Quantifying uncertainty with ensembles of surrogates for blackbox optimization

Charles Audet¹ · Sébastien Le Digabel¹ · Renaud Saltet¹

Received: 25 June 2021 / Accepted: 19 May 2022 / Published online: 3 July 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
Blackbox optimization tackles problems where the functions are expensive to evaluate and where no analytical information is available. In this context, a tried and tested technique is to build surrogates of the objective and the constraints in order to conduct the optimization at a cheaper computational cost. This work introduces an extension to a specific type of surrogates: ensembles of surrogates, enabling them to quantify the uncertainty on the predictions they produce. The resulting extended ensembles of surrogates behave as stochastic models and allow the use of efficient Bayesian optimization tools. The method is incorporated in the search step of the mesh adaptive direct search (MADS) algorithm to improve the exploration of the search space. Computational experiments are conducted on seven analytical problems, two multi-disciplinary optimization problems and two simulation problems. The results show that the proposed approach solves expensive simulation-based problems at a greater precision and with a lower computational effort than stochastic models.

Keywords Blackbox optimization · Derivative-free optimization · Ensembles of surrogates · Mesh adaptive direct search · Bayesian optimization

1 Introduction

This work considers the constrained optimization problem
where $\mathcal{X}$ is a subset of $\mathbb{R}^n$; $\Omega$ denotes the feasible set $\{x \in \mathcal{X} \mid c_j(x) \leq 0, j \in \{1, 2, \ldots, m\}\}$, where the functions $c_j : \mathbb{R}^n \rightarrow \mathbb{R} = \mathbb{R} \cup \{-\infty, +\infty\}$ are the constraint functions of the problem; and $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is the objective function. The set $\mathcal{X}$ contains the points that satisfy unrelaxable constraints [40]: every point explored during the optimization process must lie in $\mathcal{X}$ either because $f$ is not defined elsewhere or because a point outside $\mathcal{X}$ has no meaning in the original problem, e.g., a negative length or a probability greater than one. The set $\mathcal{X}$ typically represents bound constraints of the form $\mathcal{X} = \{x \in \mathbb{R}^n \mid \ell \leq x \leq u\}$ where $\ell$ and $u$ are vectors of $\{\mathbb{R} \cup \{-\infty\}\}^n$ and $\{\mathbb{R} \cup \{+\infty\}\}^n$, respectively. The functions $c_j, j \in \{1, 2, \ldots, m\}$, denote relaxable constraints, which means that they can be violated during the optimization process, however, the final solution must satisfy these constraints.

In blackbox optimization (BBO), no information is available on the functions $f$ and $c_j, j \in \{1, 2, \ldots, m\}$, beyond the mere values they produce, hence the blackbox designation. In particular, no derivatives can be used either because they are especially hard to estimate or because they do not exist. Designing algorithms that do not use derivatives is referred to as Derivative-Free Optimization (DFO). This context typically occurs when the functions are the results of numerical simulations. Consequently, a blackbox is assumed to be costly, i.e., one evaluation might take seconds [21], minutes [4, 48], hours [15, 66] or even days [42]. BBO consists in designing algorithms capable of finding the best possible solution to such a problem within a given budget of function evaluations. For a better understanding of the theoretical importance of the existence of derivatives in DFO, see [11]. Two reference books are available in DFO and BBO [10, 24] as well as an extended review [39].

BBO algorithms can be roughly divided in two categories: direct-search methods and methods using surrogates. Direct-search algorithms only use comparisons between points and no other information like an approximation of derivatives. This philosophy has led to fruitful algorithmic frameworks such as the state-of-the-art algorithms generalized pattern search (GPS) [65] and mesh adaptive direct search (MADS) [7], the latter is described in this article. Both frameworks lie on the search-poll paradigm [3]: the search step offers flexibility for the user to implement any method they see fit for the problem to optimize, while the poll step imposes more rigid procedures in order to guarantee convergence and explore further the surroundings of the best known solution. The DFN (derivative-free nonsmooth) algorithm [28] used for comparison in the computational experiments combines a direct-search procedure with a line-search and an exact penalty approach to address nonsmooth constrained problems for which first-order information on the functions is not available. The optimization problem consists in minimizing an objective function that penalizes the general nonlinear constraints subject to the simple bound constraints. The coordinate directions are first explored by means of a continuous search procedure and then, if the step lengths resulting from the line-search are smaller than a given threshold, another direction is explored.
The second category resorts to *surrogates*, i.e., functions that are expected to mimic the behaviour of the objective and the constraints while being significantly cheaper to evaluated. A surrogate can be a simplified and static version of the black-box that do not evolve over the optimization, or a dynamic surrogate that is based on regression or interpolation on the previously evaluated points. A dynamic surrogate is also called a model. One can then expect that minimizing a surrogate of the objective while satisfying surrogates of the constraints will lead to a promising new candidate point for the true problem. The SHEBO and SACOBRA algorithms, also used for comparison in the computational experiments, follow the steps of general surrogate model algorithms. The SHEBO (surrogate optimization of problems with hidden constraints and expensive black-box objectives) algorithm [47] proceeds as follows. First, a radial basis function (RBF) model is built on an initial experimental design. It is then used to predict the true function values at unsampled points and make sampling decisions. The model is updated whenever a new objective function evaluation has been done. Although SHEBO does not handle general constraints, it can address problems with hidden constraints, i.e., problems where the objective function might fail to return a value even when all constraints are met. To do so, another model is used to predict the feasibility of an unevulated point based on successful and failed evaluations. Three different search approaches are designed and used sequentially to find candidates points that balance the following three factors: the potential improvement of the objective function predicted by the RBF model, the expected feasibility predicted by the piecewise linear model, and the distance to already sampled points. The SACOBRA (self-adjusting constrained optimization by radial basis function approximation) algorithm [16] also uses RBF models of the objective that are updated every time a new sample point is added. It uses RBF models for the constraints too, thus natively enabling problems with general constraints. At each iteration, the surrogate problem is solved to return a new solution. A variable distance requirement factor enables to determine how close the solution can be to the points in the sample set. If it is not feasible, a repair algorithm tries to find a feasible solution in the vicinity. The new solution is then evaluated with the true blackbox and added to the sample set to update the models. Another factor enables to restrict the feasible region predicted by the surrogates of the constraints in order to compensate the lack of accuracy of the models. This factor dynamically evolves depending on the number of feasible and unfeasible sample points. SACOBRA automatically adjusts certain parameters such as the input rescaling, the constraint normalization and the distance requirement factor.

The combination of direct-search and model-based methods has proven to be fruitful. Surrogates can be incorporated either in a subproblem embedded in the search step to find candidates points, or as means of ranking candidate points of the poll step when opportunistic strategies are used.

Among the model-based methods, ensembles of models and stochastic models are two efficient techniques. Ensembles of models consist in giving each model a weight that is supposed to reflect its quality, and the combination of all weighted models yields an *aggregate model* that can be used as a standard surrogate. Stochastic models not only produce a prediction at a given point but also a measure of the uncertainty on this prediction, which is fit for Bayesian
optimization. Ensembles of models and stochastic models have both proven to be efficient but remain fundamentally separated. Using several deterministic surrogates naturally produces a deterministic aggregate model which is incompatible with Bayesian optimization. If some of the models used are stochastic though, the provided uncertainty can be exploited by all the models as in [70]. However, typical stochastic surrogates as Gaussian processes (GPs) become particularly costly to train as the training set grows. The proposed approach is to identify regions where the predictions of the models differ from each other in order to derive some form of uncertainty. The idea of using the correlation between several models to guide the optimization has already been tackled in the literature. In [30], the deviation between the predicted values of several models is used a posteriori to check the overall quality of the aggregate model. In [46], two surrogates of the objective are available: a low-fidelity one that is cheap to compute but not accurate and a high-fidelity one that is in contrast more expensive and more reliable. Then RBF models of the two surrogates are computed and the correlation between them is used to choose which surrogate, high or low-fidelity, to evaluate next. In [64], the correlation between variable fidelity co-kriging models is exploited to determine which model to evaluate. In [56], the correlation between variable fidelity multi-level generalized co-kriging models is incorporated in an extended probability of improvement.

The contribution of this work is an extension to ensembles of models when used in the form of aggregate models. For a point \( x \) of the search space \( \mathcal{X} \), the extended aggregate models produce not only a prediction \( \hat{y}(x) \), but also an uncertainty \( \hat{\sigma}(x) \), therefore imitating a stochastic model. The formulation of this uncertainty is the contribution of the article. With this new tool, in addition to obtaining the accurate predictions for which aggregate models are intended, one can get a measure of uncertainty even when stochastic models are not used in the ensemble. As way of testing, the resulting surrogate has been exploited in the search step of MADS in subproblems inspired by Bayesian optimization. The extended aggregate models are expected to help MADS find better candidate points than state-of-the-art models. The proposed approach has been tested on seven analytical problems, two multi-disciplinary optimization problems and two simulation problems. It has been compared to other versions of MADS as well as the three other BBO solvers DFN, SHEBO and SACOBRA. Results show that the proposed extended aggregate models manage to find solutions of most of the difficult real-world problems at a greater precision than the other algorithms, and with less computational effort than the competing stochastic models.

The manuscript is structured as follows. Section 2 introduces ensemble of surrogates and the Bayesian optimization framework as well as a high-level description of the MADS algorithm. Section 3 describes the quantification of uncertainty when using ensembles of models and our incorporation of the resulting extended aggregate models into the MADS algorithm. Section 4 shows the computational results on the set of problems. A concluding discussion is proposed in Sect. 5.
2 Background

This section describes the MADS algorithm and the use of surrogates in BBO, with a special focus on stochastic surrogates and ensembles of surrogates.

2.1 The MADS algorithm

MADS [7] is a direct-search algorithmic framework that follows a search-poll paradigm in which the mandatory poll step guarantees the convergence and the optional search step gives room for flexible exploration techniques. In order to ensure convergence, every candidate point must lie on a mesh defined at Iteration $k$ by

$$M^k = \{ x + \delta^k D y : x \in V^k, y \in \mathbb{N}^{n_D} \} \subset \mathbb{R}^n$$

where $V^k \subset \mathbb{R}^n$ is the cache, i.e., the set of all evaluated points up to Iteration $k$; $\delta^k > 0$ is the mesh size parameter; and $D$ is a fixed matrix of $\mathbb{R}^{n \times n_D}$, the columns of which represent $n_D$ directions of $\mathbb{R}^n$. Before the algorithm starts, $V^0$ is the set of one or more initial points provided by the user.

The search step enables to use various strategies to explore the space of variables. When the search is unsuccessful, i.e., when no better solution is found, a poll step is launched. Every candidate point generated during the poll step must lie within a frame centred around the incumbent solution $x^k$ and which size is parameterized by the poll size parameter $\Delta^k \geq \delta^k$.

At the end of an iteration, the mesh and poll size parameters are updated depending on the outcome. If the iteration is successful both are increased, and conversely, if the iteration is unsuccessful both are decreased in such a way that the set of possible directions during the poll gets richer. In this work, OrthoMADS [1] is used to deterministically generate $2^n$ orthogonal directions at the poll step. A high-level description of MADS is given Algorithm 1.

---

**Algorithm 1: The Mesh Adaptive Direct Search algorithm (MADS).**

1. **Initialization**
   - $V^0 \subset \mathbb{R}^n$: set of starting points
   - $\Delta^0 \geq \delta^0 > 0$: initial mesh and poll size parameters
   - $k \leftarrow 0$: iteration counter

2. **Search (optional)**
   - Evaluate a finite set of points included in the mesh $M^k$.
   - If the search is successful, go to 3, otherwise go to 2.

3. **Poll**
   - Evaluate a finite set of points included in the poll frame.

4. **Update parameters**
   - Update the cache $V^{k+1}$ with the newly sampled points.
   - Update the mesh and poll size parameters $\delta^{k+1}$ and $\Delta^{k+1}$.
   - Increase the iteration counter $k \leftarrow k + 1$ and go to 1.
The algorithm stops either when the poll size parameter falls under a given threshold or when the prescribed budget of function evaluations is spent. Using the Clarke nonsmooth calculus [22], one can prove that under some mild assumptions on the smoothness of the problem, the MADS algorithm globally converges to a solution satisfying local optimality conditions provided that all candidate points lie on the mesh $M^k$. The interested reader may refer to [7] and [10, Chapter 8].

