and Université Côte d'Azur, Inria, CNRS, LJAD, Sophia Antipolis, France

V. Picheny * VICTOR@PROWLER.IO
PROWLER.io, Cambridge, UK and MIAT, Université de Toulouse, INRA, Castanet-Tolosan, France

P. Taillandier PATRICK.TAILLANDIER@INRA.FR
MIAT, Université de Toulouse, INRA, Castanet-Tolosan, France

A. Habbal HABBAL@UNICE.FR
Université Côte d'Azur, Inria, CNRS, LJAD, UMR 7351, Parc Valrose, 06108 Nice, France and Mohammed VI Polytechnic University, Benguerir, Morocco.

Editor: Bayesian Optimization Special Issue

Abstract

An ongoing aim of research in multiobjective Bayesian optimization is to extend its applicability to a large number of objectives. While coping with a limited budget of evaluations, recovering the set of optimal compromise solutions generally requires numerous observations and is less interpretable since this set tends to grow larger with the number of objectives. We thus propose to focus on a specific solution originating from game theory, the Kalai-Smorodinsky solution, which possesses attractive properties. In particular, it ensures equal marginal gains over all objectives. We further make it insensitive to a monotonic transformation of the objectives by considering the objectives in the copula space. A novel tailored algorithm is proposed to search for the solution, in the form of a Bayesian optimization algorithm: sequential sampling decisions are made based on acquisition functions that derive from an instrumental Gaussian process prior. Our approach is tested on four problems with respectively four, six, eight, and nine objectives. The method is available in the R package `GPGame` available on CRAN at `https://cran.r-project.org/package=GPGame`.

Keywords: Gaussian process, Game theory, Stepwise uncertainty reduction

1. Introduction

Bayesian optimization (BO) is recognized as a powerful tool for global optimization of expensive objective functions and now has a broad range of applications in engineering and machine learning (see, for instance, Shahriari et al., 2016). A typical example is the calibration of a complex numerical model: to ensure that the model offers an accurate representation of the system it emulates, some input parameters need to be chosen so that some model outputs match real-life data (Walter and Pronzato, 1997). Another classical

∗. Both authors contributed equally to this manuscript.

application is the optimization of performance of large machine learning systems via the tuning of its hyperparameters (Bergstra et al., 2011). In both cases, the high cost of evaluating performance drastically limits the optimization budget, and the high sample-efficiency of BO compared with alternative black-box optimization algorithms makes it highly competitive.

Many black-box problems, including those cited, involve several (or many) performance metrics that are typically conflicting, so no common minimizer to them exists. This is the field of multiobjective optimization, where one aims at minimizing simultaneously a set of p objectives with respect to a set of input variables over a bounded domain $\mathbb{X} \subset \mathbb{R}^d$:

$$\min_{\mathbf{x}\in\mathbb{X}} \left\{ y^{(1)}(\mathbf{x}), \ldots, y^{(p)}(\mathbf{x}) \right\}. \tag{1}$$

We assume that the $y^{(i)} : \mathbb{X} \to \mathbb{R}$ functions are expensive to compute, nonlinear, and potentially observed in noise. Defining that a point $\mathbf{x}^*$ dominates another point $\mathbf{x}$ if it is better in at least one of the objectives and not worse in others, the usual goal of multiobjective optimization (MOO) is to uncover the Pareto set, that is, the subset $\mathbb{X}^* \subset \mathbb{X}$ containing all the Pareto nondominated solutions:

$$\forall \mathbf{x}^* \in \mathbb{X}^*, \forall \mathbf{x} \in \mathbb{X}, \exists k \in \{1, \ldots, p\} \text{ such that } y^{(k)}(\mathbf{x}^*) \leq y^{(k)}(\mathbf{x}).$$

The image of the Pareto set in the objective space, $\mathcal{P}^* = \{y^{(1)}(\mathbb{X}^*), \ldots, y^{(p)}(\mathbb{X}^*)\}$, is called the Pareto front. Since $\mathbb{X}^*$ is in general not finite, most MOO algorithms aim at obtaining a discrete representative subset of it.

Most of the well-established Pareto-based algorithms, such as evolutionary (Deb et al., 2002; Chugh et al., 2017), descent-based (Das and Dennis, 1998), or Bayesian optimization (Wagner et al., 2010; Hernández-Lobato et al., 2016a), perform well on two or three objectives problems but poorly when $p \geq 4$. Indeed, the complexity of computing Pareto fronts dramatically increases when the number of objectives increases to four and more (Ishibuchi et al., 2008), such as the exponential increase in the number of points necessary to approximate the –possibly singular– Pareto front hyper-surface and the difficulties in its graphical representation. Moreover, one has to deal with a more intrinsic problem with many objectives, which is that almost any admissible design becomes nondominated. In this paper, our aim is precisely to address these "many-objective optimization problems" (MaO) where "many" does refer to four or more objectives (that is, as soon as the dominance relation looses its significance). Typical many-objective benchmark problems have a number of objectives that range between 3 and 10 (Ishibuchi et al., 2016) or 3 and 15 (Cheng et al., 2017), usually with far less complex (typically convex or linear) functions that the ones we consider herein (test-cases with 4, 6, 8 and 9 nonlinear and/or non convex objectives).

To circumvent these issues, Kukkonen and Lampinen (2007) advocated the use of ranks instead of nondomination and Bader and Zitzler (2011) contributions to the hypervolume. However, such algorithms require many objective evaluations and hence do not adapt well to expensive black boxes. In addition, they do not solve the problem of exploiting the resulting very large Pareto set. Some authors proposed methods to reduce the number of objectives to a manageable one (Singh et al., 2011), to use the so-called decomposition-based approaches (Asafuddoula et al., 2015), or to rely on a set of fixed and adaptive reference vectors (Chugh et al., 2018). Remarkably, while the difficulty of representing the Pareto front is highlighted, the question of selecting a particular solution on the Pareto front is mostly left to the user.

An alternative way of solving this problem is by scalarization. Among popular scalarization methods are goal programming (Charnes and Cooper, 1977), preference point methodology (Wierzbicki, 1979) -which generalizes goal programming-, and aggregation of all objective functions via weights, or by weighted metrics. This allows the use of any technique dedicated to optimization of expensive black-boxes, generally using surrogates (see, e.g., Knowles, 2006; Zhang et al., 2010; Paria et al., 2019). Nevertheless, the scalar ersatz function may become harder to model than its components, and the relation between the scalarization parameters and the corresponding Pareto optimal solution is generally not trivial. It may even be harder for hyperparameter optimization tasks (Smithson et al., 2016) that have no physical intuition backing e.g., weights in aggregative ersatz. In addition, these issues worsen with more objectives, advocating taking objectives separately.

Our present proposition amounts to searching for a single, but remarkable in some sense, solution. To do so, we adopt a game-theoretic perspective, where the selected solution arises as an equilibrium in a (non) cooperative game played by p virtual agents who own the respective p objectives (Désidéri et al., 2014). In the following, we show that the so-called Kalai-Smorodinsky (KS) solution (Kalai and Smorodinsky, 1975) is an appealing alternative. We emphasize on the fact that the following development is built on the assumption that all objectives have a priori equal importance. If any preponderance was to be set as a rule in the theoretic game framework or as a decision maker preference, then KS would be useless and ad hoc approaches should be considered, see e.g., Abou El Majd et al. (2010) for a fluid-structure application with preponderant aerodynamic criterion.

Intuitively, solutions at the "center" of the Pareto front are preferable compared with those at extremities –which is precisely what the KS solution consists of. Yet, the notion of center is arbitrary, since transforming the objectives (nonlinearly, e.g., with a log scale) would modify the Pareto front shape and affect the decision. Consider for instance the tuning of hyperparameters of neural networks with accuracies, entropies and times as objectives. Is a 0.05 improvement in entropy worth 0.05 second additional prediction time, when most designs in the domain have entropies below 0.1, or is some rescaling needed? Still, most MOO methods are sensitive to a rescaling of the objective, which is not desirable (Svenson, 2011). Our second proposition is thus to make the KS solution insensitive to monotone transformations, by operating in the copula space (Nelsen, 2006; Binois et al., 2015) with what we called the copula-KS (CKS) solution. As we show in an experiment on hyperparameter tuning, this solution provides a complementary alternative to the KS one.

Uncovering the KS and CKS solutions are nontrivial tasks for which, to our knowledge, no algorithm is available in an expensive black-box framework. Our third contribution is a novel Gaussian-process-based algorithm, building on the stepwise uncertainty reduction (SUR) paradigm (Bect et al., 2012). SUR, which is closely related to information-based approaches (Hennig and Schuler, 2012; Hernández-Lobato et al., 2016b), has proven to be efficient for solving single- and multiobjective optimization problems (Villemonteix et al., 2009; Picheny, 2013), while enjoying strong asymptotic properties (Bect et al., 2019).

The rest of the paper is organized as follows. Section 2 describes the KS solution and its extension in the copula space. Section 3 presents the Bayesian optimization algorithm developed to find KS solutions. Section 4 reports empirical performances of our approach on three challenging problems with respectively four, six, eight, and nine objectives. Section 5 summarizes our conclusions and discusses areas for future work.

2. The Kalai-Smorodinsky solution

2.1. The standard KS solution

The Kalai-Smorodinsky solution was first proposed by Kalai and Smorodinsky in 1975 as an alternative to the Nash bargaining solution in cooperative bargaining. The problem is as follows: starting from a *disagreement* or *status quo* point $\mathbf{d}$ in the objective space, the players aim at maximizing their own benefit while moving from $\mathbf{d}$ toward the Pareto front (i.e., the efficiency set). The KS solution is of egalitarian inspiration (Conley and Wilkie, 1991) and states that the selected efficient solution should yield equal benefit ratio to all the players. Indeed, given the utopia (or ideal, or shadow) point $\mathbf{u} \in \mathbb{R}^p$ defined by

$$u^{(i)} = \min_{\mathbf{x} \in \mathbb{X}^*} \; y^{(i)}(\mathbf{x}),$$

selecting any compromise solution $\mathbf{s} = [y^{(1)}(\mathbf{x}), \ldots, y^{(p)}(\mathbf{x})]$ would yield, for objective i, a benefit ratio

$$r^{(i)}(\mathbf{s}) = \frac{d^{(i)} - s^{(i)}}{d^{(i)} - u^{(i)}}.$$

Notice that the benefit from staying at $\mathbf{d}$ is zero, while it is maximal for the generically unfeasible point $\mathbf{u}$. The KS solution is the Pareto optimal choice $\mathbf{s}^* = [y^{(1)}(\mathbf{x}^*), \ldots, y^{(p)}(\mathbf{x}^*)]$ for which all the benefit ratios $r^{(i)}(\mathbf{s})$ are equal. One can easily show that $\mathbf{s}^*$ is the intersection point of the Pareto front and the line $(\mathbf{d}, \mathbf{u})$ (Figure 1, left).

We use here the extension of the KS solution to discontinuous fronts proposed in Hougaard and Tvede (2003) under the name *efficient maxmin solution*. Indeed, for discontinuous fronts the intersection with the $(\mathbf{d}, \mathbf{u})$ line might not be feasible, so there is a necessary trade-off between Pareto optimality and centrality. The efficient maxmin solution is defined as the Pareto-optimal solution that maximizes the smallest benefit ratio among players, that is:

$$\mathbf{s}^{**} \in \arg\max_{\mathbf{y} \in \mathcal{P}^*} \min_{1 \leq i \leq p} r^{(i)}(\mathbf{s}). \tag{2}$$

It is straightforward that when the intersection is feasible, then $\mathbf{s}^*$ and $\mathbf{s}^{**}$ coincide.

Figure 1 shows $\mathbf{s}^{**}$ in two situations when the feasible space is nonconvex. $\mathbf{s}^{**}$ is always on the Pareto front (hence not on the $(\mathbf{d}, \mathbf{u})$ line). In the central graph, it corresponds to the point on the front closest to the $(\mathbf{d}, \mathbf{u})$ line, which is intuitive. In the right graph, the closest point on the front (on the right hand side of the line) is actually a poor solution, as there exists another point with identical performance in y_2 but much better in terms of y_1 (on the right hand side of the line): the latter corresponds to $\mathbf{s}^{**}$.

In the following, we refer indifferently to $\mathbf{s}^*$ (if it exists) and $\mathbf{s}^{**}$ as the KS solution. Note that this definition also extends to discrete Pareto sets, which will prove useful in Section 3.

For $p \geq 3$, the KS solution, defined as the intersection point above, fulfills some of the bargaining axioms: Pareto optimality, affine invariance, and equity in benefit ratio. Moreover, for $p = 2$, KS is the *unique* solution that fulfills all the bargaining solution axioms that are Pareto optimality, symmetry, affine invariance, and restricted monotonicity (Kalai and Smorodinsky, 1975).

It is particularly attractive in a many-objective context since it scales naturally to a large number of objectives and returns a single solution, avoiding the difficulty of exploring

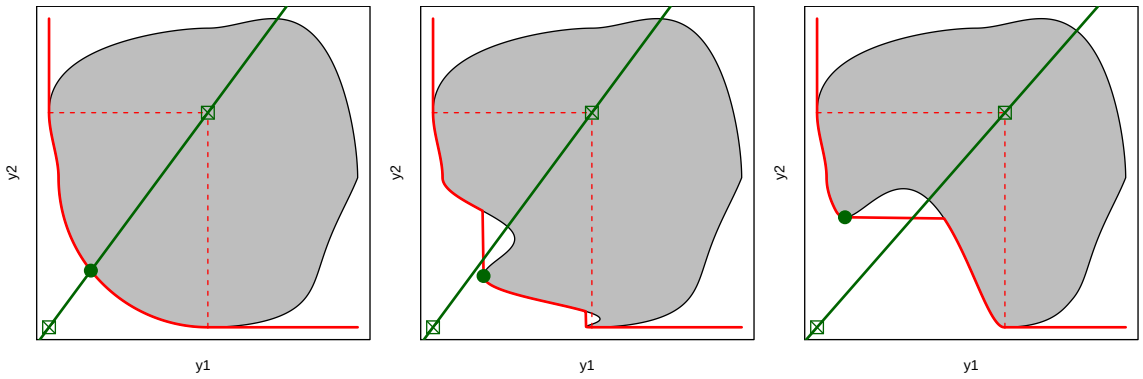

Figure 1: KS solution for a continuous (left) and discontinuous (center and right) $\mathcal{P}^*$. The KS is shown with a green disk, along with the utopia and nadir points (squares). The shaded area shows the feasible space, and the Pareto front is depicted in red.

and approximating large p-dimensional Pareto fronts–especially with a limited number of observations.

The KS solution is known to depend strongly on the choice of the disagreement point $\mathbf{d}$. A standard choice is the nadir point N given by $N_i = \max_{\mathbf{x}\in\mathbb{X}^*} \; y^{(i)}(\mathbf{x})$. Some authors introduced alternative definitions, called extended KS, to alleviate the dependence on choice of $\mathbf{d}$ (Bozbay et al., 2012), for instance by taking as disagreement point the Nash equilibrium arising from a noncooperative game, but such a choice would need a prebargaining split of the decision variable $\mathbf{x}$ among the p players.

Closely related to the Kalai and Smorodinsky solution is the reference point methodology, developed by Wierzbicki (1980). The reference point methodology uses achievement functions, which refer to aspiration and reservation references. Ideal and nadir points respectively could be taken as such reference points, and achievement functions play the same role as the benefit ratio of our present setting. Moreover, the notion of neutral compromise and max-min (of achievement functions) were introduced by Wierzbicki in the cited reference, yielding a framework objectively close to the one of Kalai and Smorodinsky. One important difference between the reference point and KS approaches is that the latter relies on a game theoretic axiomatic construction, in view of palliating the Nash equilibrium inefficiency.

2.2. Incorporating preferences or constraints

The egalitarian inspiration of the KS solution is based on the assumption that all the objectives have equal importance. However, in many practical cases, the end user may want to favor a subset of objectives which are of primary importance, while still retaining the others for optimization. One way of incorporating those *preferences* (Junker, 2004; Thiele et al., 2009; Abdolshah et al., 2019) is to discard solutions with extreme values and actually

solve the following problem:

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{X}} \quad & \left\{y^{(1)}(\mathbf{x}), \dots, y^{(p)}(\mathbf{x})\right\} \\ \text{s.t.} \quad & y^{(i)}(\mathbf{x}) \leq c_i, \qquad i \in J \subset [1, \dots, p]\,, \end{aligned} \tag{3}$$

with c_i's predefined constants. Choosing a tight value (i.e., difficult to attain) for c_i may discard a large portion of the Pareto set and favor the i-th objective over the others.

Incorporating such preferences in the KS solution can simply be done by using $\mathbf{c}$ as the disagreement point if all objectives are constrained, or by replacing the coordinates of the nadir with the c_i values. In a game theory context, this would mean that each player would state a limit of acceptance for his objective before starting the cooperative bargaining.