2.2 Surrogates in BBO

A common approach in BBO is to use surrogates of the objective and the constraints in order to guide the optimization. A surrogate shares similarities with the true functions of the problem while being significantly cheaper. Two types of surrogates can be distinguished: static surrogates and dynamic models. A static surrogate is a simplified version of the blackbox that can be obtained for example through a simplified physics model, a coarser mesh in a finite elements simulation or a looser stopping criterion in a numerical method. Such a surrogate is fixed and does not evolve over the optimization, hence the static designation. On the other hand, a dynamic model is an interpolation or regression model that approaches the true functions by fitting previously evaluated sample points. Since it attempts to approximate the true function, the model designation is more appropriate than surrogate in this case. Common models used in BBO are polynomial response surfaces (PRS) [2, 49] and especially quadratic models [23, 25, 26], radial basis functions (RBF) [17, 37, 53–55, 63, 72, 73], support vector machines (SVM) [68], kernel smoothing (KS) [2], and Gaussian processes (GPs) [18, 51, 52, 57].

Surrogates are basically used for two purposes in BBO: finding new candidate points and ranking existing candidate points before evaluation by the true problem. One obvious approach is to find optimizers of the surrogate subproblem and use them as starting candidate points for the optimization of the true problem. However, this is not always the best approach, especially when given dynamic models that evolve when the set of sample points grows. In this case, the surrogate subproblem would not be exploited to its full potential if limited to finding starting points. The surrogate management framework (SMF) [19] establishes a more sophisticated interplay between the surrogate evaluations and the blackbox evaluations in the context of direct search methods. The basic idea is to use the surrogate subproblem to rank candidate points and then evaluate them in an opportunistic strategy with the true problem, i.e., stop evaluating as soon as an improvement is found.

Figure 1 illustrates the interaction between the optimization algorithm, the true blackbox problem and the surrogate subproblem. At a given iteration, the optimization algorithm creates a list $\mathcal{L}$ of candidate points that is entirely evaluated by the surrogate subproblem. Based on the values of $\tilde{f}$ and $\tilde{c}_j$, $j \in \{1, 2, \ldots, m\}$, the algorithm sorts the points so that the more promising ones will be evaluated first by the true problem. In doing so, the blackbox will likely not be evaluated at non promising points. If the surrogates can be updated, i.e, if they are dynamic models, the newly acquired values of the blackbox are used for the update, as illustrated in the figure.
The SMF is formally described in Algorithm 2 as in Algorithm 13.1 from [10, Chapter 13].

Algorithm 2: Surrogate management framework (SMF).

1. **Exploration using the surrogate**
   - Use the surrogate subproblem to generate a list \( \mathcal{L} \) of candidate points
   - Evaluate the true functions at points in \( \mathcal{L} \) in an opportunistic way
   - If a new incumbent solution is found, go to 3; otherwise go to 2

2. **Ranking using the surrogate**
   - Use the optimization algorithm to generate a list \( \mathcal{L} \) of candidate points
   - Use the surrogate functions to order the points in \( \mathcal{L} \)
   - Evaluate the true functions at points in \( \mathcal{L} \) in an opportunistic way

3. **Parameters update**
   - Update algorithmic parameters
   - Check stopping criteria or go to 4

4. **Model update (optional)**
   - Update the model by using the new values of the true functions obtained in 1 and 2

In the first step, the surrogate subproblem is used not only to rank the points in the list \( \mathcal{L} \) but also to create the list. Ideally, \( \mathcal{L} \) is produced rapidly and contains a limited amount of candidate points to be evaluated opportunistically by the blackbox. The second step is fundamentally different in that it generates a list of candidate points using the native processes of the optimization algorithm instead of the surrogate subproblem, thus ensuring the theoretical convergence thereof. The fourth step is reached when points were evaluated by the true problem during the first and second steps of the iteration but the stopping criteria have not been met yet. This step is not relevant if the surrogates are static, i.e., can not be updated, hence it is optional. If possible, the newly acquired blackbox values should be used to update the models in the surrogate subproblem that will expectedly become more accurate.

The SMF fits well into the search-poll paradigm since surrogates can be efficiently used either in the search step or in the poll step, or both. In the MADS context, surrogates can be used to find new candidate points during the search step. For instance, the few best solutions of the surrogate subproblem might be some promising candidate points. Several works have tackled the incorporation of surrogates in MADS like...
quadratic models [23], treed Gaussian processes [31], LOWESS models [61], hybrid models between static surrogates and dynamic models [6], or ensembles of models [12].

For extended reviews on model-based or model-assisted optimization, see [11, 71] and [39, Section 2.2].

2.2.1 Ensemble of models

In an optimization context, there is not one type of models that dominates the others [2, 30]. Even for a given problem, the best performance can be obtained with different models depending on the initial sampling [30]. A tempting strategy is to resort to several models simultaneously. This idea has proven to be efficient in several papers [2, 12, 20, 30, 49, 69, 74] in which an ensemble of models is used to build an aggregate model defined by

\[ \hat{f}(x) = \sum_{p=1}^{s} w^p \tilde{f}^p(x) \]

where \( s \geq 1 \) is the number of models; \( \tilde{f}^1, \tilde{f}^2, \ldots, \tilde{f}^s \) are \( s \) models of the objective \( f \) built from sample points; and \( w^1, w^2, \ldots, w^s \) are positive weights such that \( \sum_{p=1}^{s} w^p = 1 \). The main difficulty lies in the attribution of the weights that must reflect the quality of the models. To do so, weights can be attributed so that the error of the aggregate model will be minimal, thus introducing an optimization subproblem [2, 69]. Another approach is to compute an error metric \( E^p \) for each model \( \tilde{f}^p \) and then attribute a weight \( w^p \) that is a function of \( E^p \) [12, 30, 49]. In this type of strategy, an error metric must be chosen first. Common metrics for this purpose are statistical measures with cross-validation like root mean square error (RSME) and predicted residual sum of squares (PRESS) that take into account the gaps between the values of the models and those of the true objective. In [12], the authors propose an error metric that is not statistical but is rather a measure of a model’s capacity to rank points in the same order as the objective would do. The rationale for this latter metric is that in a BBO context a good model does not necessarily approximate well the values of the true function but is rather capable of discriminating candidate points and telling apart promising ones.

Once an error metric is chosen, weights must be attributed accordingly. For instance, in [30], the authors propose the three following options:

- \( w^p \propto E^{\text{tot}} - E^p \)
- \( w^p \propto \frac{E^p}{E^\text{min}} \)
- \( w^p \propto (E^p + \alpha E^{\text{av}})^\beta \)

where \( E^{\text{tot}} \) is the total error of all models, \( E^{\text{min}} \) is the minimal error, \( E^{\text{av}} \) is the average error, and \( \alpha < 1 \) and \( \beta < 0 \) are adjustable parameters.
2.2.2 Bayesian optimization

In general, models may be trusted in regions where the true functions have been sufficiently sampled. But the further away from these regions, the less accurate are the models. In Bayesian optimization, the objective function is interpreted as a stochastic process: an \emph{a priori} distribution $P[f]$ is assumed, then with the set of sample points $\mathcal{X}$ and a likelihood model $P[\mathcal{X} | f]$, an \emph{a posteriori} distribution $P[f | \mathcal{X}]$ is built thanks to Bayes’ rule. Consequently, for any point $x$ of the search space, a stochastic model not only produces a prediction $\mu(x)$ but also a measure of uncertainty on that prediction $\sigma(x)$. If properly exploited, this uncertainty enables to explore regions in which the model confesses to be unreliable, instead of spending the entire budget on restricted regions. This is called the balance between exploration and exploitation. Commonly used stochastic models are generalized linear models [50], Gaussian processes [52], and dynamic trees [60].

The balance between exploration and exploitation is then realized with an acquisition function. A simple example is the upper confidence bound (UCB) [59] that takes into account the most optimistic value, i.e., minimizes $\mu(x) - \kappa \sigma(x)$. A more sophisticated instance is the probability of improvement (PI) [34] which is the probability that the objective decreases from the best known value at a given point. Finally, a very popular example is the expected improvement (EI) [44] that not only takes into account the probability of decrease but also the expected amplitude thereof.

Figure 2 shows an example of Gaussian process regression—also know as \emph{kriging}—as well as the resulting expected improvement on a one-dimensional objective function. The dashed curve represents the objective $f : x \mapsto x \sin x$; the five dots are the sample points; the curve interpolating the dots is the prediction $\mu : x \mapsto \mu(x)$; an the filled area represents the 95% confidence interval given at any point $x$ by $[\mu(x) + 1.96\sigma(x), \mu(x) - 1.96\sigma(x)]$. The curve at the bottom represents—in a different scale—the expected improvement EI. The resulting candidate point maximizes EI and is indicated by the vertical dashed line.

Maximizing EI is a method introduced by Jones et al. [35] that is efficient and easy to grasp. More elaborated techniques have since been developed. Talgorn et al. [62]
propose various formulations for the surrogate subproblem that use multiple acquisition functions at once.

For an extended literature review on Bayesian optimization, see [32, 58].

3 Quantifying uncertainty with ensembles of models

Ensembles of models enable to combine several models in the hope of taking advantage of each of them. However, this technique creates an aggregate model that can produce a prediction at any point but not an uncertainty on that prediction, thus prohibiting any Bayesian-like approach. Yet, because several models are useful to describe a single function, it means that their predictions are not identical. Consequently, there should be regions in the search space where the predictions show discrepancies, e.g., one model increases while another model decreases or remains constant, thus resulting in some form of uncertainty that is not apparent in the aggregate prediction. In view of the preceding comment, the approach that is introduced consists in locally detecting and quantifying the discrepancies between models, or lack thereof, in order to produce a measure of uncertainty.

The idea that the disparity of the models’ predictions can be used is tackled in [30] where the uncertainty at a given point $x$ is produced with the standard deviation between the predictions defined by

$$
\sigma(x) = \left( \frac{\sum_{p=1}^{s} (\bar{f}^p(x) - \bar{f}(x))^2}{s - 1} \right)^{\frac{1}{2}}
$$

where $\bar{f}(x) = \sum_{p=1}^{s} \bar{f}^p(x) / s$. This metric quantifies the gaps between the values of the models, however, as it was said earlier, in a BBO context the actual values matter less than the variations of the models. For instance, the two following models possess significantly different values: $\bar{f}^1$ and $\bar{f}^2 = \bar{f}^1 / 10 + 20$. Yet, their variations are the same, i.e., when $\bar{f}^1$ increases, $\bar{f}^2$ increases too and reciprocally, and therefore they have the same optima. In this case the uncertainty shall be minimum since using either model will yield the same candidate points. Now the two following models differ: $\bar{f}^1$ and $\bar{f}^3 = -\bar{f}^1$, but in addition their variations will be opposite so that their optima will certainly be different. In this case, the uncertainty shall be maximum even though the actual standard deviation between $\bar{f}^1$ and $\bar{f}^3$ might be less than between $\bar{f}^1$ and $\bar{f}^2$. In light of this, a measure of uncertainty suited to BBO should rather take into account the variations of the models in the form of some local correlation. In [46] and [64], a correlation coefficient is built between a high-fidelity surrogate $\bar{f}^{\text{high}}$ and a low-fidelity model $\bar{f}^\text{low}$.
Quantifying uncertainty with ensembles of surrogates for...

\[ r = \frac{\sum_{j=1}^{M} (\tilde{f}_{\text{high}}(x^{(j)}) - \bar{f}_{\text{high}})(\tilde{f}_{\text{low}}(x^{(j)}) - \bar{f}_{\text{low}})}{\sqrt{\sum_{j=1}^{M} (\tilde{f}_{\text{high}}(x^{(j)}) - \bar{f}_{\text{high}})^2} \sqrt{\sum_{j=1}^{M} (\tilde{f}_{\text{low}}(x^{(j)}) - \bar{f}_{\text{low}})^2}} \]

where \( \{x^{(j)}\}_{j \in \{1,2,\ldots,M\}} \) is a set of \( M \geq n \) points sampled locally around a region of interest; and \( \bar{f}_{\text{high}} \) and \( \bar{f}_{\text{low}} \) are the average values of \( \tilde{f}_{\text{high}} \) and \( \tilde{f}_{\text{low}} \) on this set of points, respectively. For the reasons aforementioned, this quantity is more relevant than the standard deviation in BBO.

### 3.1 A new expression for the uncertainty

In the proposed approach, this idea of correlation between models is exploited to produce an expression of the uncertainty at a given point \( x \). Two alternatives are built: a smooth and a nonsmooth uncertainty. In addition each alternative is proposed in two versions: an uncertainty dedicated to the objective and another one dedicated to the constraints.

#### 3.1.1 Smooth uncertainty for the objective

The correlation between two models can be interpreted geometrically through the cosine similarity measure of two non-zero vectors. Estimating the correlation between two models \( \tilde{f}_{p} \) and \( \tilde{f}_{q} \) at a given point \( x \) can be done by measuring the cosine similarity of their respective gradients, i.e. their respective linearizations, at \( x \). In the present context, the true gradients of the models \( \tilde{f}_{p} \), \( p \in \{1, 2, \ldots, s\} \) are not always easy to obtain. Consequently, the simplex gradients [36] are used instead. The simplex gradient of a function \( f \) at point \( x \), denoted by \( \nabla S f(x) \), is the gradient of the linear model of \( f \) at \( x \). Computing the simplex gradient around a given point \( x \) requires the evaluation of \( f \) on a simplex, i.e., a set of \( n+1 \) affinely independent points, around \( x \). In this work, the correlation of two models \( \tilde{f}_{p} \) and \( \tilde{f}_{q} \) at \( x \) is measured by the cosine between their simplex gradients defined by

\[
\cos \left\langle \nabla S \tilde{f}_{p}(x), \nabla S \tilde{f}_{q}(x) \right\rangle = \frac{\nabla S \tilde{f}_{p}(x) \top \nabla S \tilde{f}_{q}(x)}{|| \nabla S \tilde{f}_{p}(x) ||_{2} \times || \nabla S \tilde{f}_{q}(x) ||_{2}}.
\]

The larger the cosine, the more correlated the models around \( x \). With this notion in mind, the uncertainty between two models \( \hat{\sigma}_{p,q} \) can be produced as an inversely proportional function of the cosine

\[
\hat{\sigma}_{p,q}(x) := \frac{1}{2} (1 - \cos \left\langle \nabla S \tilde{f}_{p}(x), \nabla S \tilde{f}_{q}(x) \right\rangle).
\]  

(2)

When the models are highly correlated, the cosine is close to 1 so that the uncertainty is close to its minimum 0. When the models are poorly correlated, the cosine is closer to 0 and the uncertainty increases to 0.5. And when the models are
anti-correlated, the cosine is close -1 so that the uncertainty reaches its maximum 1. The choice of a simplex for the simplex gradient is left at the discretion of the user. A small simplex around $x$ will yield a simplex gradient that is a good approximation of the true gradient for smooth functions, but a wider simplex will have a smoothing effect that can be appreciable with nonsmooth or noisy functions. See Appendix A for the practical construction of the simplex used in this work.

The generalization of this expression to more than two models will be described after the other versions of uncertainties are introduced.

### 3.1.2 Nonsmooth uncertainty for the objective

An alternative for the uncertainty that does not require the computation of simplex gradients is proposed. The idea is the same as the smooth alternative: measuring the similarity of the models variations close to a given point, although this time the values of the models are directly compared instead of their gradients. This alternative requires a positive spanning set of directions $D$, i.e., a set of at least $n + 1$ vectors of $\mathbb{R}^n$ such that any point of $\mathbb{R}^n$ can be written as a positive linear combination thereof [27]. For each direction $d \in D$, the measure of uncertainty shall increase if the models predict contradictory trends, i.e., if model $\tilde{f}^p$ increases from $x$ to $x + d$ while $\tilde{f}^q$ decreases, or conversely. The same should be applied to, and averaged over, all directions. The nonsmooth alternative is consequently defined by

$$\hat{\sigma}_{p,q}(x) = \frac{1}{|D|} \sum_{d \in D} \text{xor}(\tilde{f}^p(x + d) < \tilde{f}^p(x), \tilde{f}^q(x + d) < \tilde{f}^q(x))$$

(3)

where xor(·, ·) is the exclusive or logical operator. The requirement that $D$ be a positive spanning set ensures that every half-space contains one or more directions of $D$. The term $1/|D|$ scales the sum between 0 and 1 so that the uncertainty will not be influenced by the number of directions in $D$. Here again, the choice of a positive spanning set is left at the discretion of the user. Not only can the size of the directions vary as with the simplex, but also the number of directions can increase in order to explore the surroundings of $x$ better. See Appendix A for the practical construction of the positive spanning set used in this work.

### 3.1.3 Smooth uncertainty for the constraints

The expressions for the uncertainty proposed in Eqs. (2) and (3) are suited for the objective $f$ since they take into account variations of the models. However, when handling constraints, the key information is the sign of the function rather than whether it increases or not. If two models $\tilde{c}_j^p$ and $\tilde{c}_j^q$ of the same constraint $c_j$ are available, the uncertainty shall increase when one model or the other tends towards 0, and increase even more when their signs are opposite, meaning that their predictions on the feasibility are contradictory. Hence the following expression

$$\hat{\sigma}_{p,q}(x) = \text{sigm}(-\tilde{c}_j^p(x) \times \tilde{c}_j^q(x))$$

(4)
where \( \text{sigm}(\cdot) \) is the sigmoid function. It acts as an activation function that increases as the product of \( \hat{c}_j^p \) and \( \hat{c}_j^q \) decreases. This uncertainty also ranges from 0 to 1.

### 3.1.4 Nonsmooth uncertainty for the constraints

Here again, a nonsmooth alternative is proposed to the smooth uncertainty for the constraints and is the discrete version thereof. It uses the logical operator \( \text{xor} \) to indicate whether the two models \( \hat{c}_j^p \) and \( \hat{c}_j^q \) predict the same feasibility result at a given point \( x \) or not. This alternative is defined at point \( x \) by

\[
\hat{\sigma}_{p,q}(x) = \text{xor}\left(\hat{c}_j^p(x) \leq 0, \hat{c}_j^q(x) \leq 0\right).
\]

Although only the values 0 and 1 can be taken with just two models, the set of possible values becomes richer when the number of models increases, as described later. In this work, the nonsmooth alternative for the constraints has been designed to be used along with the nonsmooth alternative for the objective for the sake of coherence. However, the specific advantages of each alternative for the constraints has not been studied and is proposed as future work in the discussion.

### 3.1.5 Generalization to an arbitrary number of models

Expressions (2), (3), (4) and (5) consider two models of the objective or a constraint. Ensembles of models usually comprise more than two models though, hence the need for a general expression that can consider an arbitrary number of models. In addition, this general expression must take into account the weights \( w_p \), \( p \in \{1, 2, \ldots, s\} \), which reflect the quality of the models. Just as in the prediction defined in Eq. (1), the good models should have a strong influence in the determination of the uncertainty whereas the poor models should not. The following quantity meets those requirements

\[
\left( \sum_{p=1}^{s-1} \sum_{q=p+1}^s w_p w_q \times \hat{\sigma}_{p,q}(x) \right) / \sum_{p=1}^{s-1} \sum_{q=p+1}^s w_p w_q.
\]

The ratio in (6) considers all the possible pairs of models once. For each pair \( (p, q) \in \{1, 2, \ldots, s\}^2 \) such that \( p \neq q \), the uncertainty \( \hat{\sigma}_{p,q}(x) \) stemming from the models \( \hat{f}_p \) and \( \hat{f}_q \) at point \( x \) is weighted by the product of the corresponding weights \( w_p \) and \( w_q \). Consequently, the better the models, the more \( \hat{\sigma}_{p,q}(x) \) will weight up in the total uncertainty at point \( x \). Then the sum on all pairs of models is normalized by \( 1 / \sum_{p=1}^{s-1} \sum_{q=p+1}^s w_p w_q \) so that the result does not depend on the number of models \( s \). Since the four versions of \( \hat{\sigma}_{p,q} \) range from 0 to 1, this ratio applies to any case: objective and constraint versions, smooth and nonsmooth alternatives.

At this point the uncertainty takes into account an arbitrary number of models and also the weights as required. But the ratio in (6) is between 0 and 1 by construction, and therefore it is most likely not at the right scale for the problem at hand. A final step is to multiply by a factor \( \alpha > 0 \) that scales the ratio in a relevant way.
For this purpose, $\alpha = 10 \times \text{Var}(g(V))$ was chosen, where $g$ is either the objective or a constraint; $g(V) = \{g(x^{(1)}), g(x^{(2)}), \ldots, g(x^{(N_q)})\}$ is the set of already sampled values of the function $g$; and 10 is a factor that empirically gave better results. This choice is motivated by the fact that the problem’s scale can only be known through the true function’s values. The final expression of the uncertainty is

$$\hat{\sigma}(x) = \alpha \frac{w^T \Sigma(x) w}{w^T T w}$$

(7)

where the ratio (6) has been rewritten in a more compact form with $\Sigma(x) \in \mathbb{R}^{s \times s}$ being the upper triangular matrix such that $[\Sigma(x)]_{p,q} = \hat{\sigma}_{p,q}(x)$ if $p < q$ and 0 otherwise; $T \in \mathbb{R}^{s \times s}$ being the upper triangular matrix such that $[T]_{p,q} = 1$ if $p < q$ and 0 otherwise; and $w$ being the vector of weights $[w^1, w^2, \ldots, w^s]^T$. An aggregate model combined to this measure of uncertainty is called an extended aggregate model.

The different uncertainties are illustrated in Fig. 3. Seven points have been sampled in $[-10, 10]^2$ from an unknown function $g$ that takes two input variables $x_1$ and $x_2$. On each subfigure, the bottom surface is the aggregate prediction $\hat{g}$.

**Fig. 3** The four uncertainties on the same sample set. **a, b** correspond to the smooth and nonsmooth alternatives of the objective version, respectively (Eqs. (2) and (3)). **c, d** correspond to the smooth and nonsmooth alternatives of the constraint version, respectively (Eqs. (4) and (5)).
resulting from eleven different polynomial and RBF models, and the top surface, shifted up for readability, is the uncertainty on that prediction which becomes darker as it increases. The weights of the models \( w^p, p \in \{1, 2, \ldots, s\} \), are attributed as described in Sect. 3.2. In Fig. 3a, b the function is interpreted as the objective whereas in Fig. 3c, d the same function is interpreted as a constraint, resulting in significantly different uncertainties. In addition, in Fig. 3c, d, the points close to the assumed border of the constraint, i.e., where the prediction is close to zero, were darkened in order to better understand the uncertainty.

Comparatively, Fig. 4 shows a GP model’s prediction and uncertainty fit on the same sample points. It can be noticed that the uncertainty is lower close to the sample points and increases with the distance to them, which is expected with a GP model. The same observation cannot be made in Fig. 3, especially in Fig. 3c, d where the uncertainty is higher close to the border of the constraint. Consequently, the extended aggregate model does not interpret the data as the stochastic model does on the same sample set. The former is able to capture the richness of having multiple models in regions where sample points are manifold, unlike the latter for which the uncertainty simply increases far away from the sampled region. However, the extended aggregate model might fail to produce an informative uncertainty measure when the models predictions are alike in poorly sampled regions, thus not encouraging exploration.