From geometrical considerations, one may note that $\mathbf{d}$ (and hence $\mathbf{c}$) does not need to be a feasible point. As long as $\mathbf{d}$ is dominated by the utopia point $\mathbf{u}$, the KS solution remains at the intersection of the Pareto front and the $(\mathbf{d}, \mathbf{u})$ line (optimistic $\mathbf{d}$ may simply lead to negative ratios but Eq. (2) still applies). If $\mathbf{d}$ is not dominated by $\mathbf{u}$, the $(\mathbf{d}, \mathbf{u})$ direction is no more relevant and the solution of Eq. (2) may not be Pareto-efficient. A classical way to mitigate this issue is to allow some interactivity (so that the players can reset their preferences to more realistic values). We refer to Hakanen and Knowles (2017); Tabatabaei et al. (2019); Miettinen (2012) for further ideas on interactive preference-based algorithms.

2.3. Robust KS using copulas

A drawback of KS is that it is not invariant under a monotonic (nonaffine) transformation of the objectives (this is not the case of the Pareto set, since a monotonic transformation preserves ranks, hence domination relations). In a less-cooperative framework, some players could be tempted to rely on such transformations to influence the choice of a point on the Pareto front. It may even be involuntary, for instance when comparing objectives of different natures.

To circumvent this problem, we use copula theory, which has been linked to Pareto optimality by Binois et al. (2015). In short, from a statistical point of view, the Pareto front is related to the zero level-line ∂F_0 of the multivariate cumulative density function F_Y of the objective vector $Y = (y_1(X), \dots, y_p(X))$ with X any random vector with support equal to $\mathbb{X}$. That is, $\partial F_0 = \lim_{\alpha\to 0^+} \{\mathbf{y} \in \mathbb{R}^p, F_Y(\mathbf{y}) = \mathbb{P}(Y_1 \leq y_1, \dots, Y_p \leq y_p) = \alpha\}$. Notice that solving the MaO problem by random sampling (with X uniformly distributed), as used for hyperparameter optimization by Bergstra and Bengio (2012), amounts to sample from F_Y. Extreme-level lines of F_Y actually indicate how likely it is to be close to the Pareto front (or on it if the Pareto set has a nonzero probability mass).

A fundamental tool for studying multivariate random variables is the copula. A copula C_Y is a function linking a multivariate cumulative distribution function (CDF) to its univariate counterparts, such that for $\mathbf{y} \in \mathbb{R}^p$, $F_Y(\mathbf{y}) = C_Y(F_1(y_1), \dots, F_p(y_p))$ with $F_i = \mathbb{P}(Y_i \leq y_i)$, $1 \leq i \leq p$ (Nelsen, 2006). When F_Y is continuous, C_Y is unique, from Sklar's theorem (Nelsen, 2006, Theorem 2.3.3). Consequently, as shown by Binois et al. (2015), learning the marginal CDFs F_i's and extreme levels of the copula C_Y, that is $\partial C_0 = \lim_{\alpha\to 0^+} \left\{\mathbf{u} \in [0,1]^p, C_Y(\mathbf{u}) = F_Y(F_1^{-1}(u_1), \dots, F_p^{-1}(u_p)) = \alpha\right\}$, amounts to learning ∂F_0, hence the Pareto front.

A remarkable property of copulas is their invariance under monotone increasing transformations of univariate CDFs (Nelsen, 2006, Theorem 2.4.3). Extending their proof to the p-dimensional case, suppose that $g_1, \dots, g_p$ are strictly increasing transformations on the ranges of $Y_1, \dots, Y_p$, respectively; hence they are invertible. Denote $G_1, \dots, G_p$ the marginal distribution functions of $\mathbf{g}(Y)$. It then holds that $G_i(y_i) = \mathbb{P}(g_i(Y_i) \leq y_i) = \mathbb{P}(Y_i \leq g_i^{-1}(y_i)) = F_i(g_i^{-1}(y_i))$. Then $C_{\mathbf{g}(Y)}(G_1(y_1), \dots, G_p(y_p)) = \mathbb{P}(g_1(Y_1) \leq y_1, \dots, g_p(Y_p) < y_p) = \mathbb{P}(Y_1 \leq g_1^{-1}(y_1), \dots, Y_p \leq g_p^{-1}(y_p)) = C_Y(F_1(g_1^{-1}(y_1)), \dots, F_p(g_p^{-1}(y_p))) = C_Y(G_1(y_1), \dots, G_p(y_p))$.

Now, our proposition is to consider the KS solution in the copula space, that is, taking $F_1, \dots, F_p$ as objectives instead of $y_1, \dots, y_p$. This "copula-KS solution" (henceforth CKS) is Pareto-efficient and invariant to any monotonic transformation of the objectives. In addition, in the copula space the utopia point is always $(0, \dots, 0)$. While the nadir remains unknown, the point $(1, \dots, 1)$ may serve as an alternative disagreement point, since it corresponds to the worst solution for each objective. This removes the difficult task of learning the $(\mathbf{d}, \mathbf{u})$ line, at the expense of learning the marginal distribution and the copula function. With this choice, one may remark that when $\mathbb{X}$ is finite, our proposed solution is very intuitive, as it amounts to choosing the maxmin solution over ranks, as:

$$r_C^{(i)}(\mathbf{s}) = \frac{1}{\mathrm{Card}(\mathbb{X})} \sum_{j=1}^{\mathrm{Card}(\mathbb{X})} \delta\left(s^{(i)} \leq y^{(i)}(\mathbf{x}_j)\right), \tag{4}$$

with $\delta()$ the Kronecker delta function.

Importantly, CKS now depends on the instrumental law of X. For instance, this is the probability distribution that would be used to solve the problem via random sampling, such as a uniform distribution for a bounded domain and a normal one for unbounded domain. Without loss of generality, we can always assume that X is uniformly distributed over $\mathbb{X}$, and using a different distribution with the same support is equivalent to introducing a transformation over X. Hence, contrary to KS, CKS is sensitive to the input definition (e.g., the MOO problems $\min(y^{(1)}(x), y^{(2)}(x))$ and $\min(y^{(1)}(x^2), y^{(2)}(x^2))$ with $x \in [0, 1]$, with equal Pareto fronts, would lead to different CKS solutions). Despite this, in a MaO context, CKS remains much more robust to reformulations than KS, in the sense that transformations in the input space will affect all of the objectives and not a single one. This makes it much more difficult to bias the solution in favor of, say, a particular objective: it can only be done by increasing the mass of the distribution of X around the optimum of this objective, which is presumably unknown for expensive black-box objectives.

In the following, we only consider the CKS computed using the uniform distribution over $\mathbb{X}$. By doing so, we assume that the problem formulation is appropriate, in the sense that the outputs are roughly stationary with respect to the inputs (as opposed to an ill-defined problem, that would contain large plateaus for instance).

2.4. Illustration

Let us consider a classical two-variable, biobjective problem from the multiobjective literature (P1, see Parr, 2013). We first compute the two objectives on $5,000$ uniformly sampled points in $\mathbb{X} = [0, 1]^2$, out of which the feasible space, Pareto front, and KS solution are extracted (Figure 2, top left). The KS solution has a visibly central position in the Pareto front, which makes it a well-balanced compromise.

Applying a log transformation of the first objective does not change the Pareto set but modifies here substantially the shape of the Pareto front (from convex to concave) and the KS solution, leading to a different compromise (Figure 2, top center), which compared to the previous one favors Y_1 over Y_2.

Both original and rescaled problems share the same image in the copula space (Figure 2, top right), which provides a third compromise. Seen from the original space, the CKS solution seems here to favor the first objective: this is due to the high density of points close to the minimum on Y_1 (Figure 2, top left, top histogram). It is, however, almost equivalent to the KS solution under a log transformation of Y_1. From a game perspective, the two players agree on a solution with equal ranks: here roughly the $1,200$th best ($F_1 \approx F_2 \approx 0.24$) out of $5,000$, independently of the gains in terms of objective values.

Now, we consider the same problem, but the $5,000$ points are sampled in $\mathbb{X}$ using a triangular distribution for each marginal (between 0 and 1, asymmetric with a mode at $x = 0.25$). Using a distribution does not change the KS solution, but modifies the distributions of the objectives (Figure 2, bottom left, both histograms). The image in the copula space (Figure 2, bottom center) is thus different, which leads to a slightly different CKS solution.

Finally, we compare CKS solutions obtained by using independent beta distributions on the inputs, with parameters randomly chosen between 0.5 and 4 (Figure 2, bottom right). Despite being based on extremely different distributions, the CKS solutions are relatively similar.

3. A Bayesian optimization algorithm to find KS solutions

Computing the KS or CKS solutions are challenging problems. It requires for KS learning the ideal and disagreement points $\mathbf{u}$ and $\mathbf{d}$ (which are very difficult problems on their own; see, for instance, Bechikh et al., 2010) and for CKS the marginals and copula, as well as (for both) the part of the Pareto front that intersects the $(\mathbf{d}, \mathbf{u})$ line. An additional difficulty arises when the objective functions cannot be computed exactly but only through a noisy process.

In this section, we consider that one has access to observations of the form

$$\mathbf{f}_i = \mathbf{y}(\mathbf{x}_i) + \boldsymbol{\varepsilon}_i, \tag{5}$$

where $\boldsymbol{\varepsilon}_i$ is a zero-mean noise vector with independent components of variances $(\tau_i^{(1)}, \ldots, \tau_i^{(p)})$. Our approach readily adapts to the deterministic case by setting $\forall j : \tau_i^{(j)} = 0$. We assume here that all objectives are collected at the same time (using a single black-box), hence sharing the experimental points $\mathbf{x}_i$. The case of several black-boxes, as presented by Hernández-Lobato et al. (2016b), is not considered here and is deferred to future work.

3.1. Elements of Bayesian optimization

For our algorithm, we consider a classical BO framework, where independent Gaussian process (GP) priors are put on the objectives:

$$\forall i \in 1, \ldots, p, \quad Y^{(i)}(.) \sim \mathcal{GP}\left(\mu^{(i)}(.), \sigma^{(i)}(.,.)\right), \tag{6}$$

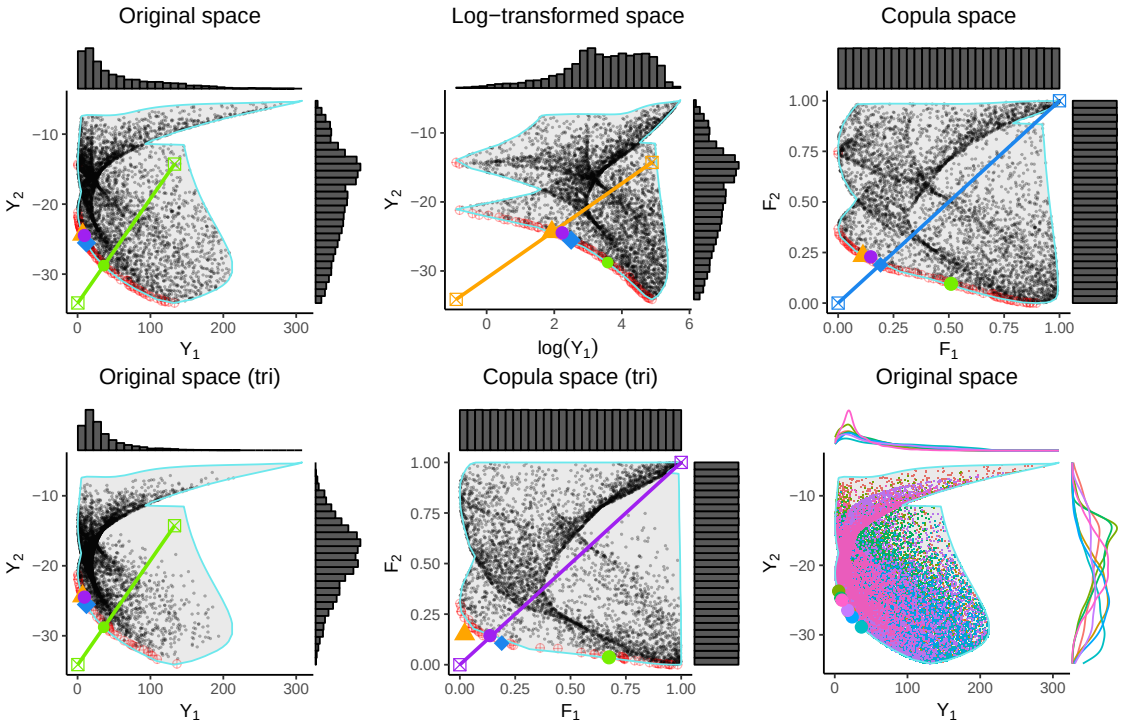

Figure 2: Top: KS (green disk), KS in log-scale (orange triangle) and CKS (blue diamond) solutions for a biobjective problem, based on $5,000$ uniformly sampled designs in $\mathbb{X} = [0,1]^2$, shown in the objective space. The black dots show all the dominated values of the grid and the red crossed circles the Pareto-optimal ones. The shaded area shows the feasible space. Marginal objective densities are reported on the corresponding axes. The $(\mathbf{d}, \mathbf{u})$ lines are shown with matching colors. Bottom left: same problem, but with $5,000$ samples in $\mathbb{X}$ from a triangular distribution. The corresponding CKS solution (bottom, center) is shown in purple. Bottom right: CKS solutions corresponding to 8 different input distributions.

where the mean $\mu^{(i)}$ and covariance $\sigma^{(i)}$ have predetermined parametric forms whose parameters are estimated by maximum likelihood (Rasmussen and Williams, 2006). Conditioning on a set of observations $\{\mathbf{f}_1, \dots, \mathbf{f}_n\}$, GPs provide flexible response fits associated with uncertainty estimates. They enable operating sequential design decisions via an *acquisition function* $J(\mathbf{x})$, which balances between exploration and exploitation in seeking global optima. Hence, the design consists of a first set of n_0 observations generated by using a space-filling design, generally from a variant of Latin hypercube design (LHD, McKay et al., 1979) to obtain a first predictive distribution of $Y^{(i)}(.)$, and a second set of sequential observations chosen as

$$\mathbf{x}_{n+1} \in \arg\max_{\mathbf{x}\in\mathbb{X}} J(\mathbf{x}), \quad (n \geq n_0). \tag{7}$$

GP equations are deferred to Appendix A. In the following, we use the subscript n to denote quantities conditional on the set of n observations (e.g., $Y_n^{(i)}$, $\mu_n^{(i)}$ or $\sigma_n^{(i)}$).

3.2. Baseline algorithms

We propose first a simple algorithm for KS, that alternatively searches for the utopia point, the nadir, the KS solution, and so on until the budget is met.

Looking for the utopia point is the simplest task, as it amounts to looking for the individual minima of the objective functions. We can use p times an acquisition function designed for single objective problems, e.g., expected improvement, EI, Jones et al. (1998) in the noiseless setting, or one of its "noisy" counterpart, see Picheny et al. (2013) for a review.

The nadir point is constructed from worst objective function values of Pareto optimal solutions. For $p = 2$, looking for the utopia directly provides the nadir. However, when $p \geq 3$, it does not coincide with individual objective extrema. Accurately estimating the nadir is a difficult task, especially when the number of objective functions increases, for which only a handful of algorithms are available (Deb et al., 2006; Bechikh et al., 2010). Noting that each coordinate is then the solution of a single objective constrained problem (maximum under constraint that $\mathbf{x}$ belongs to the Pareto front), we propose to use the constrained version of EI (Expected Feasible Improvement, EFI, Schonlau et al., 1998). Our acquisition function is the expected improvement of minus the objective ($\mathrm{EI}_{\mathrm{nad}}$, to point towards maximal values instead of minimal ones) multiplied by the probability of being nondominated by the observation set (p_{ND}, which is available in closed form from the GP marginal (Couckuyt et al., 2014)). See Appendix A for expressions of EI, $\mathrm{EI}_{\mathrm{nad}}$ and p_{ND}.

Finally, given an $\mathbf{u}$ and $\mathbf{d}$, the KS solution can be sought by using a GP-UCB (Srinivas et al., 2012) analog for Eq. (2), that is, replacing the $r^{(i)}(\mathbf{s})$ values by a lower confidence bound provided by the GP model, which is simply the GP posterior mean minus a positive constant β_n (that grows with the logarithm of n) times the GP posterior standard deviation:

$$r_{LCB}^{(i)}(y(\mathbf{x})) = \frac{d^{(i)} - \mu_n^{(i)}(\mathbf{x}) + \beta_n\sqrt{\sigma_n^{(i)}(\mathbf{x},\mathbf{x})}}{d^{(i)} - u^{(i)}}, \quad \beta_n \geq 0.$$

In total, for p objectives we have $2p+1$ different acquisition functions. Some corresponding tasks may be more important than others, but it is difficult to prioritize them in a principled way. Hence, as a baseline algorithm, we propose to optimize and sample the corresponding maximizer of each acquisition function sequentially (see Algorithm 1).