### 3.2 Error metric and weight attribution

In this work, the strategy to assign weights is to compute an error metric \( \mathcal{E}^p \) for each model \( \tilde{f}^p \) and then attribute a weight \( w^p \) that is a function of \( \mathcal{E}^p \). The error metric chosen is the order error cross-validation metric (OECV) [12] mentioned earlier and denoted by \( \mathcal{E}_{OECV} \). Broadly speaking, it measures a model capacity to rank points in the same order as the actual objective would do, or, for that matter, its capacity to predict the same feasibility result as an actual constraint would do. When the metric \( \mathcal{E}_{OECV}^p \) is computed for every model \( \tilde{f}^p, p \in \{1, 2, \ldots, s\} \), the weights can be attributed. Assigning a weight of 1 to the best model and 0 to the others is the choice made in [12]. However, in the present work there must be at least two strictly
positive weights otherwise the ratio \((6)\) is a division by zero and has no meaning. The approach chosen instead is to select the \(N_{\text{best}}\) models that have the smallest error metrics and to assign to them a weight proportional to the metric. Formally, if \(I \subset \{1, 2, \ldots, s\}\) is the subset of the selected models indices, then \(w_p \propto E_{\text{tot}}^I - E_p^p\) if \(p \in I\) and \(w_p = 0\) otherwise, where \(E_{\text{tot}}^I\) is the total error of the selected models. The weights are then normalized so that \(\sum_{p \in I} w_p = 1\). If more than \(N_{\text{best}}\) models have an error metric that is equal to the best metric, all of them will be selected and be assigned an equal weight. Preliminary tests showed that \(N_{\text{best}} = 3\) and \(N_{\text{best}} = 4\) were appropriate values for the smooth and nonsmooth alternatives, respectively.

### 3.3 Incorporation into the MADS algorithm

The MADS algorithm offers an important flexibility through its search step. Many works have already included model-based subproblems (SP) into the search step as described earlier. The proposed implementation falls within this category. At each iteration, a surrogate subproblem is solved during the search step and the best solution is used as the next candidate point for the true problem. Having said that, there are many ways to design a surrogate SP.

In [62], eight different formulations for SP are proposed, denoted by SP1 through SP8. They are specifically designed to take advantage of the double information that stochastic models provide, i.e., the prediction and the uncertainty. These formulations tackle general constraints and involve the following statistical measures:

\[
\begin{align*}
\text{Expected improvement} \ [35] &: \quad EI(x) = \mathbb{E}[\max(f_{\min} - f(x), 0)] \\
\text{Probability of feasibility} \ [62] &: \quad P(x) = P[c_j(x) \leq 0, j = 1, 2, \ldots, m] \\
\text{Probability of improvement} \ [34] &: \quad PI(x) = P[f_{\min} > f(x)]
\end{align*}
\]

Then some other measures are derived from these quantities: the expected feasible improvement \(\text{EFI}(x) = EI(x)P(x)\), the probability of feasible improvement \(\text{PFI}(x) = PI(x)P(x)\) and the uncertainty on the feasibility \(\mu(x) = 4P(x)(1 - P(x))\). In the formulations, all these measures are arranged in different ways in order to highlight various properties of the sample points. The eight formulations of SP are given in Appendix B.

With stochastic models, the probability distribution at any point \(x\) is known and used to practically compute \(EI(x)\), \(P(x)\) and \(PI(x)\). However, an extended aggregate model, although inspired by the stochastic modelling philosophy, has no probabilistic foundation. More specifically, there is no cumulative distribution function available at a given point \(P[g(x) < g_0]\), for any \(g_0 \in \mathbb{R}\). Consequently, the statistical quantities defined above are not applicable as such. To address this issue, \(P\), \(PI\) and \(EI\) are replaced by substitutes \(\tilde{P}\), \(\tilde{PI}\) and \(\tilde{EI}\) that are inspired from the case when the stochastic model yields at any point \(x\) a value that follows a normal distribution \(\mathcal{N}(\hat{\gamma}(x), \hat{\sigma}^2(x))\), which is the case of GPs.

When \(\hat{c}_j\) is a stochastic model, \(\hat{c}_j(x)\) follows a normal distribution \(\mathcal{N}(\hat{\gamma}_j, \hat{\sigma}^2_j)\) where \(\hat{\gamma}_j\) and \(\hat{\sigma}_j\) are the predicted mean and standard deviation naturally produced by the model. The expression of \(P\) then becomes
Quantifying uncertainty with ensembles of surrogates for…

\[ P(x) = \prod_{j=1}^{m} \Phi\left( \frac{-\hat{y}_j}{\hat{\sigma}_j} \right) \]

where \( \Phi \) is the normal cumulative distribution function. Here the product implies that the constraints are assumed to be independent from each other, which might be incorrect but it is the best available approximation in a BBO context. When using an extended ensemble of models, although the model cannot be assumed to follow a normal distribution at a given point like an actual surrogate model, \( \hat{y}_j \) and \( \hat{\sigma}_j \) are still available through formulae (1) and (7) (here in the constraint version), respectively, and therefore the proposed adaptation is

\[ \tilde{P}(x) = \prod_{j=1}^{m} \text{sigm}_\lambda\left( \frac{-\hat{y}_j}{\hat{\sigma}_j} \right) \]

where \( \text{sigm}_\lambda \) is the sigmoid function of parameter \( \lambda \), i.e., \( \text{sigm}_\lambda(x) = \text{sigm}(\lambda x) \). Both \( \Phi \) and \( \text{sigm}_\lambda \) tend to 1 when the ratio \( -\hat{y}_j/\hat{\sigma}_j \) tends to \(+\infty\), i.e., either when \( \hat{y}_j \) is highly negative or when the uncertainty is low for whatever negative value of \( \hat{y}_j \), which in both cases means that the constraint \( c_j \) is most likely satisfied. They also tend to 0 when the ratio \( -\hat{y}_j/\hat{\sigma}_j \) tends to \(-\infty\), i.e., either when \( \hat{y}_j \) takes high values or when the uncertainty is low for whatever positive value of \( \hat{y}_j \), which in both cases means that the constraint \( c_j \) is most likely not satisfied. \( \lambda = 2 \) produces the closest approximation of \( \Phi \) but choosing other values enable to control the shape of the function. Preliminary tests showed that \( \lambda = 3 \) and \( \lambda = 1 \) were interesting values for the smooth and nonsmooth alternatives, respectively.

As for PI, when \( \hat{f} \) is a stochastic model, \( \hat{f}(x) \) follows a normal distribution \( \mathcal{N}(\hat{y}, \hat{\sigma}^2) \) and the expression becomes

\[ \text{PI}(x) = \Phi\left( \frac{f_{\text{min}} - \hat{y}}{\hat{\sigma}} \right) \]

where \( f_{\text{min}} \) is the best know value of the objective. The proposed alternative when using an extended aggregate model instead of a stochastic model is

\[ \tilde{\text{PI}}(x) = \text{sigm}_\lambda\left( \frac{f_{\text{min}} - \hat{y}}{\hat{\sigma}} \right) \]

where \( \hat{y} \) and \( \hat{\sigma} \) are drawn from (1) and (7) (in the objective version), respectively. Here again, \( \Phi \) and \( \text{sigm}_\lambda \) have the same behaviour but the parameter \( \lambda \) enables to control the shape of PI. The values chosen for the smooth and nonsmooth alternatives are \( \lambda = 0.1 \) and \( \lambda = 0.5 \), respectively.

Finally, when \( \hat{f} \) is a stochastic model, \( \hat{f}(x) \) follows a normal distribution \( \mathcal{N}(\hat{y}, \hat{\sigma}^2) \) and the expression of EI becomes

\[ \text{EI}(x) = (f_{\text{min}} - \hat{y})\Phi\left( \frac{f_{\text{min}} - \hat{y}}{\hat{\sigma}} \right) + \hat{\sigma}\phi\left( \frac{f_{\text{min}} - \hat{y}}{\hat{\sigma}} \right) \]

where \( \phi \) is the normal density function. This expression is intimately related to the Gaussian aspect of the model an therefore is not \textit{a priori} suited for non-Gaussian
models, let alone models that are not truly stochastic. However, it possesses interesting properties that are independent from the Gaussian nature of the model and that can be seen as essential to the very notion of expected improvement. Firstly, for a fixed $\hat{\sigma}$, EI decreases in $\hat{y}$, tends to 0 when $\hat{y}$ tends to $+\infty$ and is almost proportional to $\hat{y}$ when $\hat{y}$ tends to $-\infty$, which is judicious in a minimization context. In addition, when $\hat{y}$ gets closer to $f_{\min}$, EI becomes almost proportional to $\hat{\sigma}$, meaning that when the prediction does not improve the objective (i.e., $\hat{y} \approx f_{\min}$) the expected improvement mostly relies on the uncertainty. Then, for a fixed $\hat{f}$, EI increases in $\hat{\sigma}$ and is almost proportional to $\hat{\sigma}$ when $\hat{\sigma}$ tends to $+\infty$, which is sensible since for a given prediction the higher the uncertainty, the larger the potential improvement. Finally, when $\hat{\sigma}$ tends to 0, the behavior of EI depends on the values of $f_{\min}$ and $\hat{y}$: if $f_{\min} \geq \hat{y}$, then EI tends to $f_{\min} - \hat{y}$, and if $f_{\min} < \hat{y}$, then EI tends to 0, meaning that when the uncertainty is low, EI mostly relies on the comparison between $f_{\min}$ and the prediction $\hat{y}$. Taking into account these considerations, the proposed adaptation for EI when an extended aggregate model is used instead of a stochastic model is very close to the actual EI and is defined by

$$\hat{\text{EI}}(x) = (f_{\min} - \hat{y}) \text{sign} \left( \frac{f_{\min} - \hat{y}}{\hat{\sigma}} \right) + \hat{\sigma} \gamma \left( \frac{f_{\min} - \hat{y}}{\hat{\sigma}} \right)$$

where $\gamma(t) = e^{-t^2/2}$. Here $\lambda = 1$ was chosen. The functions $\phi$ and $\gamma$ only differ by a factor $1/\sqrt{2\pi}$ and the reason for choosing $\gamma$ instead of $\phi$ is that this factor is no more justified without an actual stochastic model that produces normal distributions. Moreover, preliminary tests showed that the proposed uncertainty seemed on average lower than the uncertainty provided by a kriging model. The terms $P$, $PI$ and $EI$ were then simply replaced by $\hat{P}$, $\hat{PI}$ and $\hat{EI}$ in the formulations of SP given in Appendix B.

**Algorithm 3:** The MADS algorithm with aggregate models.

0. **Initialization**

- $SP \in \{SP_1, SP_2, \ldots, SP_8\}$: surrogate subproblem formulation
- $\tilde{g}^1, \tilde{g}^2, \ldots, \tilde{g}^s$: choice of models for the objective and the constraints
- $V^0 \subseteq \mathbb{R}^n$: set of starting points
- $\Delta^0 > \delta^0 > 0$: initial mesh and poll size parameters
- $k \leftarrow 0$: iteration counter

1. **Models and weights update**

   - Build or update $f^1, f^2, \ldots, f^s$ using the values of $f$ in $V^k$
   - Update $w_1^1, w_2^1, \ldots, w_s^1$ using the OECV metric for the objective
   - Build or update $\tilde{c}_j^1, \tilde{c}_j^2, \ldots, \tilde{c}_j^s$ using the values of $c_j$ in $V^k$, for $j \in \{1, 2, \ldots, m\}$
   - Update $w_1^j, w_2^j, \ldots, w_s^j$ using the OECV metric for the constraints, for $j \in \{1, 2, \ldots, m\}$

2. **Search**

   - Solve SP to find the best solution $x_{SP}^k$
   - Project $x_{SP}^k$ onto the mesh $M^k$
   - Evaluate the resulting point with the true problem

3. **Standard poll**

4. **Standard parameters update**
Algorithm 3 summarizes the incorporation of extended aggregate models in MADS. First, one formulation must be chosen among \{SP1, SP2, \ldots, SP8\}. At iteration \(k\), the best solution found for SP, denoted by \(x_{sp}^k\), is projected onto the mesh \(M^k\) and is used as the candidate point of the search step. The freshly evaluated points are then added to the cache \(V^k\) so that the models and the weights will be adjusted accordingly before iteration \(k + 1\) begins. The resulting algorithm benefits from the convergence results of MADS since all the candidate points lie on the mesh \(M^k\).

4 Computational results

The proposed approach has been tested on seven analytical problems; two multidisciplinary optimization (MDO) applications: the aircraft range problem and the simplified wing problem; and two simulation problems: solar 1 and styrene. Version 4 of the NOMAD software [13, 14] was used on a PC Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz on Linux. The aggregate models used the default selection of eighteen models comprised of polynomial response surfaces of various degrees, kernel smoothing, modified radial basis functions as in [12], and k-nearest neighbours with \(k=1\). The proposed extended aggregate models have been extensively compared to quadratic models within the MADS framework. The latter are typically used in model-based or model-assisted approaches because they provide good local approximations and are relatively cheap to build. They can be considered as the archetypal models in BBO and are the default choice when activating surrogate models in NOMAD. The proposed method has also been compared to kriging models. However, due to the long running times they required, these models were only tested on the analytical problems and the aircraft range problem. Both quadratic and kriging models were readily available in NOMAD. Every instance of MADS in this work uses the last direction of success at the poll step [7]. Three other BBO solvers have been included in this study, one that does not use models: DFN [28]; and two that use models: SHEBO [47] and SACOBRA [16]. DFN and SACOBRA innately handle general constraints, but SHEBO is designed for problems with hidden constraints only. When using SHEBO, the problems were adapted so that a violated general constraint will be interpreted as a hidden constraint.

The interpretation of the results mostly relies on data profiles [45] which enable to compare multiple solvers on a given set of problems. Broadly speaking, the data profile of a solver indicates the proportion of problems solved to a given tolerance within a prescribed number of evaluations. More precisely, for a set of problems \(P\), the data profile \(d_s\) of some solver \(s\) is defined as a function of an integer \(\kappa\) defined by

\[
d_s(\kappa) = \frac{1}{|P|} \left| \left\{ p \in P : t_{p,s} \leq \kappa(n_p + 1) \right\} \right|
\]

where \(n_p\) is the dimension of problem \(p\) and \(t_{p,s}\) is the number of evaluations required to satisfy a specific convergence criterion. The reason why groups of \(n_p + 1\)
evaluations, also known as simplex gradient estimates, are taken into account rather than the sheer number of evaluations is that the dimension of the problems should not influence the results. The convergence criterion considered is the following

\[ \frac{f_p^w - f_{p,s}}{f_p^w - f_p^*} \geq 1 - \tau \]

where, for a given problem \( p \), \( f_p^w \) is either the value of the objective at the starting point when feasible, or the worst feasible value found by all solvers; \( f_{p,s} \) is the best solution found by solver \( s \) to date; \( f_p^* \) is the best solution found by all solvers; and \( 0 < \tau < 1 \) is a given precision. In summary, \( d_s(\kappa) \) gives the proportion of problems for which solver \( s \) managed to approach the best solution found to the precision \( \tau \) within \( \kappa \) simplex gradient estimates, thus enabling to see the evolution over the entire optimization. For a fixed \( \kappa \) and two solvers \( s_1 \) and \( s_2 \), if \( d_{s_1}(\kappa) \geq d_{s_2}(\kappa) \), then solver \( s_1 \) has solved, i.e., reached the convergence criterion for, more problems than solver \( s_2 \) to the precision \( \tau \). Subsequently, a solver is better to the precision \( \tau \) if the curve of its data profile is above the others. Nonetheless, a solver \( s_1 \) may be better than another solver \( s_2 \) for small precisions, meaning that it solves more problems from afar, but be worse for higher precisions, meaning that solver \( s_2 \) performs very well on the few problems to which it manages to find a good solution.

Since SHEBO does not take a single starting point as an input, it is not fit for comparison to the other algorithms through data profiles. In addition, SHEBO and SACOBRA were significantly longer to run than MADS and DFN, thus preventing from running them as many times as data profiles would require. Consequently, Sect. 4.6 provides tabular comparisons with SHEBO and SACOBRA.

Unless otherwise specified, due to the randomness contained in MADS, every version thereof was run four times on each problem with a different seed for the random generator each time. Similarly, DFN enables to choose between the Halton and Sobol sequences so the two were tested on each problem and taken into account in the data profiles.

### 4.1 Analytical problems

The seven analytical problems are listed in Table 1 with the number of variables \( n \) and constraints \( m \), whether the variables are bounded or not, and the number of starting points used. By taking into account the additional starting points for problems HS83, HS114 and MAD6, the total number of problems is fifteen. The evaluation budget is 1200(\( n + 1 \)).

As in [62], the eight SP formulations were compared, and the following values were tested for the parameter \( \lambda \) when applicable: \( \{0, 0.01, 0.1, 1\} \), thus resulting in twenty-three distinct formulations. When this parameter is involved in a formulation, it is denoted as a subscript, e.g., \( \text{SP2}_{0.1} \). The purpose here is not to exhaustively compare the formulations with each other but rather to identify the bests formulations and compare their performances to the existing versions of NOMAD and to DFN.
The formulations with extended aggregate models were compared to NOMAD without any search step, referred to as “no search”, NOMAD with a search step involving the minimization of quadratic models, referred to as “quad search”, and DFN. It turns out that all formulations perform better than no search, confirming that the approach is valid and does not “waste” evaluations. However, quad search is most of the times as good as, and sometimes better than, the proposed approach, confirming the effectiveness of quadratic models on analytical functions. DFN presents heterogeneous performances. It found good solutions for three problems but performed poorly on the others, hence the low overall performance.

The best formulation found for the smooth alternative in terms of the proportion of problems solved within the budget is SP$_{30}$ defined by

$$
\min_{x \in \mathcal{X}} -\text{EI}(x)
\quad \text{s.t. } \hat{c}_j(x) \leq 0, \ j = 1, 2, \ldots, m
$$

As for the nonsmooth alternative, the best formulation is SP$_{5,0.01}$ which consists in maximizing $\text{EI}(x) + 0.01\hat{\sigma}_f(x)$. Figure 5 shows the data profiles of the five following algorithms: no search, quad search, SP$_{30}$ with smooth uncertainty, SP$_{5,0.01}$ with nonsmooth uncertainty, and DFN at variable tolerance: $\tau = 10^{-1}$, $\tau = 10^{-3}$, $\tau = 10^{-5}$ and $\tau = 10^{-7}$.

The profiles show that the performances of SP3 and SP5 are close to that of quad search for high tolerance. However, quad search becomes significantly better for low tolerance ($\tau = 10^{-7}$). These first results show that the extended aggregate models combined with the formulations manage to find good solutions as efficiently as quad search. However, quadratic models do so slightly faster in terms of the number of evaluations, and more importantly, they are especially accurate on analytical problems, thus resulting in superior performances at low tolerance.

Since extended aggregate models are meant to mimic and therefore supersede actual stochastic models, the comparison to the available kriging models in NOMAD is most appropriate. In [62], the authors recommend SP1 and SP2 with large values of $\lambda$ when using stochastic models. For that reason, these two formulations have been tested with $\lambda = 0.1$ and $\lambda = 1$. On this set of problems, SP$_{2,0.1}$ turns out to be the best formulation. The latter was therefore tested against SP$_{3,0}$ with

| # | Name | Source | $n$ | $m$ | Bounds | # starting points |
|---|---|---|---|---|---|---|
| 1 | G2 | [9] | 10 | 2 | Yes | 1 |
| 2 | HS19 | [33] | 2 | 2 | Yes | 1 |
| 3 | HS83 | [33] | 5 | 6 | Yes | 4 |
| 4 | HS114 | [41] | 9 | 4 | Yes | 3 |
| 5 | MAD6 | [41] | 5 | 7 | No | 4 |
| 6 | PENTAGON | [41] | 6 | 15 | No | 1 |
| 7 | SNAKE | [8] | 2 | 2 | No | 1 |
smooth uncertainty and $\text{SP}^5_{0.01}$ with nonsmooth uncertainty, that is the best formulations seen above. The resulting data profiles in Fig. 6 show that on the present set of problems extended aggregate models are as good as, or better than, the kriging alternative depending on the tolerance. In addition, due to the inversion of a covariance matrix that grows with the size of the sample set, the kriging models typically take minutes to tens of minutes to solve one problem, which is prohibitive when optimizing cheap functions with large budgets of evaluations. In comparison, the proposed approach and quad search typically take minutes and no search and DFN take seconds. As a result, replacing classic kriging models by extended aggregate models does not harm the performances in terms of the number of evaluations on the set of analytical problems, while improving the real optimization time. Overall, based on the present results, quad search must be favoured on cheap analytical problems.

### 4.2 The aircraft range MDO problem

The aircraft range problem [38] is a multi-disciplinary optimization problem (MDO), meaning that it is combined of several interconnected disciplines so that the input of one discipline is the output of the others, and several cycles between them are necessary to stabilize the result. The aircraft range problem aims at

![Fig. 5 Data profiles. no search vs. quad search vs. SP3, with smooth uncertainty vs. SP5, with nonsmooth uncertainty versus DFN on analytical problems](image-url)
maximizing the range of a supersonic business jet by considering aerodynamics, structure, and propulsion, under constraints of engine, performance and structure. The problem has \( n = 10 \) variables and \( m = 10 \) constraints. It is nonsmooth and has several local optima.

In order to test the algorithms truthfully, ten starting points were sampled with Latin hypercube sampling [43]. Here again, the 23 formulations were run with the two uncertainty alternatives in order to identify the best combinations and analyze their performances. The evaluation budget is \( 600(n + 1) \). The best formulation with the smooth alternative is \( \text{SP}^{0.1} \) defined by

\[
\min_{x \in \mathcal{X}} -\text{EI}(x) - 0.1\hat{\sigma}(x)
\]

s.t. \( \hat{c}_j(x) - 0.1\hat{\sigma}(x) \leq 0, \ j = 1, 2, \ldots, m \) \hspace{1cm} (SP3-EI\sigma)

and the best formulation identified with the nonsmooth alternative is \( \text{SP}^{8} \) which consists in maximizing \( \text{PFI}(x) \).

This time, each of the aforementioned formulations is compared individually not only to quad search, but also to the same formulation with kriging models instead. This choice is motivated by the fact that the aircraft range problem is a real-world simulation-based problem that takes significantly more time to compute than the previous analytical problems, and therefore computing expensive

---

**Fig. 6** Data profiles. \( \text{SP}^{0.1} \) with smooth uncertainty vs. \( \text{SP}^{0.01} \) with nonsmooth uncertainty versus \( \text{SP}^{0.1} \) with kriging models on analytical problems.
kriging models might be worth the trade-off. In this section, because we are more interested in the comparison to quad search and because it performs better than no search, the latter is not shown in the interest of readability. DFN neither because it performed poorly on this problem and its data profiles were flat. Results of these two algorithms are available in Sect. 4.6. Fig. 7a, b show the data profiles of SP30.1 with smooth uncertainty with tolerances $\tau = 10^{-7}$ and $\tau = 10^{-9}$, respectively; and Fig. 7c, d show the data profiles of SP8 with nonsmooth uncertainty with tolerances $\tau = 10^{-7}$ and $\tau = 10^{-9}$, respectively. All the algorithms presented equivalent performances for high tolerances ($\tau = 10^{-1}$ and $\tau = 10^{-3}$) and consequently the data profiles did not reveal significant difference between the solvers with tolerances under $\tau = 10^{-5}$, meaning that all the formulations manage to reach a good solution.

For every above-mentioned formulation, the performance is relatively comparable to that of quad search for tolerance $\tau = 10^{-7}$. However, for $\tau = 10^{-9}$, the proposed extended aggregate models coupled with the right formulations turn out to be significantly better than both quad search and their kriging counterpart, i.e., the same formulations with kriging models instead. It can be noticed that the smooth alternative eventually solves more problem, but the nonsmooth one solves problems faster. In Fig. 7b, SP30.1 with smooth uncertainty solves 87.5% of the problems at tolerance $\tau = 10^{-9}$ with the allocated budget, while in Fig. 7d SP8 with nonsmooth uncertainty only solves 77.5% of the problems at the same tolerance. However, the latter takes only $300(n+1)$ evaluations to do so, while the former has only solved 50% of the problems after the same number of evaluations. This trend has been observed on this problem with all the other formulations not presented in this paper. The higher achievements of the smooth uncertainty over the long run may be attributed to its intrinsically rich range of values, while the relative rapidity of the nonsmooth uncertainty may be the result of a more aggressive behaviour that helps find a good solution faster.