Algorithm 1 Pseudo-code for the baseline algorithm for KS

Require: p GP models trained on $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, $\{\mathbf{f}_1, \ldots, \mathbf{f}_n\}$
while $n \leq N$ **do**
  for $i \leftarrow 1$ to p **do**
    Choose $\mathbf{x}_{n+1} = \arg\max_{\mathbf{s}\in\mathbb{X}} EI^{(i)}(\mathbf{x})$
    Compute $\mathbf{f}_{n+1}$ using the expensive black-box
    Update the GP models by conditioning on $\{\mathbf{x}_{n+1}, \mathbf{f}_{n+1}\}$.
    $n \leftarrow n + 1$
  end for
  for $i \leftarrow 1$ to p **do**
    Choose $\mathbf{x}_{n+1} = \arg\max_{\mathbf{x}\in\mathbb{X}} {EI_{\text{nad}}}^{(i)}(\mathbf{x}) \times p_{\text{ND}}(\mathbf{x})$
    Compute $\mathbf{f}_{n+1}$ using the expensive black-box
    Update the GP models by conditioning on $\{\mathbf{x}_{n+1}, \mathbf{f}_{n+1}\}$.
    $n \leftarrow n + 1$
  end for
  Choose $\mathbf{x}_{n+1}$ solution of Eq. (2) with $r_{LCB}^{(i)}$.
  Compute $\mathbf{f}_{n+1}$ using the expensive black-box
  Update the GP models by conditioning on $\{\mathbf{x}_{n+1}, \mathbf{f}_{n+1}\}$.
  $n \leftarrow n + 1$
end while

Finding the CKS solution requires learning the copula and marginals, which does not easily convert into acquisition functions. However, both quantities correspond to global features of the objectives. As a baseline, we propose to sample where the GP models are most uncertain, that is, using as acquisition functions the posterior variances $\sigma_n^{(i)}$. Using the same structure as Algorithm 1), we propose $2p$ maximizations of the $\sigma_n^{(i)}$'s, followed by an exploitation step. Instead of using the LCB, which is not available in closed form for CKS, we propose here to sample at the CKS solution of the GP posterior mean.

3.3. Stepwise uncertainty reduction

The drawback of the presented baseline algorithms are that they do not balance the different learning tasks in a principled way, which may result in very subefficient sampling as the number of objectives increases. Instead, we propose here to design a single acquisition $J(\mathbf{x})$ tailored to our problem. To do so, we follow a step-wise uncertainty reduction (SUR) approach, similar to the one proposed by Picheny et al. (2019) to solve Nash equilibria problems.

Let us denote by $\Psi : \mathbb{Y} \to \mathbb{R}^p$ the mapping that associates a KS or CKS solution with any multivariate function $\mathbf{y}$ defined over $\mathbb{X}$. Given a distribution $\mathbf{Y}_n()$, $\Psi(\mathbf{Y}_n)$ is a random vector (of unknown distribution). The first step of designing a SUR strategy is to define an uncertainty measure $\Gamma(\mathbf{Y}_n)$ that reflects our lack of knowledge of our problem solution (2). A simple measure of variability of a vector is the determinant of its covariance matrix (Fedorov, 1972):

$$\Gamma(\mathbf{Y}_n) = \det\left[\text{cov}\left(\Psi(\mathbf{Y}_n)\right)\right]. \tag{8}$$

Intuitively, $\Gamma(\mathbf{Y}_n)$ tends to zero when all the components of $\Psi(\mathbf{Y}_n)$ become known accurately.

The SUR strategy greedily chooses the next observation that reduces the most this uncertainty:

$$\max_{\mathbf{x}\in\mathbb{X}} \Gamma(\mathbf{Y}_n) - \Gamma(\mathbf{Y}_{n,\mathbf{x}}),$$

where $\mathbf{Y}_{n,\mathbf{x}}$ is the GP conditioned on $\{\mathbf{y}(\mathbf{x}_1), \dots, \mathbf{y}(\mathbf{x}_n), \mathbf{y}(\mathbf{x})\}$. However, such an ideal strategy is not tractable as it would require evaluating all $\mathbf{y}(\mathbf{x})$ while maximizing over $\mathbb{X}$.

For a tractable strategy, we consider the expected uncertainty reduction:

$$\Gamma(\mathbf{Y}_n) - \mathbb{E}_{\mathbf{Y}_n(\mathbf{x})}\left[\Gamma(\mathbf{Y}_{n,\mathbf{x}})\right], \tag{9}$$

where $\mathbb{E}_{\mathbf{Y}_n(\mathbf{x})}$ denotes the expectation taken over $\mathbf{Y}_n(\mathbf{x})$. Removing the constant term $\Gamma(\mathbf{Y}_n)$, our policy is defined with

$$\mathbf{x}_{n+1} \in \arg\min_{\mathbf{x}\in\mathbb{X}} \left\{ J(\mathbf{x}) = \mathbb{E}_{\mathbf{Y}_n(\mathbf{x})}\left[\Gamma(\mathbf{Y}_{n,\mathbf{x}})\right] \right\}. \tag{10}$$

Note that $J(\mathbf{x})$ defines a trade-off between exploration and exploitation, as well as a trade-off between the different learning tasks ($\mathbf{d}$ and $\mathbf{u}$ points, copula, Pareto front).

3.4. Computational aspects

Given a distribution $\mathbf{Y}_n()$, solving exactly Eq. (10) is impossible for two reasons. First, the mapping Ψ can be very complex and there are no algorithms to approach it accurately. Second, the nonlinearity of Γ prevent us from getting a closed-form expression for $J(\mathbf{x})$.

We propose to solve both problems by replacing the (continuous) design space $\mathbb{X}$ by a discrete representation $\mathbb{X}^\star$ of size N, and evaluate our acquisition function by Monte-Carlo. The two key-points for our approach to work in practice are, first, that the cost of the Monte-Carlo evaluation can be substantially alleviated by specific pre-computations and update formulae: this is exposed in Section 3.4.1. Second, the Monte-Carlo approach has a cubic cost which only allows the use of small discrete sets, but carefully choosing those sets allows us to estimate our acquisition function with precision: this is proposed in Section 3.4.2.

3.4.1. Computing and optimizing the SUR criterion

Let $\mathbb{X}^\star = \{\mathbf{x}_1^\star, \dots, \mathbf{x}_N^\star\} \subset \mathbb{X}$. Since $\mathbb{X}^\star$ is discrete, one can easily generate independent drawings of $\mathbf{Y}_n(\mathbb{X}^\star)$: $\boldsymbol{\mathcal{Y}}_1^\star, \dots, \boldsymbol{\mathcal{Y}}_M^\star$. For each sample, the corresponding KS solution $\Psi(\boldsymbol{\mathcal{Y}}_i^\star)$ can be computed by exhaustive search. The following empirical estimator of $\Gamma(\mathbf{Y}_n)$ is then available:

$$\hat{\Gamma}\left(\boldsymbol{\mathcal{Y}}_1^\star, \dots, \boldsymbol{\mathcal{Y}}_M^\star\right) = \det\left[\mathbf{Q}_{\mathcal{Y}}\right],$$

with $\mathbf{Q}_{\mathcal{Y}}$ the sample covariance of $\Psi(\boldsymbol{\mathcal{Y}}_1^\star), \dots, \Psi(\boldsymbol{\mathcal{Y}}_M^\star)$.

Now, let $\mathbf{x}$ be a candidate observation point, and $\boldsymbol{\mathcal{F}}_1, \dots, \boldsymbol{\mathcal{F}}_K$ be independent drawings of $\mathbf{Y}_n(\mathbf{x})$. Denoting $\boldsymbol{\mathcal{Y}}_1^\star|\boldsymbol{\mathcal{F}}_i, \dots, \boldsymbol{\mathcal{Y}}_M^\star|\boldsymbol{\mathcal{F}}_i$ independent drawings of $\mathbf{Y}_{n,\mathbf{x}}(\mathbb{X}^\star) = \mathbf{Y}_n(\mathbb{X}^\star)|\mathbf{Y}_n(\mathbf{x}) = \boldsymbol{\mathcal{F}}_i$, an estimator of $J(\mathbf{x})$ is then obtained by using the empirical mean:

$$\hat{J}(\mathbf{x}) = \frac{1}{K}\sum_{i=1}^{K} \hat{\Gamma}\left(\boldsymbol{\mathcal{Y}}_1^\star|\boldsymbol{\mathcal{F}}_i, \dots, \boldsymbol{\mathcal{Y}}_M^\star|\boldsymbol{\mathcal{F}}_i\right).$$

Hence, computing $\hat{J}(\mathbf{x})$ requires $M \times K$ samples $\boldsymbol{\mathcal{Y}}^\star_j|\boldsymbol{\mathcal{F}}_i$. This can simply be done by nesting two Monte-Carlo loops, but would be very intensive computationally. To overcome this problem, we employ here the *fast update of conditional simulation ensemble* algorithm proposed by Chevalier et al. (2015), as detailed below and illustrated in Figure 3. As shown by Chevalier et al. (2015), samples $\boldsymbol{\mathcal{Y}}^\star|\boldsymbol{\mathcal{F}}$ of $\mathbf{Y}_{n,\mathbf{x}}(\mathbb{X}^\star)$ (conditioned on $n+1$ observations) can be obtained efficiently from a joint sample $\{\boldsymbol{\mathcal{Y}}^\star, \boldsymbol{\mathcal{F}}\}$ of $\mathbf{Y}_n(\{\mathbb{X}^\star, \mathbf{x}\})$ (conditioned only on n observations) by working on residuals using, with $1 \leq i \leq p$:

$$\mathcal{Y}^{\star(i)}|\mathcal{F}^{(i)} \quad = \quad \mathcal{Y}^{\star(i)} + \boldsymbol{\lambda}^{(i)}(\mathbf{x}) \left(\mathcal{F}^{(i)} - \mathcal{Y}^{(i)}(\mathbf{x}) \right), \tag{11}$$

with $\boldsymbol{\mathcal{Y}}^\star = [\mathcal{Y}^{\star(1)}, \ldots, \mathcal{Y}^{\star(p)}] \in \mathbb{R}^{N \times p}$, $\boldsymbol{\mathcal{Y}}(\mathbf{x}) = [\mathcal{Y}^{(1)}(\mathbf{x}), \ldots, \mathcal{Y}^{(p)}(\mathbf{x})] \in \mathbb{R}^p$,
$\boldsymbol{\mathcal{F}} = [\mathcal{F}^{\star(1)}, \ldots, \mathcal{F}^{\star(p)}] \in \mathbb{R}^p$ and

$$\boldsymbol{\lambda}^{(i)}(\mathbf{x}) = \frac{1}{\boldsymbol{\sigma}_n^{(i)}(\mathbf{x}, \mathbf{x})} \left[\boldsymbol{\sigma}_n^{(i)}(\mathbf{x}^\star_1, \mathbf{x}), \ldots, \boldsymbol{\sigma}_n^{(i)}(\mathbf{x}^\star_N, \mathbf{x}) \right].$$

Hence, the two expensive operations are computing $\boldsymbol{\lambda}^{(i)}(\mathbf{x})$ and drawing samples of $\mathbf{Y}_n(\{\mathbb{X}^\star, \mathbf{x}\})$, the later having an $\mathcal{O}((N+1)^3)$ complexity when using the standard decomposition procedure based on Cholesky, (see, e.g., Diggle and Ribeiro, 2007).

A first decisive advantage of the update approach is that computing $\boldsymbol{\lambda}^{(i)}(\mathbf{x})$ needs to be done only once for each $\mathbf{x}$, regardless of the values taken by $\mathcal{Y}^{\star(i)}$ and $\mathcal{F}^{(i)}$. This considerably eases the nested Monte-Carlo loop.

Moreover, if we restrict $\mathbf{x}$ to belong to $\mathbb{X}^\star$, drawing $\{\boldsymbol{\mathcal{Y}}^\star, \boldsymbol{\mathcal{F}}\}$ reduces to drawing $\boldsymbol{\mathcal{Y}}^\star$ and needs to be done only once, independently of $\mathbf{x}$, hence prior to minimizing the acquisition function. Hence, although optimizing $\hat{J}(\mathbf{x})$ over the continuous space $\mathbb{X}$ is definitely feasible, it requires completing the simulation over $\mathbb{X}^\star$ with $\mathbf{x}$, incurring an additional computational cost. We thus restrict $\mathbf{x}$ to $\mathbb{X}^\star$, and solve Eq. (10) by exhaustive search.

3.4.2. Choosing integration points $\mathbb{X}^\star$

Due to the cubic complexity, generating $\boldsymbol{\mathcal{Y}}^\star_1, \ldots, \boldsymbol{\mathcal{Y}}^\star_M$ limits $\mathbb{X}^\star$ in practice to at most a couple thousand points. In small dimension (say, $1 \leq d \leq 3$), $\mathbb{X}^\star$ may consist simply of a Cartesian grid or a dense space-filling design (Niederreiter, 1988). For larger d, such approaches do not scale and $\mathbb{X}^\star$ may not be large enough to cover $\mathbb{X}$ sufficiently. However, accurate approximations of $J(\mathbf{x})$ can still be obtained, as long as $\mathbb{X}^\star$ covers the influential parts of $\mathbb{X}$ with respect to J, that is, regions where (a) the KS or CKS solution is likely to lie, (b) (for KS) minimal values of $\mathbf{Y}$ are likely (to estimate the utopia), (c) (for KS, without preferences) maximal non-dominated values of $\mathbf{Y}$ are likely (to estimate the nadir).

To design a set $\mathbb{X}^\star$ of limited size that contains those three components, we proceed as follows. First, a large space-filling set $\mathbb{X}_{\text{large}}$ is generated, and the marginal posterior distribution is computed for each element of $\mathbb{X}_{\text{large}}$. Since we do not consider the joint distribution, this has a cost of only $\mathcal{O}(N^2_{\text{large}})$. Then, we use this marginal distribution to pick a small subset of $\mathbb{X}_{\text{large}}$ that is likely to contain the components a, b, and c. For robustness, the choice of the subset corresponds to an exploration (large $\mathbf{Y}_n$ variance) / exploitation (following the $\mathbf{Y}_n$ mean) trade-off, as we show below.

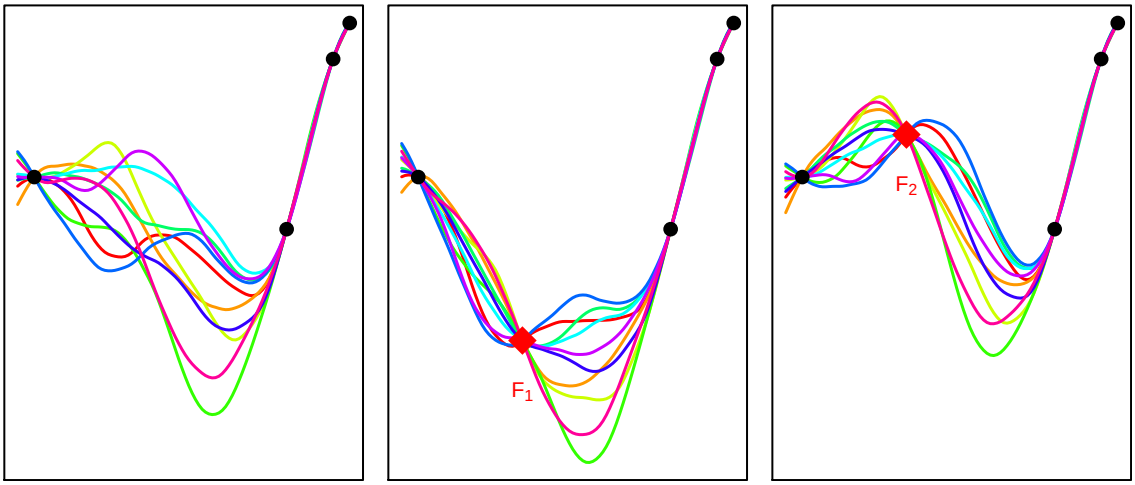

Figure 3: Left: initial drawings $\boldsymbol{\mathcal{Y}}_1^\star, \ldots, \boldsymbol{\mathcal{Y}}_M^\star$. Middle and right: same drawings, but conditioned further on an observation $\mathcal{F}^1$ (middle) and $\mathcal{F}^2$ (right).

Central part For the first BO iteration, we obtain a rough estimate of the solution by computing the KS or CKS solution of the posterior GP mean. As the mean is likely to be smoother than the actual function, a better solution, available as soon as one iteration has been performed, is to use the set of simulated KS/CKS solutions generated during the previous computation of J: $\Psi(\boldsymbol{\mathcal{Y}}_1^\star), \ldots, \Psi(\boldsymbol{\mathcal{Y}}_M^\star)$. Then, using the marginal distribution, we compute the probability p_{box} for each element of $\mathbb{X}_{\text{large}}$ to belong to a box defined by the extremal values of the simulated KS (see Appendix A for formulas). A first set $\mathbb{X}_{\text{central}}$ is obtained by sampling from $\mathbb{X}_{\text{large}}$ randomly with probabilities proportional to p_{box}.

Utopia To include points with an exploration / exploitation trade-off, we compute for each objective the expected improvement for all elements in $\mathbb{X}_{\text{large}}$. The p EI maximizers constitute a second set $\mathbb{X}_{\text{utopia}}$.

Nadir Following the baseline strategy (Section 3.2), to capture the nadir we compute for each objective the expected improvement of minus the objective multiplied by the probability of non-domination. The p maximizers of these criteria constitute a third set $\mathbb{X}_{\text{nadir}}$.

Finally, the reduced set $\mathbb{X}^\star$ is taken as $\mathbb{X}_{\text{central}} \cup \mathbb{X}_{\text{utopia}} \cup \mathbb{X}_{\text{nadir}}$ for KS, $\mathbb{X}_{\text{central}} \cup \mathbb{X}_{\text{utopia}}$ for KS with a predefined disagreement point and $\mathbb{X}_{\text{central}}$ for CKS.

$\mathbb{X}^\star$ may be renewed every time a new observation is added to the GP model, firstly to account for the changes in the GP distribution, but also to improve robustness (a critical region can be missed at an iteration but accounted for during the next).

Algorithm 2 provides the pseudo-code of the SUR loop for KS, that replaces the standard acquisition function maximisation of Eq. (7) in the classical BO structure. The CKS algorithm is almost identical, except that $\mathbb{X}^\star = \mathbb{X}_{\text{central}}$, and the copula requires a specific treatment, as we describe next.

Algorithm 2 Pseudo-code for the SUR loop for KS

Require: p GP models trained on $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, $\{\mathbf{f}_1, \ldots, \mathbf{f}_n\}$
Require: $\mathbb{X}_{\text{large}}$, $N_\star$, M
Compute the marginal distribution of $\mathbf{Y}(\mathbb{X}_{\text{large}})$
Using the marginal compute: p_{box}, EI, EI_{nad}, p_{ND}
Choose:

- $\mathbb{X}_{\text{central}} \subset \mathbb{X}_{\text{large}}$ with p_{box}, size $N_\star - 2p$
- $\mathbb{X}_{\text{utopia}} \subset \mathbb{X}_{\text{large}}$ with EI
- $\mathbb{X}_{\text{nadir}} \subset \mathbb{X}_{\text{large}}$ with $EI_{\text{nad}} \times p_{\text{ND}}$

Define $\mathbb{X}^\star = \mathbb{X}_{\text{central}} \cup \mathbb{X}_{\text{utopia}} \cup \mathbb{X}_{\text{nadir}}$
Generate M draws of $\mathbf{Y}(\mathbb{X}^\star)$: $(\boldsymbol{\mathcal{Y}}_1^\star, \ldots, \boldsymbol{\mathcal{Y}}_M^\star)$
for $i \leftarrow 1$ to $N_\star$ **do**
    From $(\boldsymbol{\mathcal{Y}}_1^\star, \ldots, \boldsymbol{\mathcal{Y}}_M^\star)$, extract M draws of $\mathbf{Y}(\mathbf{x}_\star^i)$: $\mathcal{F}_1, \ldots, \mathcal{F}_M$
    for $k \leftarrow 1$ to M **do**
        Compute $(\boldsymbol{\mathcal{Y}}_1 | \mathcal{F}^k, \ldots, \boldsymbol{\mathcal{Y}}_M | \mathcal{F}^k)$ using Eq. (11)
        Compute $\Gamma_{ik} = \hat{\Gamma}\left(\boldsymbol{\mathcal{Y}}_1 | \mathcal{F}^k, \ldots, \boldsymbol{\mathcal{Y}}_M | \mathcal{F}^k\right)$
    end for
    Compute $\hat{J}_i = \frac{1}{M} \sum_{k=1}^{M} \Gamma_{ik}$
end for
Choose $\mathbf{x}_{n+1} = \mathbf{x}_I^\star$ with $I = \arg\min_{i \in [1, N_\star]} \hat{J}_i$
Compute $\mathbf{f}_{n+1}$ using the expensive black-box
Update the GP models by conditioning on $\{\mathbf{x}_{n+1}, \mathbf{f}_{n+1}\}$.

3.4.3. Copula estimation

In the continuous case, unless additional information is available about the marginal distributions and copula function, empirical estimators may be used; see, for instance, Omelka et al. (2009). Note that as we are interested in CKS, only a good approximation of the diagonal of the copula is critical, which makes it an easier problem than regular copula inference. Additionally, the CKS may not be in the tails of the marginal distributions, which also eases the inference.

Nonetheless, empirical estimates are based on a sample of $\mathbf{Y}(X)$, with X i.i.d. according to the instrumental distribution (i.e., uniform). Hence, both the observation set and $\mathbb{X}^\star$ are inappropriate, as they are too small sets, and more importantly, not uniformly distributed. Instead, the copula and marginals are estimated on a large auxiliary i.i.d. sample $\{\mathbb{X}_{\text{aux}}, \mathbf{Y}_{\text{aux}}\}$. One may choose $\mathbb{X}_{\text{aux}} = \mathbb{X}_{\text{large}}$. Since jointly sampling $\mathbf{Y}(\mathbb{X}_{\text{aux}})$ is out of reach for a large sample (Section 3.4.2), we follow Oakley (2004) and use the conditional simulations $\boldsymbol{\mathcal{Y}}_j$ on $\mathbb{X}^\star$ as pseudo-observations to update the GP predictive mean, and we take our sample $\mathbf{Y}_{\text{aux}}$ for this updated mean $\mathbb{E}\left[Y(\mathbb{X}_{\text{aux}}) | \boldsymbol{\mathcal{Y}}_j\right]$ (this approach has also been referred to as *hallucinated observations*, Desautels et al., 2014).

With that setting, in practice there is no need to explicitly estimate the copula and marginals. The CKS is simply given by the ranks with respect to the auxiliary set, i.e.,

$$r_C^{(i)}(\mathbf{s}) = \frac{1}{N_{\text{aux}}} \sum_{j=1}^{N_{\text{aux}}} \delta \left(s^{(i)} \leq y_{\text{aux}}^{(i)}(\mathbf{x}_j) \right),$$

with $N_{\text{aux}} = \text{Card}(\mathbb{X}_{\text{aux}})$.

4. Experiments

This section details numerical experiments on four test problems: two toy problems from the multiobjective and BO literature, a problem of hyperparameter tuning, and the calibration of a complex simulator. All experiments were conducted in `R` (R Core Team, 2018), by using the dedicated package `GPGame`; see the work of Picheny and Binois (2018) for details.

4.1. Synthetic problems

As a proof of concept, we consider the DTLZ2 function (Deb et al., 2002), with five variables and four objectives, that has a concave dome-shaped Pareto front, and a six variables and six objectives problem obtained by rotating and rescaling six times the classical mono-objective function hartman (Dixon and Szegö, 1978) (see Appendix D for the problem formulation).

The objectives of these experiments are 1) to compare the performance of SUR with baseline strategies, and 2) to evaluate the effect of discretizations. On both cases, we consider two finite domains $\mathbb{X}$ with respectively 10^5 and 10^7 elements uniformly distributed in $[0, 1]^d$. This allows the computation of the exact KS and CKS solutions.

For the smaller domain, we set $\mathbb{X}_{\text{large}} = \mathbb{X}$, while for the larger one we take $\mathbb{X}_{\text{large}}$ as a random subset of size 10^5 of $\mathbb{X}$ (renewed every iteration), in order to emulate the loss incurred by SUR for working with discrete representations of continuous spaces.

For SUR, we test two configurations corresponding respectively to a small or large computational effort to estimate the acquisition function. The parameters are reported in Table 4.1, along with the computational times required for a single SUR iteration on ten CPUs @2.3Ghz.

As competitors, we use the baseline algorithms presented in Section 3.2 and a purely exploratory algorithm based on uniform sampling on $\mathbb{X}$. In addition, we ran SMS-EGO (Ponweiser et al., 2008), a multi-objective Bayesian optimization algorithm based on hypervolume improvement that has been shown to compare favorably to other indicator-based algorithms (Wagner et al., 2010; Binois and Picheny, 2019), in particular on the DTLZ2 problem.

Table 1: Configurations and wall clock times (for a single iteration on hartman) of the SUR algorithm. CKS relies on 10^4 auxiliary points.

Configuration	N	M	clock time (KS)	clock time (CKS)
Coarse	250	25	33s	114s
Fine	1000	100	528s	3420s

All BO strategies (baselines and SUR) start with $n_0 = 2 \times d$ observations generated from an optimized LHD, followed by respectively 60 (for DTLZ2) and 78 (for hartman) infill points. Each strategy is run 10 times.

Results for one run of SUR are given in Figure 4 in the form of projections on the marginal 2D spaces. Notice the central location of the KS point on this problem (Figure 4, first row), while the CKS point leans toward areas of larger densities (for instance, second line, third and fifth plots, CKS is close to the upper left corner). For KS, new points are added close to the reference solution, but some are also more exploratory, in particular near the individual minima to reduce uncertainty on the $(\mathbf{d}, \mathbf{u})$ line. For CKS, the behavior is more local, with points added mostly around the reference solution.

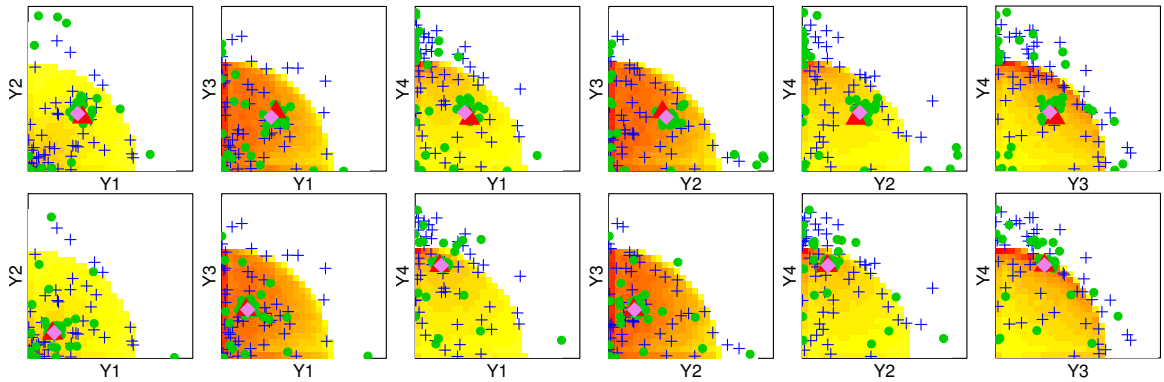

Figure 4: Results of one run for KS (top) and CKS (bottom), represented by using marginal 2D projections. The blue crosses are the initial design points, the green points the added designs, the red triangles the target true equilibria and pink diamonds the predicted ones. The heatmap represents the density of Pareto front points on the projected spaces.

Convergence results are provided in Figure 5 in terms of optimality gap with respect to the minimum benefit ratio, that is:

$$G_{KS}(\mathbf{s}) = \min_{1 \le i \le p} r^{(i)}(\mathbf{s}^*) - \min_{1 \le i \le p} r^{(i)}(\mathbf{s}), \quad G_{CKS}(\mathbf{s}) = \min_{1 \le i \le p} r_C^{(i)}(\mathbf{s}^\dagger) - \min_{1 \le i \le p} r_C^{(i)}(\mathbf{s}),$$

with $\mathbf{s}$ the evaluated solution, $\mathbf{s}^*$ and $\mathbf{s}^\dagger$ the actual KS and CKS solutions, respectively, r the ratio based on the actual nadir and utopia and r_C (as in Eq. (4)) computed using the actual objective values at the auxiliary set.

On DTLZ2, we see clearly that the hypervolume alternative never seeks central compromise solutions, and performs even worse than random search (this is despite the fact that the algorithm performs very well in terms of the hypervolume metric, see the empirical results in Binois and Picheny (2019)). The hypervolume approach is more competitive on hartman, which is due to the fact that the Pareto set actually corresponds to a relatively small part of the input space.

For both KS and CKS and both problems in the discrete case, the optimality gap is approximately log-linear on average, with a couple of runs that find the exact solution (gap

$\leq 10^{-4}$) within a very restricted budget. On all cases, SUR substantially outperform the baselines.

The difference between the coarse and fine SUR approaches is quite small on most cases, the only significant difference being on hartman, CKS, where the more expensive approach allows convergence to the exact solution much more quickly than the coarse one. On average, we can conclude that the numerical schemes to compute the SUR criterion provide a sufficient accuracy, and our "coarse" configuration results in reasonable running times (Figure 4.1) compared to the cost of an expensive experiment such as those described in the next subsections.

The effect of working on discretized versions of the problem is barely visible on DTLZ2 results, but clearly affects the rate of convergence on the hartman problem (although both SUR versions still converge to the actual solution).

Finally, an additional baseline is to reduce the number of objectives in order to use efficiently the standard multi-objective algorithms. We report in Appendix E additional experiments on the hartman case, that show that despite some correlation between objectives, dropping all but two objectives has on average a dramatic effect on the minimal ratios.

4.2. Training of a convolutional neural network

A growing need for BO methods emerges from machine learning applications, to replace manual tuning of hyperparameters of increasingly complex methods. One such example is with hyperparameters controlling the structure of a neural network. Since such methods are integrated in products, accuracy is not the only concern; and additional objectives have to be taken into account, such as prediction times.

4.2.1. Problem description

We consider here the training of a convolutional neural network (CNN) on the classical MNIST data (LeCun et al., 1998), with $60,000$ handwritten digits for training and an extra $10,000$ for testing. We use the `keras` package (Allaire and Chollet, 2018) to interface with the high-level neural networks API Keras (Chollet et al., 2015) to create and train a CNN. We follow a common structure for such a task, represented in Figure 6, with a first 2D convolutional layer, a first max pooling layer, then a second 2D convolutional layer and a second max pooling layer. Max pooling consists in keeping only the max over a window, introducing a small amount of translational invariance. Dropout, that is, randomly cutting off some neurons to increase robustness, is then applied before flattening to a dense layer, followed by another dropout before the final dense layer. Because of the dropout phases, the performance is random. This is handled by repeating five times each experiment, which takes up to 30 minutes on a desktop with a 3.2 Ghz quad-core processor and 4 Gb of RAM.

The six hyperparameters to tune, detailed in Table 2 in Appendix C, along with their range of variation, are the number of filters and dropout rates of each layer, plus the number of units of the last hidden layer and number of epochs.

The training of the CNN is performed based on the categorical cross-entropy. A validation data set is extracted from the training data to monitor overfitting, representing 20% of the initial size of the training data. The progress can be monitored by taking into account the accuracy (i.e., proportion of properly classified data) and cross-entropy on the training data,

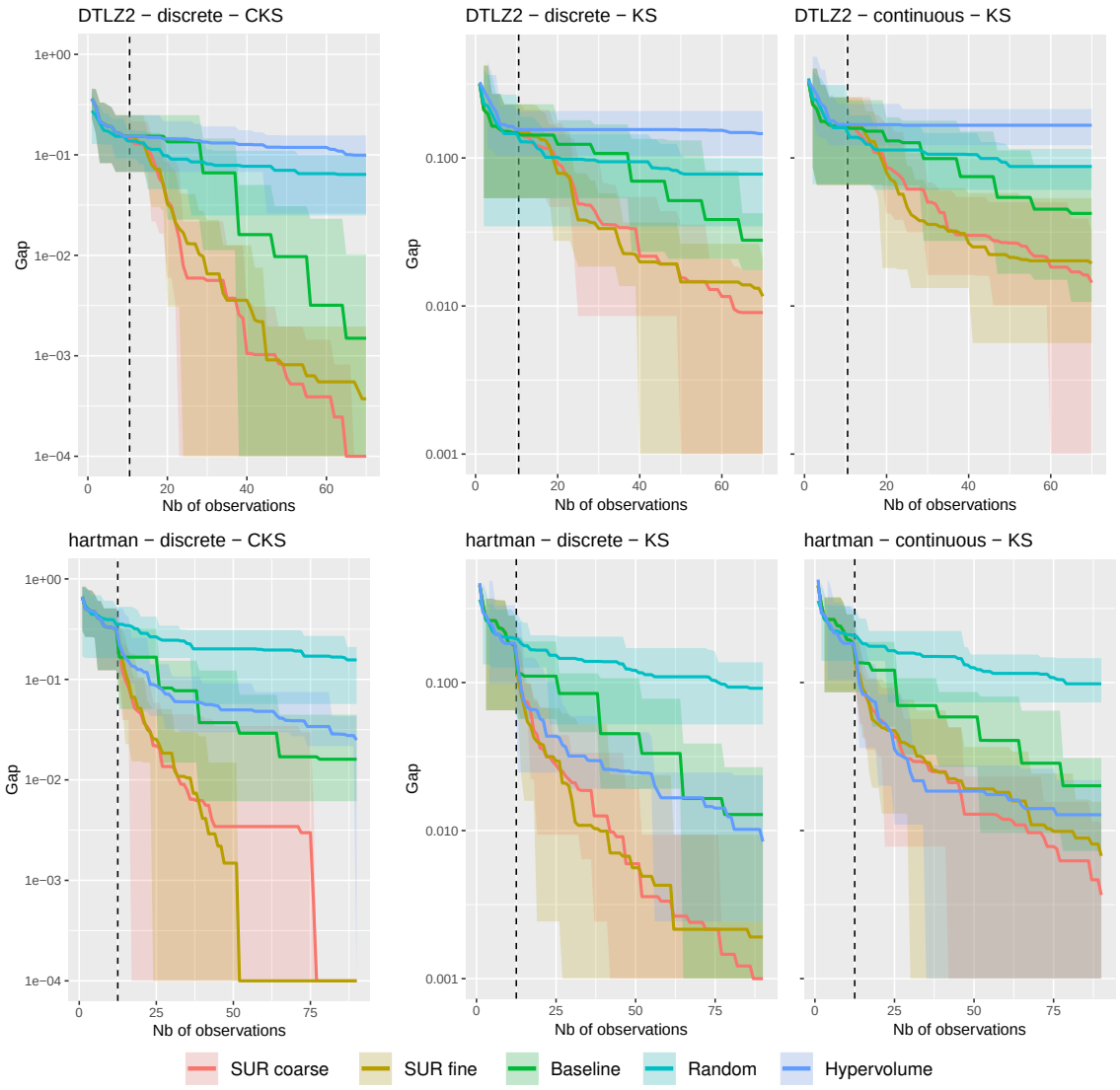

Figure 5: Optimality gap for KS (left) and CKS (right) on DTLZ2 (top) and hartman (bottom), for both discrete and continuous setups. Note that for readability, the Gap values are thresholded to 10^{-3} for KS and to 10^{-4} for CKS.