It could be argued that comparing kriging models within the best formulations found for extended aggregate models, i.e., SP30.1 and SP8, is not fair since the former might perform poorly on these formulations but yield better results on others. In [62], the authors use stochastic models and recommend SP5, SP6 and SP7 with small values of $\lambda$ for expensive simulation-based problems as is the case with the aircraft range problem. Accordingly, those three formulations were tested with kriging models and $\lambda = 0.01$. On this particular problem, SP5 turns out to be the best formulation. Consequently, extended aggregate models were compared to kriging models on SP5. Fig. 8a, b show the data profiles of SP50.01 with smooth and nonsmooth alternatives, and kriging models. The latter perform indeed better than the extended aggregate models on SP50.01 at tolerances $\tau = 10^{-7}$ and $\tau = 10^{-9}$, meaning that the best formulations with real stochastic models are not necessarily the same than the best ones with the proposed extended aggregate models.

As a result, the fair comparison is not between kriging models and extended aggregate models within the same formulation, but rather between each type of models coupled with its best formulation, that is SP30.1 with the smooth alternative, SP8 with the nonsmooth alternative, and SP5 with kriging models. Figures 8c, d show the data profiles of the above combinations and in addition quad search
as a reference. At tolerance $\tau = 10^{-7}$, the performances are alike but at tolerance $\tau = 10^{-9}$, the extended aggregate models coupled with the suitable formulations solve more problems, faster than kriging models with their own appropriate formulation, and faster than quad search. To tolerance $\tau = 10^{-9}$, the smooth and nonsmooth alternatives solve 87.5% and 75% of the problems, respectively, while kriging models and quad search solve 72.5% and 57.5% of the problems, respectively.

Since the different approaches require some non negligible amount of internal computation, it is appropriate to compare them not only in terms of the number of evaluations, but also in terms of the total real optimization time. It was highlighted in Sect. 4.1 that kriging models were prohibitively long to train when optimizing cheap analytical problems. The same question is addressed more thoroughly with the aircraft range problem. Figure 9 shows the time data profiles of formulation $\text{SP}_{30.1}$ with the smooth alternative, formulation $\text{SP}8$ with the nonsmooth alternative, and formulation $\text{SP}_{50.01}$ with kriging models. In a time data profile, the proportion of problems solved is not a function of the number of evaluations anymore but a function of the real computation time instead. The profiles suggest that even with a
more expensive, real-world problem, kriging models are especially long to train. At tolerance $\tau = 10^{-9}$, SP$_{3.0.1}$ with the smooth alternative solves 87.5% of the problems within 95 seconds and SP$_{0.1}$ with the nonsmooth alternative solves 75% of the problems within 73 seconds. After the same amount of time, SP5 with kriging models has solved less than 10% of the problems, and requires 728 seconds to solve 65% of the problems. Some instances even require more than 1500 seconds. Based on the present results, the proposed extended aggregate models constitute a cheaper and more efficient alternative to kriging models.

4.3 The simplified wing problem

The simplified wing problem [67] is also an MDO problem. It aims at minimizing the drag of a wing by taking into account aerodynamics and structure. This problem is smooth but has several local optima. It has $n = 7$ bounded variables and $m = 3$ constraints.
Ten starting point have been randomly sampled in the bounded space of variables with Latin hypercube sampling. The evaluation budget is $600(n + 1)$. Unlike the previous problems and for the rest of the study, the kriging models have not been tested because of their heavy computational cost, and not all the formulations have been tested but rather a subset of the most promising ones found on the aircraft range problem: SP1, SP2, SP3, SP7 and SP8 for the smooth alternative; and SP2, SP3, SP4, SP6 and SP8 for the nonsmooth alternative. Among the above formulations, the best ones found for this problem with the smooth and nonsmooth alternatives are SP8 and SP4, respectively. SP4 consists in maximizing $\text{EFI}(x)$. Figure 10 shows the data profiles of no search, quad search, SP8 with smooth uncertainty, SP4 with nonsmooth uncertainty, and DFN at tolerances $\tau = 10^{-1}$ and $\tau = 10^{-3}$. Lower tolerances resulted in data profiles that were too flat, especially because of the high sensitivity to the seed of NOMAD on this problem.

On this MDO problem, the extended aggregated models perform better than no search, quad search and DFN. SP4 with the smooth uncertainty is especially good at tolerance $\tau = 10^{-3}$, solving 60% of problems while quad search only solves 20%. As with the aircraft range problem, the nonsmooth alternative solves problems faster than the smooth one. However, on the simplified wing problem the smooth alternative is by far the most efficient eventually. no search and DFN, that do not make use of models, do not find accurate solutions on many runs.

### 4.4 The solar 1 problem

The solar 1 problem is part of a set of nine concentrated solar power simulation problems [29] that serve as a benchmark for blackbox optimization solvers, available at github.com/bbopt/solar. The solar 1 problem aims at maximizing the heliostat field energy output throughout one day under constraints of field geometry and cost. The problem is noisy and has several local optima. It has $n = 9$ variables, among which one is discrete, and $m = 5$ constraints. The solar 1 problem has the specificity of having two additional adjustable parameters: the seed and the number of replications. Since the problem contains some stochasticity, the seed for the
random generator can be chosen. It is not the same seed as the one of NOMAD. Two instances of solar 1 with two different seeds are considered as two different problems in this work. The other parameter enables to replicate the evaluations in order to smooth the problem and compensate the noise. It was fixed to ten in the experiments.

In order to generate several problem instances, fifteen different seeds are chosen for the problem—not for NOMAD—instead of multiple starting points. Because the problem is especially expensive, NOMAD has been tested with only one seed, and not four, in order to reduce the number of optimization runs. The evaluation budget is $800(n+1)$. As with the previous problem, only the best formulations from the aircraft range problem have been tested. The best one among them is SP8 for both smooth and nonsmooth alternatives. They manage to yield better results than no search, however, quad search is clearly the best algorithm on this problem.

Figure 11 shows the data profiles of SP8 with smooth and nonsmooth alternatives along with quad search. DFN and no search are not represent because they do not manage to solve one problem even at the largest tolerance. At tolerance $\tau = 10^{-1}$, the smooth alternative manages to solve as many problems as quad search. However, for higher tolerances, the latter is by far the best alternative. The poor performance of the extended aggregate models can be attributed to the stochasticity of the problem that importantly deteriorates the models. Nonetheless, it also impacts quad search. The superior performance of the latter is also due to the nature of the constraints: three out of five are linear, and one is cubic, which gives a non negligible advantage to quadratic models.

4.5 The styrene problem

The styrene problem is a chemical engineering simulator for styrene production described in [5] and available at github.com/bbopt/styrene. It aims at maximizing the net present value of the styrene production process under structural, chemical and financial constraints with variables comprised of physical parameters and structure. The problem is deterministic but nonsmooth and with omnipresent
hidden constraints, i.e., the simulation often fails to return a value even when all constraints are met. A random sampling resulted in almost 60% of failures in \cite{31}. In addition, four constraints are binary. The problem has $n = 8$ bounded variables and $m = 11$ constraints.

Feasible regions may be especially hard to find on this problem. Consequently, twelve starting points were generated in three relatively easy regions. The evaluation budget is $600(n + 1)$. Like the previous problem solar 1, the same subset of the most promising formulations found on the aircraft range problem has been tested. The best formulations are SPI$_1$ with the smooth alternative and SP3$_{0.01}$ with the nonsmooth alternative.

Figure 12 shows the data profiles of no search, quad search, SPI$_1$ with the smooth uncertainty and SP3$_{0.01}$ with the nonsmooth uncertainty at tolerance $\tau = 10^{-1}$ and $\tau = 10^{-2}$. On this problem, the extended aggregate models perform better than both no search and quad search. Besides, unlike all the others problems the latter yields worse results than no search due to the binary constraints. DFN struggles to find feasible solutions and the results are not presented.

![Fig. 11 Data profiles. quad search versus SP8 with smooth uncertainty versus SP8 with nonsmooth uncertainty on solar 1](image-url)
4.6 Analysis of the computational cost

In order to have a better idea of the computational time required by all solvers on the real-world problems, and to include the SHEBO and SACOBRA solvers in the analysis, the results are presented in a different manner in this section. SHEBO and SACOBRA require significantly more time to run than MADS and DFN, therefore they have been run only once on each problem, instead of multiple times with different starting points and seeds. In addition, SHEBO does not take a starting point as an input, thereby making the comparison through data profiles impossible. This section shows the time and evaluations needed to reach the best value and to terminate, for each solver and each problem. More precisely, the best value found for a given optimization run is denoted by $f^*$, the real time needed to reach $f^*$ is denoted by $t^*$ (in minutes), and the number of evaluations needed to reach $f^*$ is denoted by $k^*$. Table 2 shows the median $f^*$, $t^*$ and $k^*$ on each problem considering all starting points and seeds, denoted by $f^*_m$, $t^*_m$ and $k^*_m$, respectively, as well as the best $f^*$ found on all runs, denoted by $f^*_{\text{best}}$, and the total real time needed for all runs, denoted by $t_{\text{tot}}$. Showing the median values enables to draw conclusions not only based on the best values, which could be misleading if only a few runs get exceptional values. For every real-world problem $f^*_m$, $t^*_m$, $k^*_m$, $f^*_{\text{best}}$ and $t_{\text{tot}}$ are shown for no search, quad search, extended models with the best formulation found with the smooth uncertainty (new approach), extended models with the best formulation found with the nonsmooth uncertainty (new approach), DFN, SHEBO and SACOBRA. For each problem, the best value of $f^*_m$ and $f^*_{\text{best}}$ is underlined. Since SHEBO and SACOBRA have been run only once on each problem, the values corresponding to these two solvers relate to a single run, and therefore $f^*_m$ is equal to $f^*_{\text{best}}$.

![Fig. 12 Data profiles. no search versus quad search versus SP1 with smooth uncertainty vs. SP3 with nonsmooth uncertainty on styrene](image)

First of all, Table 2 shows that the real time ($t_{\text{tot}}$ column) needed by SHEBO and SACOBRA to complete a single run is in most cases comparable to, or even greater than the time taken by MADS, including the proposed approach, to perform 10 to 48 runs depending on the problem. This is mostly explained by the cost of the RBF models in SHEBO and SACOBRA: as the number of sample points grows, the models become very expensive to update. The DFN algorithm does not require
Table 2 Results of all algorithms

| Algorithm    | Aircraft range (40 runs except for SHEBO and SACOBRA) | Simplified wing (40 runs except for SHEBO and SACOBRA) |
|--------------|--------------------------------------------------------|--------------------------------------------------------|
|              | $f^*_m$ | $r^*_m$ | $k^*_m$ | $f^*_\text{best}$ | $t_{\text{tot}}$ | $f^*_m$ | $r^*_m$ | $k^*_m$ | $f^*_\text{best}$ | $t_{\text{tot}}$ |
| No search    | -3964.204696 | 1 | 4140 | -3964.204700 | 118 | -16.4059 | 3 | 2405 | -16.6119 | 397 |
| Quad search  | -3964.204698 | 5 | 3064 | -3964.204701 | 834 | -16.5808 | 12 | 1291 | -16.6120 | 2181 |
| Smooth       | -3964.204699 | 10 | 3218 | -3964.204701 | 1836 | -16.6063 | 4 | 2115 | -16.6120 | 921 |
| Nonsmooth    | -3964.204699 | 9 | 2555 | -3964.204700 | 1614 | -16.5924 | 2 | 836 | -16.6120 | 393 |
| DFN          | -1749.256329 | 0.1 | 73 | -3143.101404 | 18 | -15.0774 | 3 | 1134 | -16.6112 | 588 |
| SHEBO        | -784 | 4404 | -3723.074357 | 3210 | -16.5501 | 356 |
| SACOBRA      | -1445 | 2576 | -3964.200535 | 5557 | -16.6112 | 49 |
| Solar 1      | -508184.0 | 190 | 4064 | -66088.83 | 2384 | -29305150 | 12 | 1541 | -33613200 | 3611 |
| Quad search  | -7236232.2 | 510 | 3795 | -835124.4 | 6837 | -29301200 | 41 | 1675 | -33000800 | 9750 |
| Smooth       | -677669.7 | 305 | 3688 | -849192.5 | 6303 | -32704300 | 49 | 1809 | -33705600 | 19015 |
| Nonsmooth    | -658837.2 | 224 | 3359 | -805799.5 | 6190 | -32235250 | 48 | 1647 | -33697400 | 16086 |
| DFN          | -332582.8 | 68 | 1448 | -391849.8 | 735 | -22851550 | 0.3 | 38 | -28517600 | 49 |
| SHEBO        | -2865 | 6190 | -815086.2 | 4475 | -741 | 5365 | -32873400 | 774 |
| SACOBRA      | -1093 | 2367 | -487364.4 | 6257 | -11846 | 4275 | -27958100 | 12868 |
models and therefore avoids this pitfall, and all the MADS versions presented here use inexpensive models, as the results demonstrate.