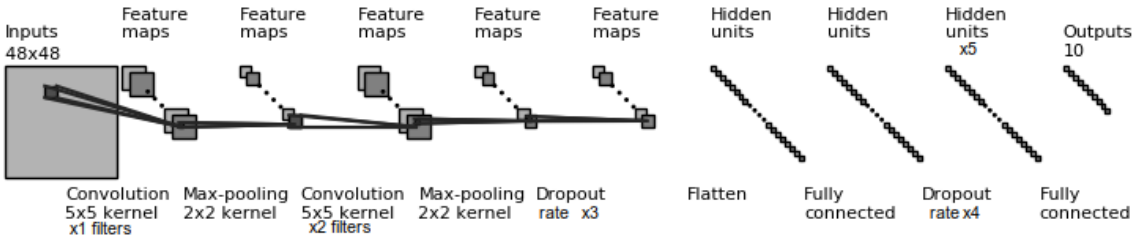

Figure 6: Architecture of the CNN on the MNIST data.

on the validation data or on the test data. Of these six corresponding objectives, training and validation accuracy are extremely correlated with the cross-entropy but they can be kept with our proposed methodology. Having both validation and testing metrics is also useful, the validation ones are driving the training, while the testing ones are less likely biased with data never used in the training. As additional objectives, we considered the training time of the CNN, as well as the prediction time on the testing data. This latter is relevant, for instance, when using a pretrained CNN for a given task. The eight objectives are summarized in Table 3 in Appendix C.

We take a total budget of 100 evaluations, split in half between initial LHD and sequential optimization using KS or CKS. GPs are trained by using Matérn 5/2 kernels with an estimated linear trend. In this case, $N = 500$ integration points were selected out of $N_{\text{large}} = 10^6$ possible candidates, renewed every iteration. The resulting time of each iteration is under a minute for KS and less than 10 minutes for CKS.

4.2.2. Results

Figure 7 represents the performance of networks obtained during optimization, in parallel coordinates (see Li et al., 2017) in both original scale (top) and in the copula space (bottom). As minimization is considered, minus the accuracies are reported. Selected 2D projections are provided in Figure 8, while the whole set of projections is in Appendix F. In the original objective coordinates, most observations have low values close to objective minima, with performance objectives that seem quite correlated. Nevertheless, 94 out of 150 observations are nondominated, and when looking at ranks in the copula domain, the Pareto front appears more complex, with conflicts between seemingly correlated objectives as shown in Figure 8. These could correspond to underfitting and overfitting. The nadir point is remarkably high in the objective space, due to the leading trade-off between training time and accuracy: bigger networks give better results but take more time. The KS and CKS solutions are close to each other in the original objective space, but much less so in the copula space, accounting for the concentration of low values on the objectives. The KS solution favors quicker training and prediction times than CKS, with seemingly marginally worse accuracy and errors. But the properly scaled CKS solution recognizes that faster architectures are actually quite extreme when looking at the ranks, thus giving more weights to a small decrease in accuracy and errors, which is significant in terms of ranks. As a result, the representation in parallel

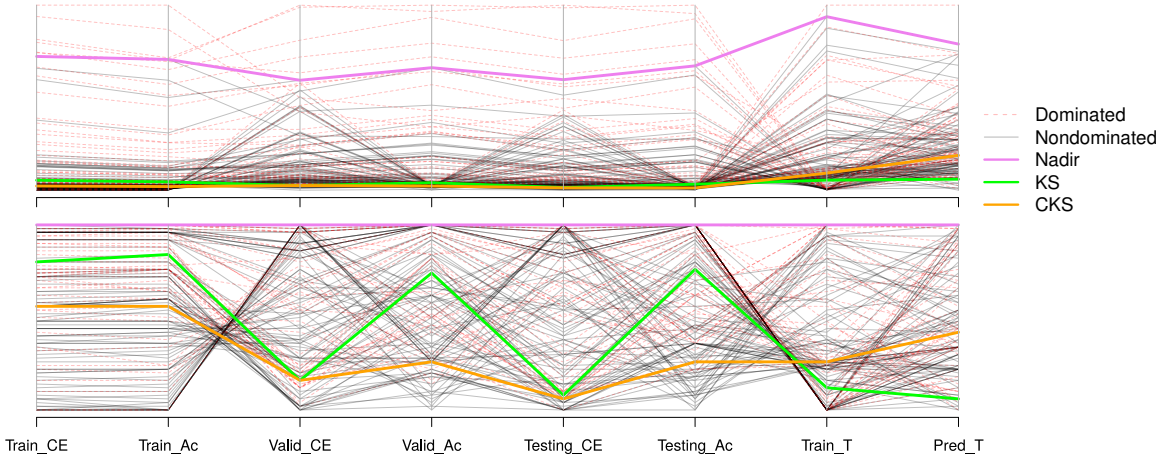

Figure 7: Parallel coordinates of the 150 observations and nadir point in the original (top) and copula (bottom) objective spaces (each line corresponds to a single observation, where objectives are rescaled on the graph). The colors highlight the dominated solutions and retained (KS and CKS) solutions.

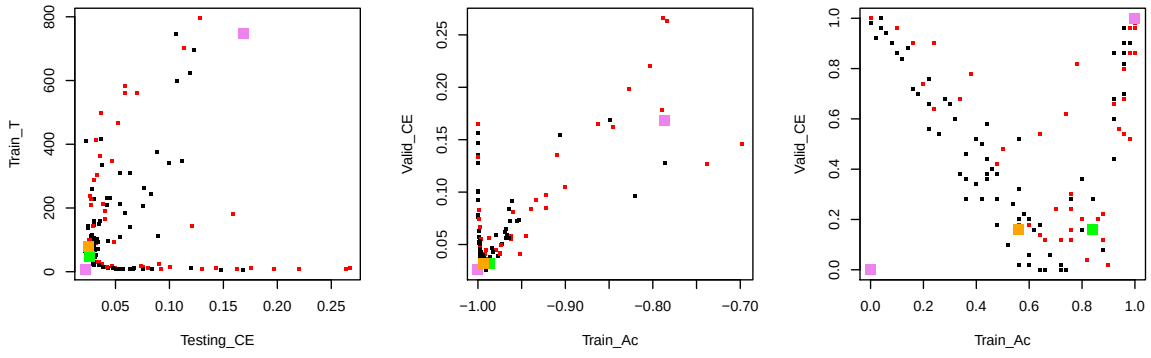

Figure 8: Projections on 2D spaces of the 150 observations, either in the original space (left, middle) or the copula space (right). The colors match the ones of Figure 7, with in addition the utopia point on the bottom left corner (pink square).

coordinates is relatively flat for CKS (a flat line would indicate a perfectly balanced solution) and quite irregular for KS, resulting in a much worse minimal ratio.

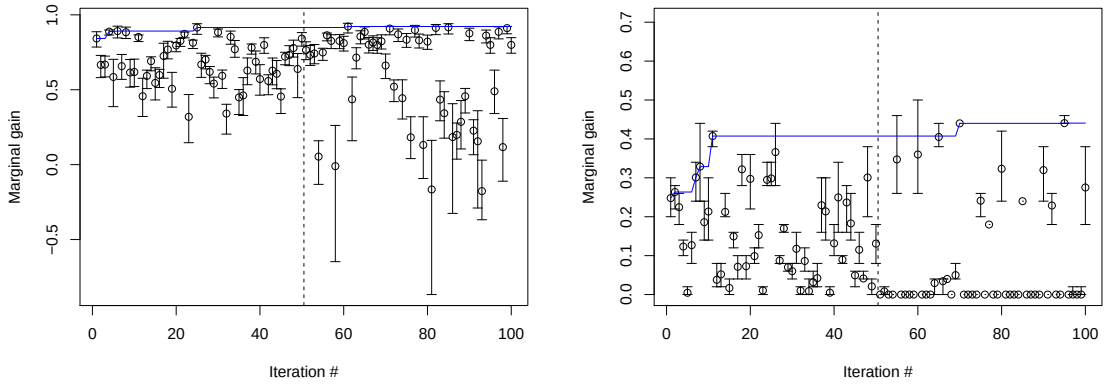

Figure 9: Evolution of the minimum ratio of the observations along the SUR algorithm for KS (left) and CKS (right). The vertical line shows where the SUR algorithm starts. Each graph corresponds to a specific ratio. The error bars account for uncertainty in the nadir and shadow, and the copula transformation, respectively.

More insight is provided by Figure 9, showing the progress over the iterations of the minimum ratio (as in Eq. (2) or (4), respectively) of the observations. Due to the computational cost, no ground truth is available, and in particular the true nadir and shadow points are unknown, which renders uncertain the ratio values. To estimate them as accurately and with as little bias as possible, we collected the 150 observations (coming from the three optimisation runs), fit a GP model to all this set and use it to estimate nadir and shadow points, as in Section 3.4.1. As a result, different ratios can be computed using the different nadir-shadow pairs, which are represented using error bars of Figure 12. A similar approach is followed for the copula transformation to obtain uncertainty for CKS.

The seemingly small progress on KS (around 1%) is related to the position of the nadir, very high on each objective. In fact, the numerous evaluations with almost zero (or negative) marginal gain correspond to the estimation of this nadir, which is crucial for KS estimation. For CKS, the progress is more visible, with a clear split between some exploitation steps (observations with high minimal ratio) and exploration steps (with very small minimal ratio). It is worth noticing that the degree of correlation between objectives did not seem to be problematic. In fact, we illustrate more thoroughly the benefits of keeping more objectives when they are all relevant in Appendix E.

With the CKS approach, we imposed a uniform distribution on the inputs. In fact, as we show in Figure 16 in appendix, when imposing random beta distributions on the inputs as in Figure 2 and estimating the corresponding copula based on the fitted GP models, the resulting CKS solutions remain concentrated and close to the uniform one. It could indicate some robustness with respect to the input distribution. Furthermore, it provides a richer pre-selection of Pareto optimal solutions to choose from, instead of a single one, similarly to the approaches proposed in Paria et al. (2019); Gaudrie et al. (2020).

Unless one is satisfied with the relative scaling of objectives of different nature, the CKS solution has the advantage of being more robust (to transformations) and, in addition, easier to estimate with a fixed nadir. The alternative to remain in the original space is to define preferences via the disagreement point, as is illustrated on the following test case.

For now, the six variables are taken as continuous. But these results could be extended by increasing the number of variables, including categorical variables affecting more profoundly the structure of the network, for instance relying on the work of Roustant et al. (2018).

4.3. Calibration of an agent-based behavioral model

Model calibration (sometimes referred to as inverse problem) consists of adjusting input parameters so that the model outputs match real-life data. In this experiment, we consider the calibration of the li-BIM model (Taillandier et al., 2017), implemented under the GAMA platform (Taillandier et al., 2018), which exhibits several challenging features: stochasticity, high numerical cost (approximately 30 minutes per run on a desktop computer with a 3.60 GHz eight-core processor and 32 Go RAM), and a large number of outputs.

4.3.1. Problem description

The Li-BIM model simulates the behavior of occupants in a building. It is structured around the numerical modeling of the building and an evolved occupational cognitive model developed with a belief-desire-intention (BDI) architecture (Bourgais et al., 2017). It simulates several quantities that strongly depend on the occupants, such as thermal conditions, air quality, lighting, etc. In the configuration considered here, three occupants are simulated over a period of one year.

In order to reproduce realistic conditions, 13 parameters can be tuned, related to either the occupant behavior or the building characteristics (see Table 4 in Appendix C for details). Nine outputs ($G_1 \dots G_9$) should match some target values ($T_1 \dots T_9$), chosen based on records or surveys (see Table 5 in Appendix C). Since the model is stochastic, we consider as objectives the squared expected relative differences between the outputs and targets:

$$y_i = \log\left(\left[\mathbb{E}\left(\frac{G_i - T_i}{T_i}\right)\right]^2 + \delta\right) \tag{12}$$

The logarithm transformation is useful here to bring more contrast for values close to zero, attenuated by a small δ (in our experiments, $\delta = 0.01$). In practice, we use estimates based on eight repeated runs. Note that such an objective focuses on the average behavior without considering the variability. As an alternative criterion, one may invert the square and expectation.

To solve this 13-variable 9-objective problem, we proceed as follow. An initial 100-point optimized LHD is generated, which is used to fit GPs (constant trend, Matérn 5/2 anisotropic covariance). From this initial design, both KS and CKS SUR strategies are conveyed independently with 100 additional points for each.

In addition, a third solution is sought by using KS with a partly prespecified disagreement point $\mathbf{d}$ to account for preferences (see Eq. (3)), so that the average error on five of the outputs does not exceed a certain percentage (either 50% or 30%), the other outputs being

unconstrained. To do so, we use

$$\tilde{d}_i = \min(N_i, c_i), \quad 1 \leq i \leq 9,$$

$N_i = \max_{\mathbf{x} \in \mathbb{X}^*} \ y^{(i)}(\mathbf{x})$ being the nadir i-th coordinate, and

$$c = \log([0.5, 0.5, +\infty, +\infty, 0.3, 0.5, +\infty, 0.5, +\infty]^2).$$

We refer to this strategy as KSpref.

Importantly, the GPs are used to fit the expected values of the outputs of the model ($\mathbb{E}G_i$) instead of the objectives (y_i). This greatly improves the prediction quality of the GPs (as G_i is smoother than $(G_i - T_i)^2$), while allowing us to convey our strategy almost without modification: on Sections 3.3 and 3.4, the drawings $\boldsymbol{\mathcal{Y}}$ and $\boldsymbol{\mathcal{F}}$ are obtained by first generating drawings $\boldsymbol{\mathcal{G}}$ of G, then transforming them ($\log \boldsymbol{\mathcal{G}}^2$).

We used $N^* = 1,000$ points for $\mathbb{X}^*$, taken from $\mathbb{X}_{\text{large}}$, a 2×10^5 space-filling design, both renewed at each iteration.

4.3.2. Results

The resulting designs of experiments and solutions are reported in graphical form in Figure 10, that shows parallel coordinates of the observations and Figure 11, that show a couple of 2D projections of the observations (the full set of 2D projections is available in Appendix G).

As a preliminary observation, of the 400 points computed during this experiment, only 55 were dominated. This result illustrates the exponential growth of Pareto sets with the number of objectives. As a consequence, in Figure 10 the non-dominated solutions (in black) cover most of the objective ranges, without apparent structure, and the nadir (in purple) is almost equal to the individual maxima.

Figure 11 shows that two pairs of objectives are strongly conflicting. Indeed, the error on Heat_W (the energy spent on heating) is in competition with the one on Winter_T (the room temperature in winter), which is intuitive, and T_relax and T_out, that are both time spent on different activities. The fact that no solution reconciles both pairs of objectives shows an intrinsic discrepancy between the data and the model, and the calibration task amounts to finding the best balance between the errors. Other pairwise antagonisms are visible in Appendix, in particular between the last 4 objectives (all corresponding to times spent on activities), or between T_cook and W_cook (time and energy corresponding to cooking).

On Figure 10, we see first that the three solutions are similar for some objectives (Heat_W, ECS_W, Cook_W, Winter_T) and differ substantially for others (in particular Other_W, Time_relax, Time_sleep, Time_out). The largest difference between the KS and KS with predefined disagreement point (PKS) appears on objectives Heat_W and Winter_W. This is explained by Figure 11 (left), where we can see that imposing a constraint on Heat_W shifts the trade-off between those objectives. The large difference between CKS and the two other solutions is explained by Figure 11 (middle and right). While KS is very central in the original space, the large concentration of solutions for large Time_out values induces a strong difference in the copula space, where CKS is central and the other solutions are not.

Finally, Figure 12 shows the performance of our algorithm, by representing the minimum ratio (as in Eq. (2) or (4), respectively) of the observations along the SUR iterations. As in

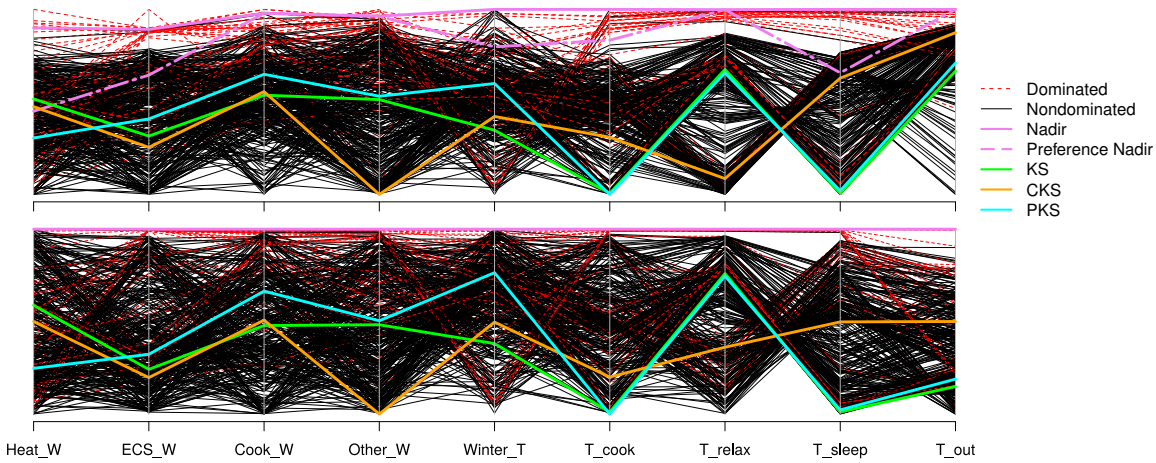

Figure 10: Parallel coordinates of the 400 observations, nadir and preference nadir points in the original (top) and copula (bottom) objective spaces (each line corresponds to a single observation, where objectives are rescaled on the graph). The colors highlight the dominated solutions and retained (KS, CKS and PKS) solutions.

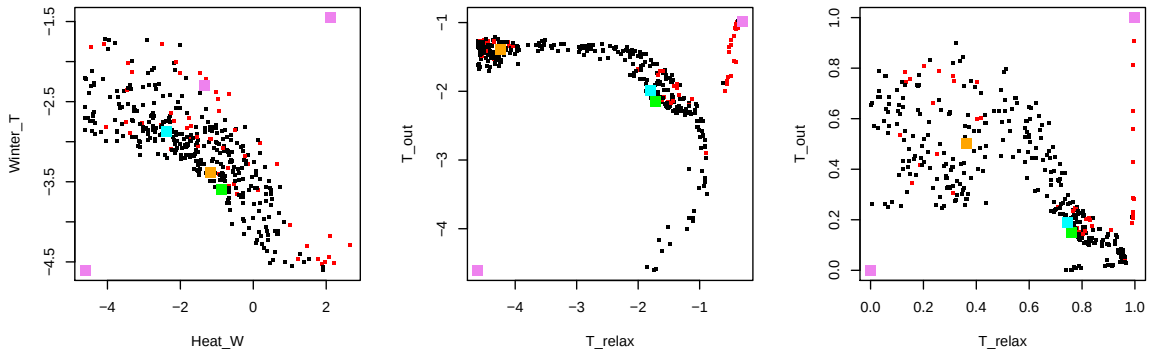

Figure 11: Projections on 2D spaces of the 400 observations, either in the original space (left, middle) or the copula space (right). The colors match the ones of Figure 10, with in addition the utopia point on the bottom left corner (pink square).

the CNN case, no ground truth is available, and we use a similar approach to account for the uncertainty in the nadir-shadow line and the copula estimates.

We can see that on all cases, the SUR algorithm improves substantially over the space-filling observations. Contrary to the toy problems, here SUR worked better on KS (with and without preferences) than on CKS (which might be simply explained by the presence of a very good initial solution for CKS). One may finally notice that the uncertainty is highest for KS and smallest for KS with preferences. This indicates that finding the nadir may be the most difficult of the learning tasks for this problem.

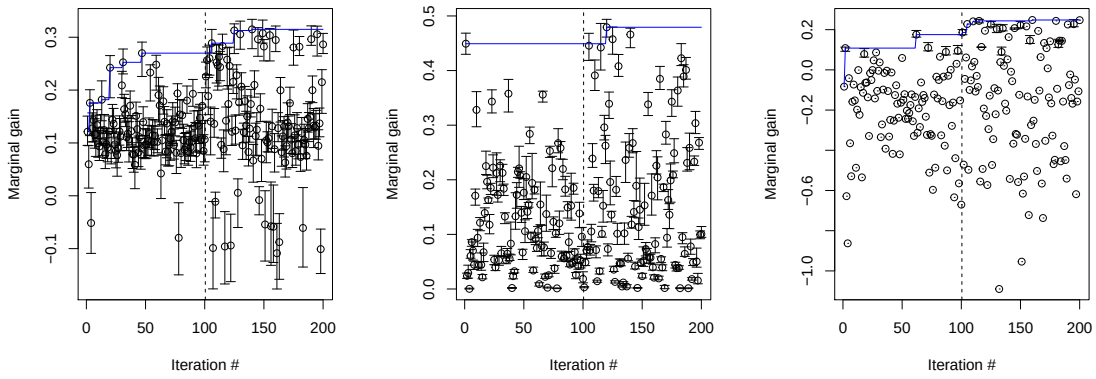

Figure 12: Evolution of the minimum ratio of the observations along the SUR algorithm for KS (left), CKS (middle) and KS with preferences (right). The vertical line shows where the SUR algorithm starts. Each graph corresponds to a specific ratio. The error bars account for uncertainty in the nadir and shadow and the copula transformation, respectively.

5. Conclusion and future work

In this article, we tackled many-objective problems by looking for a single well-balanced solution, originating from game theory. Two alternatives to this solution have been proposed, either by imposing preferences on some objective values (by specifying a disagreement point) or by working in the space of copulas, the latter solution being insensitive to monotonic transformations of the objectives. Looking for these solutions is in general a complex learning task. We proposed a tailored algorithm based on the stepwise uncertainty reduction paradigm, that automatically performs a trade-off between the different learning tasks (estimating the ideal point, the nadir point, and the marginals or exploring locally the space next to the estimated solution). We tested our algorithm on three different problems with growing complexity and found that well-balanced solutions could be obtained despite severely restricted budgets.

Choosing between the alternatives seems highly problem-dependent. On DTLZ2, KS appears as a more "central" (hence desirable) solution, while on the CNN tuning problem CKS is clearly a better choice, and on the calibration problem both solutions seem equally good. This difference may be imputed to scaling of the objectives prior to optimization. On

DTLZ2, all objectives behave similarly, while on CNN some are strongly heavy-tailed. On the other hand, CKS is dependent on the problem formulation (for instance by considering as an input $x \in [0, 1]$ or $\log(x) \in [-\infty, 0]$ could lead to very different results). Hence, a reasonable choice would be KS when the objectives formulation makes them directly comparable, and CKS when the input formulation is pertinent (in the sense that all parts of the input space have a roughly comparable effect on the outputs). As CKS is in general less likely to favor a particular objective over others, it may be used for more exploratory studies, and KS for a finer design on a pre-explored problem. Incorporating user preferences, as in the calibration problem, was proved easy and efficient and may direct the choice toward KS in such a case.

Many potential lines of future works remain. First, we may consider batch-sequential strategies instead of one observation at a time. This approach was not necessary in our test problems, since parallel computation was used to repeat simulations and average out noise. The SUR strategy naturally adapts to this case (Chevalier et al., 2014). In practice, one may also have to deal with asynchronous returns of batches, and possibly the use of several black-boxes with different costs, as in Hernández-Lobato et al. (2016b). An unused degree of freedom here is the number of replications to handle the noise, which might improve substantially the practical efficiency (Jalali et al., 2017; Binois et al., 2019). To do so, one may combine the presented approach with an efficient scheme for designing replications and estimating noise, as done by Binois et al. (2018).

Another direction is to consider alternative equilibria. A promising idea is to use a set of disagreement points instead of a single one: this could result in more robust solutions and potentially a small set of Pareto-optimal solutions, which might be preferred by decision-makers. Combining the KS solution with Nash games has been suggested in the game theory literature. Conley and Wilkie (1991) proposed using Nash equilibria as disagreement points, provided there exists a natural, or some relevant, splitting of the decision variable among the players. In the multicriteria Nash game framework (Ghose and Prasad, 1989), noncooperative players have to handle individual vector payoffs, so that to each request by other players, they have to respond with some payoff, rationally selected from their own Pareto front; a good candidate would be the KS solution. Both alternatives were poorly investigated in practice, mainly because their potential of application seems to be hindered by the lack of efficient tools that allow for an acceptable implementation. We think that our algorithmic framework could apply to both cases and provide a first solution to such problems.

Copula spaces have been used here mostly for rescaling. Taking advantage of the predictive capacity of copulas to accelerate the estimation of the Pareto front might accelerate substantially the search for the CKS solution. Furthermore, combining efficiently the two types of metamodels (GPs and copulas), as done by Wilson and Ghahramani (2010), may lead to new theoretical advances and algorithms. A complementary line of work would be to study the influence of the input distribution on CKS in order to define (or infer) distributions that lead to solutions that are independent of (or at least robust with respect to) the input problem formulation. Entertaining several inputs distributions, via a prior on the sampling distribution, is also a way to provide a more diverse set of solutions to the practitioners, and can be easily performed in parallel.

We leave to future work theoretical considerations on the convergence of the approach, following for instance the recent works on information-directed sampling by Russo and Van Roy (2014) or on SUR by Bect et al. (2019).

Acknowledgments

The work of MB is supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research under Contract No. DE-AC02-06CH11357. We thank Gail Pieper for her useful language editing.

Appendix A: Quantities related to Gaussian processes

GP moments We provide below the equations of the moments of a GP conditioned on n (noisy) observations $\mathbf{f} = (f_1, \ldots, f_n)$. Assuming a kernel function σ and a mean function $m(\mathbf{x})$, we have

$$\begin{aligned} \mu_n(\mathbf{x}) &= m(\mathbf{x}) + \lambda(\mathbf{x})\left(\mathbf{f} - m(\mathbf{x})\right), \\ \sigma_n^2(\mathbf{x}, \mathbf{x}') &= \sigma(\mathbf{x}, \mathbf{x}') - \lambda(\mathbf{x})\sigma(\mathbf{x}', \mathbf{X}_n), \end{aligned}$$

where

- $\lambda(\mathbf{x}) := \sigma(\mathbf{x}, \mathbf{X}_n)^\top \sigma(\mathbf{X}_n, \mathbf{X}_n)^{-1}$,
- $\sigma(\mathbf{x}, \mathbf{X}_n) := (\sigma(\mathbf{x}, \mathbf{x}_1), \ldots, \sigma(\mathbf{x}, \mathbf{x}_n))^\top$ and
- $\sigma(\mathbf{X}_n, \mathbf{X}_n) := (\sigma(\mathbf{x}_i, \mathbf{x}_j) + \tau_i^2 \delta_{i=j})_{1 \leq i,j \leq n}$,

δ standing for the Kronecker function.

Commonly, σ belongs to a parametric family of covariance functions such as the Gaussian and Matérn kernels, based on hypotheses about the smoothness of y. Corresponding hyperparameters are often obtained as maximum likelihood estimates; see e.g., Rasmussen and Williams (2006) for the corresponding details.

Expected improvement Denote $f_{\min} = \min_{1 \leq i \leq n}(f_i)$ the minimum of the observed values. The expected improvement is the expected positive difference between $f_{\min}$ and the new potential observation $Y_n(\mathbf{x})$:

$$\begin{aligned} EI(\mathbf{x}) &= \mathbb{E}\left(\max\left((0, f_{\min} - Y_n(\mathbf{x})\right)\right) \\ &= (f_{\min} - \mu_n(\mathbf{x}))\Phi\left(\frac{f_{\min} - \mu_n(\mathbf{x})}{\sigma_n(\mathbf{x}, \mathbf{x})}\right) + \sigma_n^2(\mathbf{x})\phi\left(\frac{f_{\min} - \mu_n(\mathbf{x})}{\sigma_n(\mathbf{x}, \mathbf{x})}\right), \end{aligned}$$

where ϕ and Φ are respectively the PDF and CDF of the standard Gaussian variable.

For maximization, we use, with $f_{\max} = \max_{1 \leq i \leq n}(f_i)$:

$$EI_{\text{nad}}(\mathbf{x}) = \mathbb{E}\left(\max\left((0, Y_n(\mathbf{x}) - f_{\max}\right)\right).$$

Probability to belong to a box ($p_{\mathbf{box}}$) Let $LB \in \mathbb{R}^p$ and $UB \in \mathbb{R}^p$ such that $\forall 1 \leq i \leq p, LB^i < UB^i$ define a box in the objective space. Defining $\mathbf{\Psi} = [\Psi(\boldsymbol{\mathcal{Y}}_1), \ldots, \Psi(\boldsymbol{\mathcal{Y}}_M)]$ the $p \times M$ matrix of simulated KS solutions, we use

$$\forall 1 \leq i \leq p \qquad LB^i = \min \mathbf{\Psi}_{i,1\ldots M} \quad \text{and} \quad UB^i = \max \mathbf{\Psi}_{i,1\ldots M}.$$

Then, the probability to belong to the box is

$$p_{\text{box}}(\mathbf{x}) = \prod_{i=1}^{p} \left[\Phi\left(\frac{UB^i - \mu_n^i(\mathbf{x})}{\sigma_n^i(\mathbf{x}, \mathbf{x})}\right) - \Phi\left(\frac{\mu_n^i(\mathbf{x}) - LB^i}{\sigma_n^i(\mathbf{x}, \mathbf{x})}\right)\right].$$

Probability of nondomination Let $\mathbb{X}_n^*$ be the subset of nondominated observations. The probability of non-domination is

$$p_{\mathrm{ND}}(\mathbf{x}) = \mathbb{P}\left(\forall \mathbf{x}^* \in \mathbb{X}_n^*, \exists k \in \{1,\dots,p\} \text{ such that } Y_n^{(k)}(\mathbf{x}) \leq Y_n^{(k)}(\mathbf{x}^*)\right).$$

Using the GP equations for Y_n, one can compute $p_{ND}(\mathbf{x})$ in closed form, for $p \leq 3$. We refer to the work of Couckuyt et al. (2014) for the formulas expressed in an efficient form. For a larger number of objectives, this probability must be computed by Monte-Carlo.

Appendix B: CNN lists of inputs and outputs

Table 2: List of inputs for the CNN training problem.

	Description	Min	Max
x_1	number of filters of first convolutional layer	2	100
x_2	number of filters of second convolutional layer	2	100
x_3	first dropout rate	0	0.5
x_4	second dropout rate	0	0.5
x_5	number of units of dense hidden layer	10	1000
x_6	number of epochs	2	50

Table 3: List of outputs for the CNN training problem

Name	Unit
Training cross-entropy	-
Training accuracy	-
Validation cross-entropy	-
Validation accuracy	-
Testing cross-entropy	-
Testing accuracy	-
Training time	s
Prediction time	s

Appendix C: li-BIM list of inputs and outputs

Table 4: List of inputs of the li-BIM model.

Name	Unit	Meaning	Min	Max
Cth_h	J/K	Thermal capacity of the dwelling	10^6	10^7
RD	W	Average power of the relaxing devices	150	1000
HW	W	Power of the boiler to produce hot water	500	3000
CD	W	Average power of the cooking devices	100	300
PmaxHeat	kW	Maximum power of the boiler to heat dwelling	500	3000
SensitiveCold	[0-1]	Sensitivity of the occupant to cold temperature	0.2	1.0
SensitiveWarm	[0-1]	Sensitivity of the occupant to warm temperature	0.5	1
NbHfreshair	hours	Average number of hours between two outings	16	36
NbHtire	hours	Average number of hours between two sleeps	8	60
NbHhungry	hours	Average number of hours between two meals	6	16
NbHdirty	hours	Average number of hours between two showers/baths	20	48
Deltamodif	hours	Time before new action if previous action insufficient	5	30
Thermal_effort	Celsius	Max difference to ideal temperature before acting	2	15

Table 5: List of outputs of the li-BIM model.