Over all four problems, the performances of SHEBO and SACOBRA compared to that of MADS vary markedly. On the aircraft range problem, the proposed approach not only finds a better $f_{\text{best}}^*$ than SHEBO and SACOBRA but also a better $f_{\text{m}}^*$ after only a fraction of the time per run. It should be noted though that while SHEBO’s $f_{\text{best}}^*$ is off the mark, SACOBRA’s is relatively close to that of the MADS solvers. However, the time required by SACOBRA is prohibitive. On the simplified wing problem, the conclusions are similar in terms of $f_{\text{best}}^*$ and $f_{\text{m}}^*$ but this time SHEBO performs better than SACOBRA, and the time required by SHEBO is less prohibitive than previously. On this problem, SACOBRA does not find an interesting region to explore and does not manage to find a better solution after the early stage of the exploration, hence the poor solution. On the solar 1 problem, SHEBO has a better $f_{\text{best}}^*$ than the $f_{\text{m}}^*$ of MADS and DFN, and even than the $f_{\text{best}}^*$ of the extended models with nonsmooth uncertainty. However, the extended aggregate models with the smooth uncertainty find a better $f_{\text{best}}^*$ over all 10 runs after only 1.5 time more real time than the single run of SHEBO. As for SACOBRA, it is not competitive on this problem since it does not find a better value than any $f_{\text{best}}^*$ of the MADS versions. Finally, on the styrene problem, SHEBO has again a better $f_{\text{best}}^*$ than the $f_{\text{m}}^*$ of MADS and DFN, but this time after a significantly shorter $t_{\text{tot}}$ than MADS, and especially than the proposed approach, making it an interesting alternative on this problem. However, all MADS algorithms find a better $f_{\text{best}}^*$ over all 48 runs, and especially the proposed approach.

Overall, on the present experiments, the algorithms with models (all MADS versions except no search, SHEBO and SACOBRA) perform better than DFN on three problems out of four (DFN performs better than SHEBO and SACOBRA on the simplified wing problem). The algorithms using models, including the proposed approach, also require more time to run due to model construction, which is corroborated by the smaller running times of no search compared to the other MADS versions, but the proposed approach finds the absolute best value on all four problems, and the median best value on three problems out of four. Based on the present results, one can infer than a single run with SHEBO guarantees a decent value, only it demands a much larger computation time per run which is better invested running the proposed approach multiple times. As for SACOBRA, it performed well on the aircraft range problem but not better than the extended aggregate models, and was not competitive on the other problems. Therefore it cannot be recommended over the new approach. Regarding the performance of quad search, Table 2 shows that in three problems out of four, the proposed approach finds a better median $f^*$ than quad search with both smooth and nonsmooth uncertainties, confirming the observations made in previous sections. It also shows that the best $f^*$ found by the extended aggregate models are better than that of quad search on all four problems with the smooth uncertainty, and on two out of four problems with the nonsmooth uncertainty. In other words, on the single problem where quad search was declared better than the proposed approach, that is solar 1, the extended aggregate models with the smooth uncertainty still manage to find the best objective value over all ten runs, and within a smaller total time.
One final remark can be made about the real time required by the extended aggregate models compared to the quadratic models. The proposed approach is not always longer than quad search, both in terms of median time and total time. The quadratic models indeed tend to result in very long runs when they find a good basin of solutions. Conversely, they result in very short runs when they do not manage to find a good solution, depending on the starting point.

5 Discussion

This work proposes an extension to ensembles of models that enables to compute an uncertainty at any given point. The resulting extended aggregate models behave like stochastic models, i.e., they produce at any given point $x$ a prediction $\hat{f}(x)$ and also an uncertainty $\hat{\sigma}(x)$, thus enabling to use tools inspired by Bayesian optimization. The proposed extended aggregate models are incorporated into the search step of MADS where at each iteration a surrogate subproblem derived from Bayesian optimization is solved in order to come up with new candidate points. The proposed approach may be used in any direct-search method based on the search-poll paradigm, or in any approach akin to efficient global optimization if adapted. Any ensemble of models can be used along with any weight attribution technique provided that at least two models have a strictly positive weight at any moment.

The resulting algorithm has been tested on seven analytical problems, two multidisciplinary optimization problems and two simulation problems. The results show that the proposed extended aggregate models incorporated into MADS find better solutions than MADS without search step or with the help of quadratic models on three expensive problems out of four. And considering only the single best run on each problem, the new approach is better than the other MADS version on all real-world problems. It also finds better solutions than the stochastic models that it replaces while requiring less computational time. It should be noted that the models used to build the aggregate models must remain moderately expensive to compute, otherwise the method might lose its advantage in terms of computation time. The new approach does not show an advantage over quadratic models on analytical problems. The latter also yields better results on most runs of the solar 1 problem which most of the constraints are linear.

The comparison to other solvers than MADS shows that solvers using models (SHEBO and SACOBRA) tend to yield better results than the one not using models (DFN), although there are some cases where DFN outperforms SACOBRA. In any case, the new approach still manage to find better solutions within less computation time. Indeed, the solvers that use models also require much more computation time due to models construction, showing that one expensive model can be more time-consuming than a reasonable-to-compute ensemble of models.

An extended study of the various sub-problem formulations has not been conducted but based on the present results the formulations SP1, SP3 and SP8 can be recommended, and the smooth uncertainty can be considered as more reliable than the nonsmooth uncertainty.
Future work may explore the uncertainty for the constraint independently from that of the objective since in this work the two were coupled, e.g., smooth uncertainty for the objective together with nonsmooth uncertainty for the constraint. More specifically, each alternative for the constraints should be studied separately in order to determine their influence in the results and whether one is better than the other. In addition, other weight attribution techniques than that described in Sect. 3.2 may also be considered. The influence of the simplex used to build simplex gradients in (2) as well as the positive spanning set in (3) have not been studied in this work. Besides, the parameter $\lambda$ of the surrogate subproblems has been carefully selected for each formulation but remains constant over the optimization once determined. It might instead be dynamically updated depending on the result of the search or merely follow a predetermined trend like decreasing with the number of iterations. The parameter $\alpha$ in (7) is proportional to the global variance of the cache. It could be refined in order to represent local trends better, for instance by taking into account the values of the cache only in a restricted region around the evaluated point, or by removing outliers. Finally, the formulations were chosen on a purely empirical basis. Little effort has been made to finely understand the behaviour and the performance thereof. A thorough analysis of the benefits of each formulations in the context of extended aggregate models might be a judicious undertaking.

A Positive spanning set and simplex construction

The simplex and the positive spanning set described below are built in the scaled search space. Before constructing the models, the NOMAD software used in this work scales each input variable $x_i, i \in \{1, 2, \ldots, n\}$, using the mean and the variance of the points of the cache. This is done to give the same importance to all the variables regardless of their initial amplitude. Consequently, the simplex and the positive spanning set are isotropic in the scaled search space, but not in the actual search space.

- The simplex centred on $x \in \mathbb{R}^n$ used to build the simplex gradients $\nabla_S f(x)$ in Eq. (2) is

$$\{ x + 0.001 d_i : i \in \{1, 2, \ldots, n+1\} \}$$

where the directions $d_i, i \in \{1, 2, \ldots, n+1\}$, are constructed according to the following procedure:

$$d_i = \begin{cases} 
ei - \frac{1}{\sqrt{n+1}} \times [1, 1, \ldots, 1]^T, & \text{if } i \in \{1, 2, \ldots, n\} \\ \frac{1}{\sqrt{2(n+1)}} \times [1, 1, \ldots, 1]^T, & \text{if } i = n+1 \end{cases}$$

\[ Springer \]
where \( e_i \) is the vector \([0, \ldots, 1, \ldots, 0]^\top\) with value 1 at the \(i\)th position. This procedure forms a regular simplex centred on \( x \) with side length equal to \( \sqrt{2} \) in any dimension. The factor 0.001 was chosen empirically on preliminary tests.

- The positive spanning set centred on \( x \in \mathbb{R}^n \) used in Eq. (3) is

\[
\{ x \pm 0.005e_i : i \in \{1, 2, \ldots, n\} \}
\]

This positive spanning set contains \(2n\) elements. The factor 0.005 was chosen empirically on preliminary tests.

### B Surrogate subproblem formulations

**[6pt]** \( \min_{x \in \mathcal{X}} \hat{f}(x) - \lambda \hat{\sigma}_f(x) \)

s.t. \( \hat{c}_j(x) - \lambda \hat{\sigma}_j(x) \leq 0, \quad j = 1, 2, \ldots, m \) (SP1-F\( \sigma \))

**[6pt]** \( \min_{x \in \mathcal{X}} -EI(x) - \lambda \hat{\sigma}_f(x) \)

s.t. \( \hat{c}_j(x) - \lambda \hat{\sigma}_j(x) \leq 0, \quad j = 1, 2, \ldots, m \) (SP3-EI\( \sigma \))

**[6pt]** \( \min_{x \in \mathcal{X}} -EFI(x) \)

s.t. \( \hat{c}_j(x) - \lambda \hat{\sigma}_j(x) \leq 0, \quad j = 1, 2, \ldots, m \) (SP4-EFI)

**[6pt]** \( \min_{x \in \mathcal{X}} -EFI(x) - \lambda \hat{\sigma}_f(x) \)

s.t. \( \hat{c}_j(x) - \lambda \hat{\sigma}_j(x) \leq 0, \quad j = 1, 2, \ldots, m \) (SP5-EFI\( \sigma \))

**[6pt]** \( \min_{x \in \mathcal{X}} -EI(x) - \lambda (EI(x) \mu(x) + P(x) \hat{\sigma}_f(x)) \)

s.t. \( \hat{c}_j(x) - \lambda \hat{\sigma}_j(x) \leq 0, \quad j = 1, 2, \ldots, m \) (SP7-EFI\( \mu \))

**[6pt]** \( \min_{x \in \mathcal{X}} -PFI(x) \)

s.t. \( \hat{c}_j(x) - \lambda \hat{\sigma}_j(x) \leq 0, \quad j = 1, 2, \ldots, m \) (SP8-PFI)

**Data availability statement** The solar and styrene problems are available at github.com/bbopt/solar and github.com/bbopt/styrene. The NOMAD software package is available at github.com/bbopt/nomad.