Name	Unit	Meaning	Target
Heat_W	kWh	Total energetic consumption of heating devices	1384
ECS_W	kWh	Total energetic consumption of hot water devices	1198
Cook_W	kWh	Total energetic consumption of cooking devices	306
Other_W	kWh	Total energetic consumption of other devices	1751
Winter_T	°C	Average temperature during winter	21.8
Time_cook	hours	Average time spent cooking	0.9
Time_relax	hours	Average time spent relaxing	3.7
Time_sleep	hours	Average time spent sleeping	8.35
Time_out	hours	Average time spent outside the building	0.58

Appendix D: hartman toy problem

The original hartman function, defined on $[0,1]^6$, is:

$$h(\mathbf{x}) = \frac{-1}{1.94}\left[2.58 + \sum_{i=1}^{4} C_i \exp\left(-\sum_{j=1}^{6} a_{ji}\left(x_j - p_{ji}\right)^2\right)\right]$$

with: $\mathbf{C} = [1.0, 1.2, 3.0, 3.2]$,

$$\mathbf{a} = \begin{bmatrix} 10.00 & 0.05 & 3.00 & 17.00 \\ 3.00 & 10.00 & 3.50 & 8.00 \\ 17.00 & 17.00 & 1.70 & 0.05 \\ 3.50 & 0.10 & 10.00 & 10.00 \\ 1.70 & 8.00 & 17.00 & 0.10 \\ 8.00 & 14.00 & 8.00 & 14.00 \end{bmatrix}, \mathbf{p} = \begin{bmatrix} 0.1312 & 0.2329 & 0.2348 & 0.4047 \\ 0.1696 & 0.4135 & 0.1451 & 0.8828 \\ 0.5569 & 0.8307 & 0.3522 & 0.8732 \\ 0.0124 & 0.3736 & 0.2883 & 0.5743 \\ 0.8283 & 0.1004 & 0.3047 & 0.1091 \\ 0.5886 & 0.9991 & 0.6650 & 0.0381 \end{bmatrix}.$$

To obtain six objectives, we rescale and translate the inputs, so that:

$$\begin{aligned}
f_1(\mathbf{x}) &= -\log(-h([\tilde{x}_1, \tilde{x}_2, \tilde{x}_3, \tilde{x}_4, \tilde{x}_5, \tilde{x}_6]) \\
f_2(\mathbf{x}) &= -\log(-h(1/2 + [\tilde{x}_6, \tilde{x}_5, \tilde{x}_4, \tilde{x}_3, \tilde{x}_2, \tilde{x}_1]) \\
f_3(\mathbf{x}) &= -\log(-h([\tilde{x}_2, 1/2 + \tilde{x}_4, \tilde{x}_6, \tilde{x}_1, 1/2 + \tilde{x}_3, 1/2 + \tilde{x}_5]) \\
f_4(\mathbf{x}) &= -\log(-h([1/2 + \tilde{x}_5, \tilde{x}_3, 1/2 + \tilde{x}_1, 1/2 + \tilde{x}_2, \tilde{x}_4, \tilde{x}_6]) \\
f_5(\mathbf{x}) &= -\log(-h([\tilde{x}_3, \tilde{x}_6, \tilde{x}_1, 1/2 + \tilde{x}_4, 1/2 + \tilde{x}_2, 1/2 + \tilde{x}_5]) \\
f_6(\mathbf{x}) &= -\log(-h([1/2 + \tilde{x}_4, 1/2 + \tilde{x}_2, 1/2 + \tilde{x}_6, \tilde{x}_5, \tilde{x}_3, \tilde{x}_1]),
\end{aligned}$$

with $\tilde{\mathbf{x}} = \mathbf{x}/2$.

Appendix E: Effect of taking more objectives

To avoid many-objective challenges, a natural way is to select fewer objectives that are taken as more important, or looking at correlations. In this section we illustrate that even if discarding objectives gives some improvements in the solutions over the selected ones, overall more may be lost on the discarded objectives. To this end, we consider the 6 objectives hartman problem.

The setup is as follows: a reference KS solution is computed on this toy example, then we estimate the Pareto front for all combinations of two objectives. The results are given in Figure 13 and summarized in Figure 14. The best ratios over two objectives are better than the KS ones, but on average more is lost on the remaining objectives, advocating to keep all objectives and perhaps use preferences rather that discarding objectives. Looking uniformly at points on those 2-objective Pareto fronts, the gain is less important while the loss on the remaining objective is even larger.

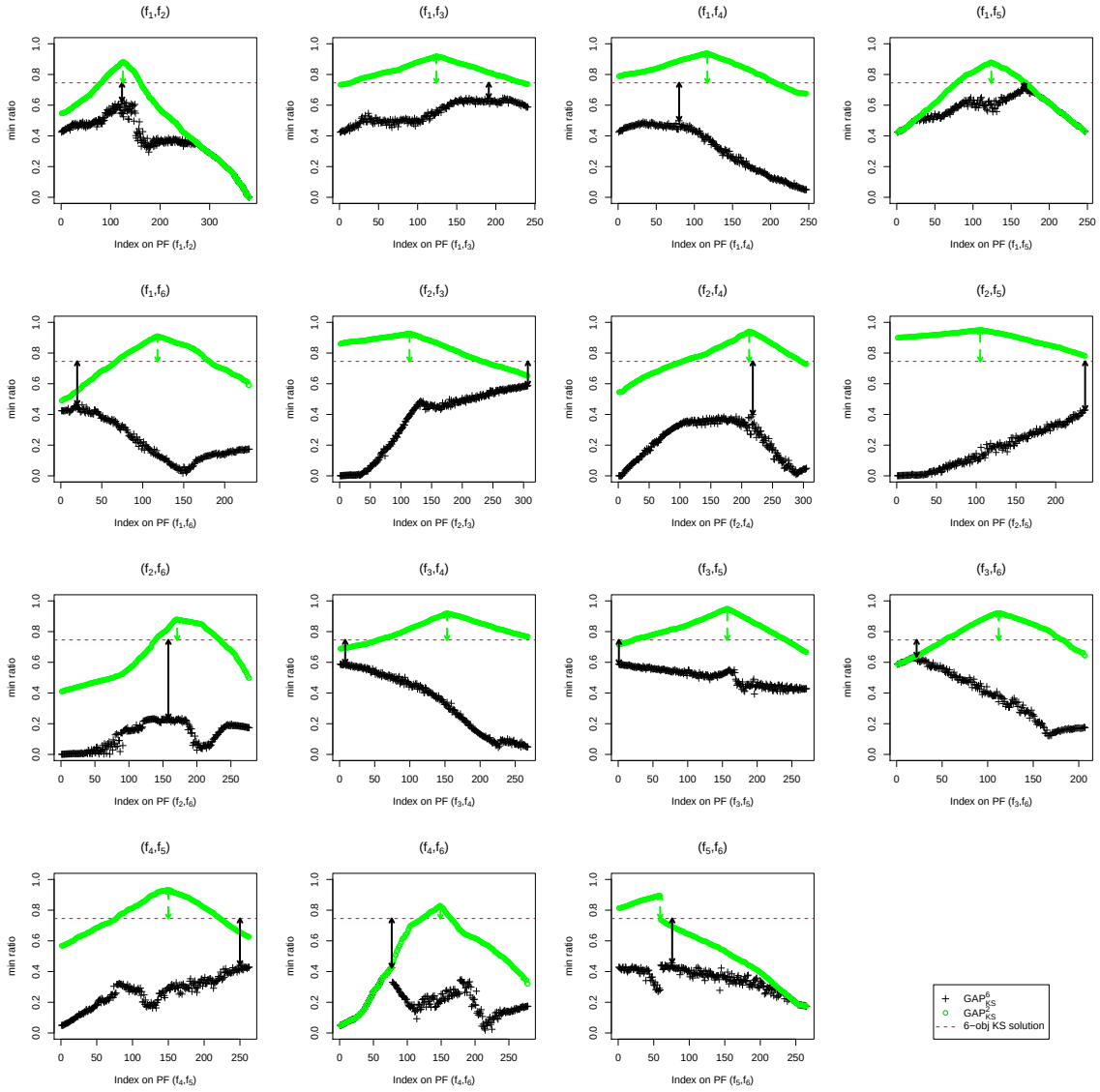

Figure 13: Benefit ratios of Pareto optimal solutions for combinations of two objectives, over the same two objectives (green ∘) and over all objectives (black +). The corresponding performance of the 6 objectives KS solution is depicted by the red dashed line. The black arrows (resp. dashed green) marks the difference between the best max min ratio over 6 (resp. 2) objectives bi-objective Pareto fronts solution and the KS.

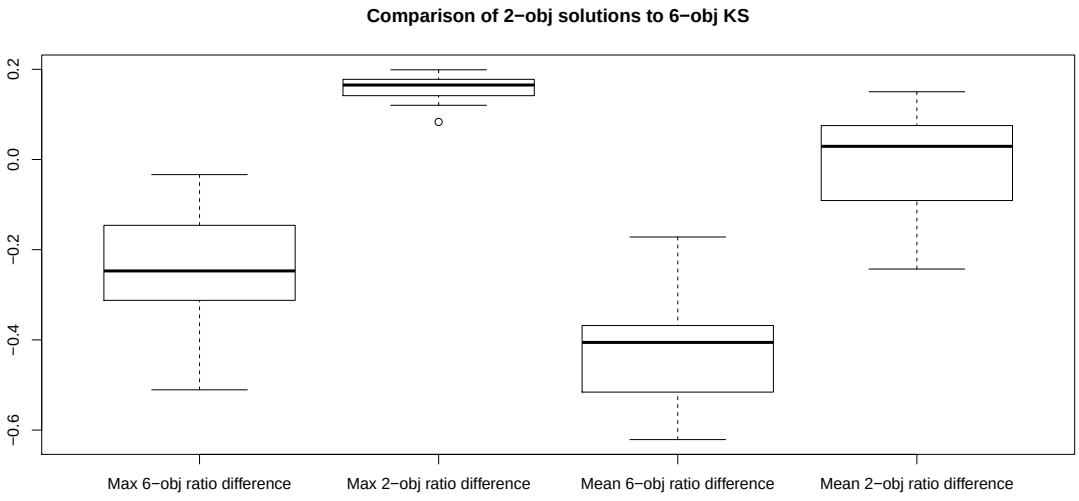

Figure 14: Loss of discarding 4 objectives on their benefit ratio (black minus KS) versus gain on the remaining two (green minus KS), either for the max (i.e., KS on 2-objective Pareto fronts, depicted by arrows in Figure 13) or the mean .

Appendix F: additional results on the convolutional neural network problem

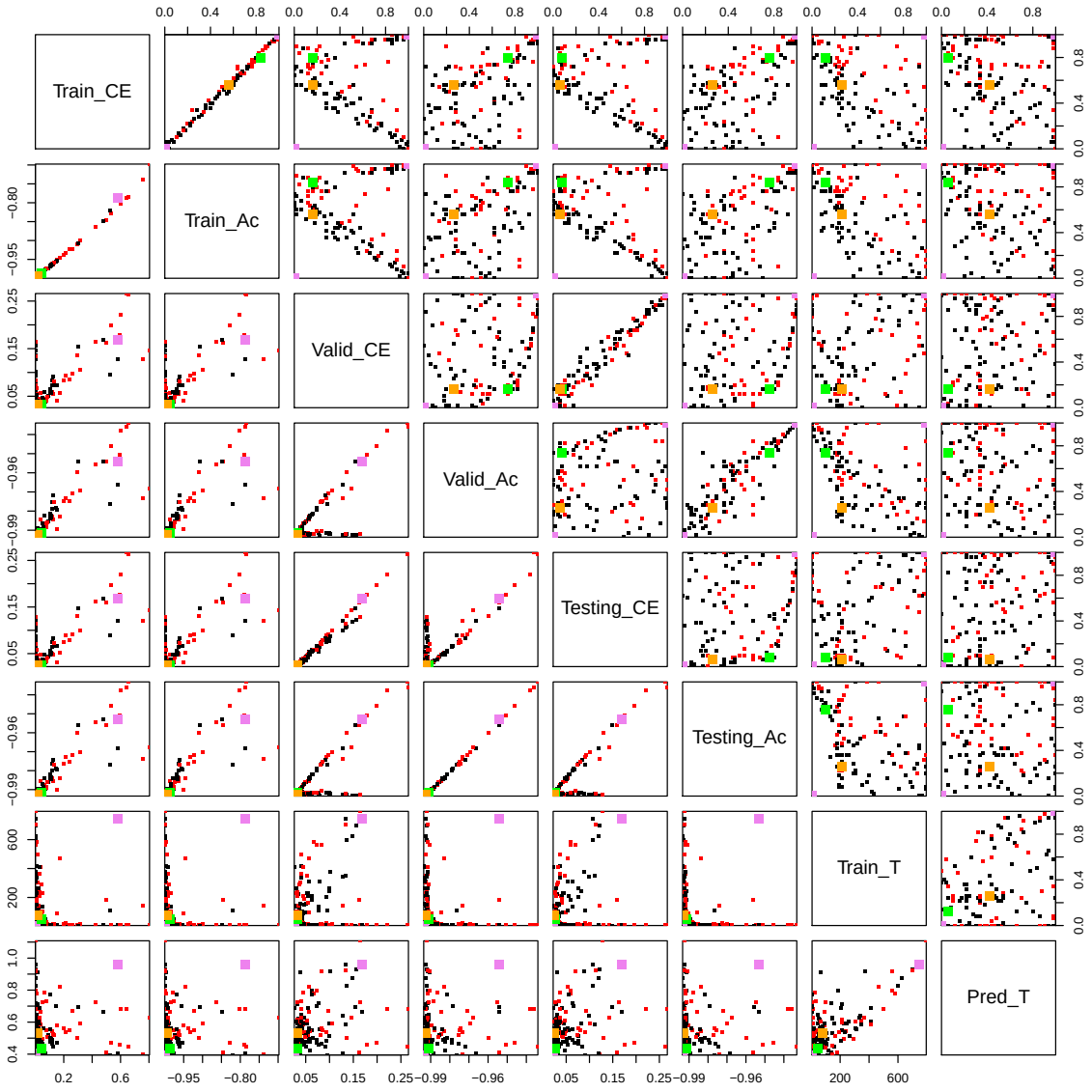

Figure 15: Projections on 2D spaces of the 150 observations, either in the original space (lower left triangle) or the copula space (upper right triangle). The colors match the ones of Figure 7. One may observe in general the central position of KS in the original space and the central position of CKS in the copula space.

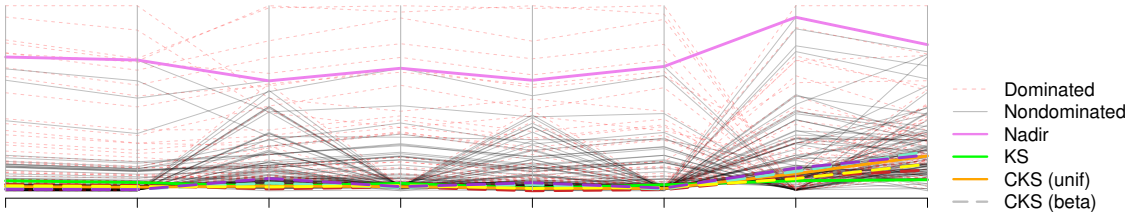

Figure 16: Elements are the same as in Figure 7, with in addition CKS solutions corresponding to four random beta distributions on the inputs (long-dashed line in blue, yellow, dark red and violet), echoing Figure 2 bottom-right.