### References

1. Abramson, M.A., Audet, C., Dennis, J.E., Jr., Le Digabel, S.: OrthoMADS: A deterministic MADS instance with orthogonal directions. SIAM J. Optim. 20(2), 948–966 (2009)
2. Acar, E., Rais-Rohani, M.: Ensemble of metamodels with optimized weight factors. Struct. Multidiscip. Optim. 37(3), 279–294 (2009)
3. Audet, C.: Convergence results for generalized pattern search algorithms are tight. Optim. Eng. 5(2), 101–122 (2004)
4. Audet, C., Béchard, V., Chaouki, J.: Spent potliner treatment process optimization using a MADS algorithm. Optim. Eng. 9(2), 143–160 (2008)
5. Audet, C., Béchard, V., Le Digabel, S.: Nonsmooth optimization through Mesh Adaptive Direct Search and Variable Neighborhood Search. J. Glob. Optim. 41(2), 299–318 (2008)
6. Audet, C., Côté-Massicotte, J.: Dynamic improvements of static surrogates in direct search optimization. Optim. Lett. 13(6), 1433–1447 (2019)
7. Audet, C., Dennis, J.E., Jr.: Mesh adaptive direct search algorithms for constrained optimization. SIAM J. Optim. 17(1), 188–217 (2006)
8. Audet, C., Dennis, J.E., Jr.: A progressive barrier for derivative-free nonlinear programming. SIAM J. Optim. 20(1), 445–472 (2009)
9. Audet, C., Dennis, J.E., Jr., Le Digabel, S.: Parallel space decomposition of the mesh adaptive direct search algorithm. SIAM J. Optim. 19(3), 1150–1170 (2008)
10. Audet, C., Hare, W.: Derivative-Free and Blackbox Optimization. Springer Series in Operations Research and Financial Engineering. Springer, Cham (2017)
11. Audet, C., Hare, W.: Model-Based Methods in Derivative-Free Nonsmooth Optimization, pp. 655–691. Springer (2020)
12. Audet, C., Kokkolaras, M., Le Digabel, S., Talgorn, B.: Order-based error for managing ensembles of surrogates in mesh adaptive direct search. J. Glob. Optim. 70(3), 645–675 (2018)
13. Audet, C., Le Digabel, S., Rochon Montplaisir, V., Tribes, C.: Algorithm XXXX: NOMAD version 4: Nonlinear optimization with the MADS algorithm. To appear in ACM Transactions on Mathematical Software (2022)
14. Audet, C., Le Digabel, S., Rochon Montplaisir, V., Tribes, C.: The NOMAD project. Software available at https://www.gerad.ca/nomad (2022)
15. Audet, C., Orban, D.: Finding optimal algorithmic parameters using derivative-free optimization. SIAM J. Optim. 17(3), 642–664 (2006)
16. Bagheri, S., Konen, W., Emmerich, M., Bäck, T.: Self-adjusting parameter control for surrogate-assisted constrained optimization under limited budgets. Appl. Soft Comput. 61, 377–393 (2017)
17. Björkman, M., Holmström, K.: Global optimization of costly nonconvex functions using radial basis functions. Optim. Eng. 1, 373–397 (2000)
18. Booker, A.J., Dennis, J.E., Jr., Frank, P.D., Serafini, D.B., Torczon, V., Trosset, M.W.: A rigorous framework for optimization of expensive functions by surrogates. Struct. Multidiscip. Optim. 17(1), 1–13 (1999)
19. Chen, L., Qiu, H., Jiang, C., Cai, X., Gao, L.: Ensemble of surrogates with hybrid method using global and local measures for engineering design. Struct. Multidiscip. Optim. 57(4), 1711–1729 (2018)
20. Choi, T.D., Eslinger, O.J., Kelley, C.T., David, J.W., Etheridge, M.: Optimization of automotive valve train components with implicit filtering. Optim. Eng. 1(1), 9–27 (2000)
21. Clarke, F.H.: Optimization and Nonsmooth Analysis. John Wiley and Sons, New York (1983). Reissued in 1990 by SIAM Publications, Philadelphia, as Vol. 5 in the series Classics in Applied Mathematics
22. Conn, A.R., Le Digabel, S.: Use of quadratic models with mesh-adaptive direct search for constrained black box optimization. Optim. Methods Softw. 28(1), 139–158 (2013)
23. Conn, A.R., Scheinberg, K., Vicente, L.N.: Introduction to Derivative-Free Optimization. MOS-SIAM Series on Optimization. SIAM, Philadelphia (2009)
24. Conn, A.R., Toint, Ph.L.: An algorithm using quadratic interpolation for unconstrained derivative free optimization. In: Di Pillo, G., Giannessi, F. (eds.) Nonlinear Optimization and Applications, pp. 27–47. Plenum Publishing, New York (1996)
25. Custódio, A.L., Rocha, H., Vicente, L.N.: Incorporating minimum Frobenius norm models in direct search. Comput. Optim. Appl. 46(2), 265–278 (2010)
26. Davis, C.: Theory of positive linear dependence. Am. J. Math. 76, 733–746 (1954)
27. Fasano, G., Liuzzi, G., Lucidi, S., Rinaldi, F.: A linesearch-based derivative-free approach for nonsmooth constrained optimization. SIAM J. Optim. 24(3), 959–992 (2014)
28. Lemyre Garneau, M.: Modelling of a solar thermal power plant for benchmarking blackbox optimization solvers. Master’s thesis, Polytechnique Montréal, 2015. https://publications.polymtl.ca/1996/
29. Goel, T., Hafika, R.T., Shyy, W., Queipo, N.V.: Ensemble of surrogates. Struct. Multidiscip. Optim. 33(3), 199–216 (2007)
30. Gramacy, R.B., Le Digabel, S.: The mesh adaptive direct search algorithm with treed Gaussian process surrogates. Pac. J. Optim. 11(3), 419–447 (2015)
31. Greenhill, S., Rana, S., Gupta, S., Vellanki, P., Venkatesh, S.: Bayesian optimization for adaptive experimental design: a review. IEEE Access 8, 13937–13948 (2020)
32. Hock, W., Schittkowski, K.: Test examples for nonlinear programming codes. Lecture notes in economics and mathematical systems, vol. 187. Springer, Berlin (1981)
33. Jones, D.R.: A taxonomy of global optimization methods based on response surfaces. J. Global Optim. 21, 345–383 (2001)
35. Jones, D.R., Schonlau, M., Welch, W.J.: Efficient global optimization of expensive black box functions. J. Global Optim. 13(4), 455–492 (1998)
36. Kelley, C.T.: Detection and remediation of stagnation in the Nelder–Mead algorithm using a sufficient decrease condition. SIAM J. Optim. 10, 43–55 (1999)
37. Kitayama, S., Arakawa, M., Yamazaki, K.: Sequential approximate optimization using radial basis function network for engineering optimization. Optim. Eng. 12(4), 535–557 (2011)
38. Kodiylalam, S.: Multidisciplinary aerospace systems optimization. Technical Report NASA/CR-2001-211053, Lockheed Martin Space Systems Company, Computational AeroSciences Project, Sunnyvale, CA (2001)
39. Larson, J., Menickelly, M., Wild, S.M.: Derivative-free optimization methods. Acta Numer. 28, 287–404 (2019)
40. Le Digabel, S., Wild, S.M.: A Taxonomy of Constraints in Simulation-Based Optimization. Technical Report G-2015-57, Les cahiers du GERAD (2015)
41. L. Lukšan and J. Vlček. Test problems for nonsmooth unconstrained and linearly constrained optimization. Technical Report V-798, ICS AS CR (2000)
42. Marsden, A.L., Wang, M., Dennis, J.E., Jr., Moin, P.: Trailing-edge noise reduction using derivative-free optimization and large-eddy simulation. J. Fluid Mech. 572, 13–36 (2007)
43. McKay, M.D., Beckman, R.J., Conover, W.J.: A 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)
44. Mockus, J., Tiesis, V., Zilinskas, A.: The application of Bayesian methods for seeking the extremum, pp. 117–129. North-Holand (1978)
45. Moré, J.J., Wild, S.M.: Benchmarking derivative-free optimization algorithms. SIAM J. Optim. 20(1), 172–191 (2009)
46. Müller, J.: An algorithmic framework for the optimization of computationally expensive bi-fidelity black-box problems. INFOP Inf. Syst. Oper. Res. 58(2), 264–289 (2020)
47. Müller, J., Day, M.: Surrogate Optimization of Computationally Expensive Black-Box Problems with Hidden Constraints. INFORMS J. Comput. 31(4), 689–702 (2019)
48. Müller, J., Park, J., Sahu, R., Varadharajan, C., Arora, B., Faybishenko, B., Agarwal, D.: Surrogate optimization of deep neural networks for groundwater predictions. J. Glob. Optim. (2020)
49. Müller, J., Piché, R.: Mixture surrogate models based on Dempster-Shafer theory for global optimization problems. J. Glob. Optim. 51(1), 79–104 (2011)
50. Nelder, J.A., Wedderburn, R.W.M.: Generalized linear models. J. R. Stat. Soc.: Ser. A (Gen.) 135, 370–384 (1972)
51. Peremeshzhey, N., Hines, E., Lapkin, A., Connaughton, C.: Combining gaussian processes, mutual information and a genetic algorithm for multi-target optimization of expensive-to-evaluate functions. Eng. Optim. 46(11), 1593–1607 (2014)
52. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. The MIT Press (2006)
53. Regis, R.G.: Stochastic radial basis function algorithms for large-scale optimization involving expensive black-box objective and constraint functions. Comput. Oper. Res. 38(5), 837–853 (2011)
54. Regis, R.G.: Constrained optimization by radial basis function interpolation for high-dimensional expensive black-box problems with infeasible initial points. Eng. Optim. 46(2), 218–243 (2014)
55. Regis, R.G., Shoemaker, C.A.: Constrained global optimization of expensive black box functions using radial basis functions. J. Global Optim. 31, 153–171 (2005)
56. Ruan, X., Jiang, P., Zhou, Q., Hu, J., Shu, L.: Variable-fidelity probability of improvement method for efficient global optimization of expensive black-box problems. Struct. Multidiscip. Optim. 62, 3021–3052 (2020)
57. Sacks, J., Welch, W.J., Mitchell, T.J., Wynn, H.P.: Design and analysis of computer experiments. Stat. Sci. 4(4), 409–435 (1989)
58. Shahriari, B., Swersky, K., Wang, Z., Adams, R.P., De Freitas, N.: Taking the human out of the loop: a review of Bayesian optimization. Proc. IEEE 104(1), 148–175 (2015)
59. Srinivas, N., Krause, A., Kakade, S.M., Seeger, M.W.: Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. Technical Report 0912.3995, arXiv (2010)
60. Taddy, M.A., Gramacy, R.B., Polson, N.G.: Dynamic trees for learning and design. J. Am. Stat. Assoc. 106(493), 109–123 (2011)
61. Talgorn, B., Audet, C., Kokkolaras, M., Le Digabel, S.: Locally weighted regression models for surrogate-assisted design optimization. Optim. Eng. 19(1), 213–238 (2018)
62. Talgorn, B., Le Digabel, S., Kokkolaras, M.: Statistical surrogate formulations for simulation-based design optimization. J. Mech. Des. **137**(2), 021405–1–021405–18 (2015)
63. Le Thi, H.A., Vaz, A.I.F., Vicente, L.N.: Optimizing radial basis functions by dc programming and its use in direct search for global derivative-free optimization. TOP **20**(1), 190–214 (2012)
64. Toal, D.J.J.: Some considerations regarding the use of multi-fidelity Kriging in the construction of surrogate models. Struct. Multidiscip. Optim. **51**(6), 1223–1245 (2015)
65. Torczon, V.: On the convergence of pattern search algorithms. SIAM J. Optim. **7**(1), 1–25 (1997)
66. Tournemenne, R., Petiot, J.-F., Talgorn, B., Kokkolaras, M., Gilbert, J.: Brass instruments design using physics-based sound simulation models and surrogate-assisted derivative-free optimization. J. Mech. Des. **139**(4), 041401–01–041401–9 (2017)
67. Tribes, C., Dubé, J.-F., Trépanier, J.-Y.: Decomposition of multidisciplinary optimization problems: formulations and application to a simplified wing design. Eng. Optim. **37**(8), 775–796 (2005)
68. Verdério, A., Karas, E.W.: On the construction of quadratic models for derivative-free trust-region algorithms. EURO J. Comput. Optim. **5**(4), 501–527 (2017)
69. Viana, F.A.C., Haftka, R.T., Valder, S., Jr., Butkewitsch, S., Leal, M.F.: Ensemble of surrogates: a framework based on minimization of the mean integrated square error. In 49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials, Schaumburg, IL (2008)
70. Viana, F.A.C., Haftka, R.T., Watson, L.T.: Efficient global optimization algorithm assisted by multiple surrogate techniques. J. Glob. Optim. **56**(2), 669–689 (2013)
71. Vu, K.K., D’Ambrosio, C., Hamadi, Y., Liberti, L.: Surrogate-based methods for black-box optimization. Int. Trans. Oper. Res. **24**(3), 393–424 (2017)
72. Wild, S.M., Regis, R.G., Shoemaker, C.A.: ORBIT: optimization by radial basis function interpolation in trust-regions. SIAM J. Sci. Comput. **30**(6), 3197–3219 (2008)
73. Wild, S.M., Shoemaker, C.A.: Global convergence of radial basis function trust region derivative-free algorithms. SIAM J. Optim. **21**(3), 761–781 (2011)
74. Ye, P., Pan, G., Dong, Z.: Ensemble of surrogate based global optimization methods using hierarchical design space reduction. Struct. Multidiscip. Optim. **58**, 537–554 (2018)

**Publisher’s Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.