Appendix G: additional results on the calibration problem

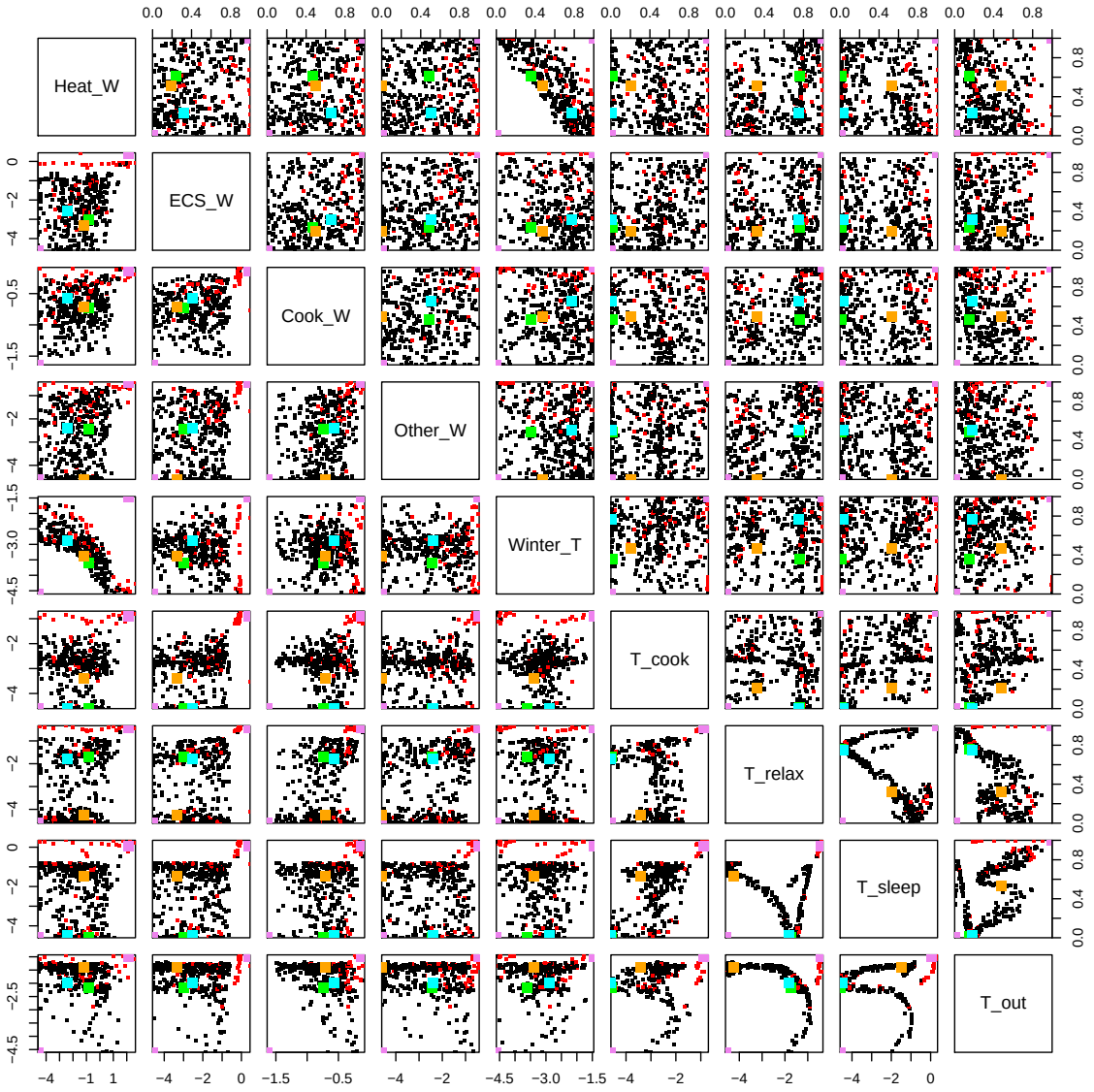

Figure 17: Projections on 2D spaces of the 400 observations, either in the original space (lower left triangle) or the copula space (upper right triangle). The colors match the ones of Figure 10. One may observe in general the central position of KS in the original space and the central position of CKS in the copula space.

References

Majid Abdolshah, Alistair Shilton, Santu Rana, Sunil Gupta, and Svetha Venkatesh. Multi-objective Bayesian optimisation with preferences over objectives. In *Advances in Neural Information Processing Systems*, pages 12214–12224, 2019.

Badr Abou El Majd, Jean-Antoine Desideri, and Abderrahmane Habbal. Optimisation de forme fluide-structure par un jeu de Nash. *Revue Africaine de la Recherche en Informatique et Mathématiques Appliquées*, 13:3–15, 2010.

Joseph J. Allaire and François Chollet. *keras: R Interface to 'Keras'*, 2018. URL `https://keras.rstudio.com`. R package version 2.1.4.

Md Asafuddoula, Tapabrata Ray, and Ruhul Sarker. A decomposition-based evolutionary algorithm for many objective optimization. *IEEE Transactions on Evolutionary Computation*, 19(3):445–460, 2015.

Johannes Bader and Eckart Zitzler. Hype: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation*, 19(1):45–76, 2011.

Slim Bechikh, Lamjed Ben Said, and Khaled Ghedira. Estimating nadir point in multi-objective optimization using mobile reference points. In *IEEE Congress on Evolutionary Computation*, pages 1–9. IEEE, 2010.

Julien Bect, David Ginsbourger, Ling Li, Victor Picheny, and Emmanuel Vazquez. Sequential design of computer experiments for the estimation of a probability of failure. *Statistics and Computing*, 22(3):773–793, 2012.

Julien Bect, François Bachoc, David Ginsbourger, et al. A supermartingale approach to gaussian process based sequential design of experiments. *Bernoulli*, 25(4A):2883–2919, 2019.

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *The Journal of Machine Learning Research*, 13:281–305, 2012.

James S Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems*, pages 2546–2554, 2011.

Mickaël Binois and Victor Picheny. GPareto: An R package for Gaussian-process-based multi-objective optimization and analysis. *Journal of Statistical Software*, 89(1):1–30, 2019.

Mickaël Binois, Didier Rullière, and Olivier Roustant. On the estimation of Pareto fronts from the point of view of copula theory. *Information Sciences*, 324:270 – 285, 2015.

Mickaël Binois, Robert B Gramacy, and Mike Ludkovski. Practical heteroscedastic Gaussian process modeling for large simulation experiments. *Journal of Computational and Graphical Statistics*, 27(4):808–821, 2018.

Mickaël Binois, Jiangeng Huang, Robert B Gramacy, and Mike Ludkovski. Replication or exploration? Sequential design for stochastic simulation experiments. *Technometrics*, 61 (1):7–23, 2019.

Mathieu Bourgais, Patrick Taillandier, and Laurent Vercouter. Enhancing the behavior of agents in social simulations with emotions and social relations. In *18th workshop on Multi-Agent-Based Simulation-MABS 2017*, 2017.

Irem Bozbay, Franz Dietrich, and Hans Peters. Bargaining with endogenous disagreement: The extended Kalai–Smorodinsky solution. *Games and Economic Behavior*, 74(1):407–417, 2012.

Abraham Charnes and William Wager Cooper. Goal programming and multiple objective optimizations: Part 1. *European journal of operational research*, 1(1):39–54, 1977.

Ran Cheng, Miqing Li, Ye Tian, Xingyi Zhang, Shengxiang Yang, Yaochu Jin, and Xin Yao. A benchmark test suite for evolutionary many-objective optimization. *Complex & Intelligent Systems*, 3(1):67–81, 2017.

Clément Chevalier, Julien Bect, David Ginsbourger, Emmanuel Vazquez, Victor Picheny, and Yann Richet. Fast parallel kriging-based stepwise uncertainty reduction with application to the identification of an excursion set. *Technometrics*, 56(4):455–465, 2014.

Clément Chevalier, Xavier Emery, and David Ginsbourger. Fast update of conditional simulation ensembles. *Mathematical Geosciences*, 47(7):771–789, 2015.

François Chollet et al. Keras. `https://github.com/fchollet/keras`, 2015.

Tinkle Chugh, Karthik Sindhya, Jussi Hakanen, and Kaisa Miettinen. A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms. *Soft Computing*, pages 1–30, 2017.

Tinkle Chugh, Yaochu Jin, Kaisa Miettinen, Jussi Hakanen, and Karthik Sindhya. A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 22(1):129–143, 2018.

John P Conley and Simon Wilkie. The bargaining problem without convexity: Extending the egalitarian and Kalai-Smorodinsky solutions. *Economics Letters*, 36(4):365–369, 1991.

Ivo Couckuyt, Dirk Deschrijver, and Tom Dhaene. Fast calculation of multiobjective probability of improvement and expected improvement criteria for Pareto optimization. *Journal of Global Optimization*, 60(3):575–594, 2014.

Indraneel Das and John E Dennis. Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, 8(3):631–657, 1998.

Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on Evolutionary Computation*, 6(2):182–197, 2002.

Kalyanmoy Deb, Shamik Chaudhuri, and Kaisa Miettinen. Towards estimating nadir objective vector using evolutionary approaches. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 643–650. ACM, 2006.

Thomas Desautels, Andreas Krause, and Joel W Burdick. Parallelizing exploration-exploitation tradeoffs in gaussian process bandit optimization. *The Journal of Machine Learning Research*, 15(1):3873–3923, 2014.

Jean-Antoine Désidéri, Régis Duvigneau, and Abderrahmane Habbal. *Computational Intelligence in Aerospace Sciences, V. M. Becerra and M. Vassile Eds.*, volume 244 of *Progress in Astronautics and Aeronautics*, chapter Multi-Objective Design Optimization Using Nash Games. AIAA, 2014.

Peter Diggle and Paulo Justiniano Ribeiro. *Model-Based Geostatistics*. Springer, 2007.

Laurence Charles Ward Dixon and Giorgio P Szegö. *Towards global optimisation*, volume 2. North-Holland Amsterdam, 1978.

Valerii Vadimovich Fedorov. *Theory of Optimal Experiments*. Elsevier, 1972.

David Gaudrie, Rodolphe Le Riche, Victor Picheny, Benoit Enaux, and Vincent Herbert. Targeting solutions in Bayesian multi-objective optimization: sequential and batch versions. *Annals of Mathematics and Artificial Intelligence*, 88(1):187–212, 2020.

Debasish Ghose and UR Prasad. Solution concepts in two-person multicriteria games. *Journal of Optimization Theory and Applications*, 63(2):167–189, 1989.

Jussi Hakanen and Joshua D Knowles. On using decision maker preferences with ParEGO. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 282–297. Springer, 2017.

Philipp Hennig and Christian J Schuler. Entropy search for information-efficient global optimization. *The Journal of Machine Learning Research*, 13:1809–1837, 2012.

Daniel Hernández-Lobato, Jose Hernandez-Lobato, Amar Shah, and Ryan Adams. Predictive entropy search for multi-objective Bayesian optimization. In *International Conference on Machine Learning*, pages 1492–1501, 2016a.

José Miguel Hernández-Lobato, Michael A Gelbart, Ryan P Adams, Matthew W Hoffman, and Zoubin Ghahramani. A general framework for constrained Bayesian optimization using information-based search. *Journal of Machine Learning Research*, 17(160):1–53, 2016b.

Jens Leth Hougaard and Mich Tvede. Nonconvex n-person bargaining: efficient maxmin solutions. *Economic Theory*, 21(1):81–95, 2003.

Hisao Ishibuchi, Noritaka Tsukamoto, and Yusuke Nojima. Evolutionary many-objective optimization: A short review. In *IEEE Congress on Evolutionary Computation, 2008. CEC 2008.*, pages 2419–2426. IEEE, 2008.

Hisao Ishibuchi, Yu Setoguchi, Hiroyuki Masuda, and Yusuke Nojima. Performance of decomposition-based many-objective algorithms strongly depends on pareto front shapes. *IEEE Transactions on Evolutionary Computation*, 21(2):169–190, 2016.

Hamed Jalali, Inneke Van Nieuwenhuyse, and Victor Picheny. Comparison of kriging-based algorithms for simulation optimization with heterogeneous noise. *European Journal of Operational Research*, 261(1):279–301, 2017.

Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

Ulrich Junker. Preference-based search and multi-criteria optimization. *Annals of Operations Research*, 130(1-4):75–115, 2004.

Ehud Kalai and Meir Smorodinsky. Other solutions to Nash's bargaining problem. *Econometrica*, 43:513–518, 1975.

Joshua Knowles. ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66, February 2006.

Saku Kukkonen and Jouni Lampinen. Ranking-dominance and many-objective optimization. In *IEEE Congress on Evolutionary Computation, 2007. CEC 2007.*, pages 3983–3990. IEEE, 2007.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Miqing Li, Liangli Zhen, and Xin Yao. How to read many-objective solution sets in parallel coordinates [educational forum]. *IEEE Computational Intelligence Magazine*, 12(4):88–100, 2017.

Michael D McKay, Richard J Beckman, and William J Conover. Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, 1979.

Kaisa Miettinen. *Nonlinear multiobjective optimization*, volume 12. Springer Science & Business Media, 2012.

Roger B Nelsen. *An Introduction to Copulas*. Springer, 2006.

Harald Niederreiter. Low-discrepancy and low-dispersion sequences. *Journal of Number Theory*, 30(1):51–70, 1988.

Jeremy Oakley. Estimating percentiles of uncertain computer code outputs. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 53(1):83–93, 2004.

Marek Omelka, Irène Gijbels, and Noël Veraverbeke. Improved kernel estimation of copulas: weak convergence and goodness-of-fit testing. *The Annals of Statistics*, 37(5B):3023–3058, 2009.

Biswajit Paria, Kirthevasan Kandasamy, and Barnabás Póczos. A flexible framework for multi-objective Bayesian optimization using random scalarizations. In *UAI*, 2019.

James Parr. *Improvement criteria for constraint handling and multiobjective optimization*. PhD thesis, University of Southampton, 2013.

Victor Picheny. Multiobjective optimization using Gaussian process emulators via stepwise uncertainty reduction. *Statistics and Computing*, pages 1–16, 2013.

Victor Picheny and Mickaël Binois. *GPGame: Solving Complex Game Problems using Gaussian Processes*, 2018. URL `http://CRAN.R-project.org/package=GPGame`. R package version 1.1.0.

Victor Picheny, Tobias Wagner, and David Ginsbourger. A benchmark of kriging-based infill criteria for noisy optimization. *Structural and Multidisciplinary Optimization*, 48(3): 607–626, 2013.

Victor Picheny, Mickaël Binois, and Abderrahmane Habbal. A Bayesian optimization approach to find Nash equilibria. *Journal of Global Optimization*, 73(1):171–192, 2019.

Wolfgang Ponweiser, Tobias Wagner, Dirk Biermann, and Markus Vincze. Multiobjective optimization on a limited budget of evaluations using model-assisted s-metric selection. In *International Conference on Parallel Problem Solving from Nature*, pages 784–794. Springer, 2008.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2018. URL `https://www.R-project.org/`.

Carl E. Rasmussen and Christopher Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006. URL `http://www.gaussianprocess.org/gpml/`.

Olivier Roustant, Esperan Padonou, Yves Deville, Aloïs Clément, Guillaume Perrin, Jean Giorla, and Henry Wynn. Group kernels for Gaussian process metamodels with categorical inputs. *arXiv preprint arXiv:1802.02368*, 2018.

Daniel Russo and Benjamin Van Roy. Learning to optimize via information-directed sampling. In *Advances in Neural Information Processing Systems*, pages 1583–1591, 2014.

Matthias Schonlau, William J Welch, and Donald R Jones. Global versus local search in constrained optimization of computer models. *Lecture Notes-Monograph Series*, pages 11–25, 1998.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando de Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

Hemant Kumar Singh, Amitay Isaacs, and Tapabrata Ray. A Pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems. *IEEE Transactions on Evolutionary Computation*, 15(4):539–556, 2011.

Sean C Smithson, Guang Yang, Warren J Gross, and Brett H Meyer. Neural networks designing neural networks: Multi-objective hyper-parameter optimization. In *2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 1–8. IEEE, 2016.

Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Information-theoretic regret bounds for Gaussian process optimization in the bandit setting. *Information Theory, IEEE Transactions on*, 58(5):3250–3265, 2012.

Joshua D. Svenson. *Computer Experiments: Multiobjective Optimization and Sensitivity Analysis*. PhD thesis, The Ohio State University, 2011.

Mohammad Tabatabaei, Markus Hartikainen, Karthik Sindhya, Jussi Hakanen, and Kaisa Miettinen. An interactive surrogate-based method for computationally expensive multiobjective optimisation. *Journal of the Operational Research Society*, 70(6):898–914, 2019.

Franck Taillandier, Alice Micolier, and Patrick Taillandier. Li-bim (version 1.0.0), 2017.

Patrick Taillandier, Benoit Gaudou, Arnaud Grignard, Quang-Nghi Huynh, Nicolas Marilleau, Philippe Caillou, Damien Philippon, and Alexis Drogoul. Building, composing and experimenting complex spatial models with the gama platform. *GeoInformatica*, pages 1–24, 2018.

Lothar Thiele, Kaisa Miettinen, Pekka J Korhonen, and Julian Molina. A preference-based evolutionary algorithm for multi-objective optimization. *Evolutionary computation*, 17(3): 411–436, 2009.

Julien Villemonteix, Emmanuel Vazquez, and Eric Walter. An informational approach to the global optimization of expensive-to-evaluate functions. *Journal of Global Optimization*, 44 (4):509–534, 2009.

Tobias Wagner, Michael Emmerich, André Deutz, and Wolfgang Ponweiser. On expected-improvement criteria for model-based multi-objective optimization. In *International Conference on Parallel Problem Solving from Nature*, pages 718–727. Springer, 2010.

Eric Walter and Luc Pronzato. *Identification of Parametric Models from Experimental Data*. Springer Verlag, 1997.

Andrzej P Wierzbicki. The use of reference objectives in multiobjective optimization-theoretical implications and practical experience. 1979.

Andrzej P Wierzbicki. The use of reference objectives in multiobjective optimization. In *Multiple criteria decision making theory and application*, pages 468–486. Springer, 1980.

Andrew G Wilson and Zoubin Ghahramani. Copula processes. In *Advances in Neural Information Processing Systems*, pages 2460–2468, 2010.

Qingfu Zhang, Wudong Liu, E. Tsang, and B. Virginas. Expensive multiobjective optimization by MOEA/D with Gaussian process model. *IEEE Transactions on Evolutionary Computation*, 14(3):456–474, 2010.