Advanced Modeling and Simulation in Engineering Sciences

RESEARCH ARTICLE　　　　　　　　　　　　　　　　　　　　**Open Access**

A comparison of mixed-variables Bayesian optimization approaches

Jhouben Cuesta Ramirez[1,3]* , Rodolphe Le Riche[3], Olivier Roustant[2], Guillaume Perrin[4], Cédric Durantin[5] and Alain Glière[1]

*Correspondence:
jycuesta@utp.edu.co

[3]LIMOS (CNRS, Mines Saint-Etienne, UCA), Saint-Etienne, France
Full list of author information is available at the end of the article

Abstract

Most real optimization problems are defined over a mixed search space where the variables are both discrete and continuous. In engineering applications, the objective function is typically calculated with a numerically costly black-box simulation. General mixed and costly optimization problems are therefore of a great practical interest, yet their resolution remains in a large part an open scientific question. In this article, costly mixed problems are approached through Gaussian processes where the discrete variables are relaxed into continuous latent variables. The continuous space is more easily harvested by classical Bayesian optimization techniques than a mixed space would. Discrete variables are recovered either subsequently to the continuous optimization, or simultaneously with an additional continuous-discrete compatibility constraint that is handled with augmented Lagrangians. Several possible implementations of such Bayesian mixed optimizers are compared. In particular, the reformulation of the problem with continuous latent variables is put in competition with searches working directly in the mixed space. Among the algorithms involving latent variables and an augmented Lagrangian, a particular attention is devoted to the Lagrange multipliers for which a local and a global estimation techniques are studied. The comparisons are based on the repeated optimization of three analytical functions and a beam design problem.

Introduction

A key task in engineering design is to find an optimal configuration from a very large set of alternatives. When the performance of the candidate solutions is measured through a realistic simulation, the numerical cost of the procedure becomes a bottleneck. The optimization of computationally expensive simulators is a topic widely studied in the literature Thi et al. [26].

In this work, we focus on Bayesian optimization (BO), which is particularly suitable for solving such problems Frazier [9]. Bayesian optimization is a sequential design strategy that requires a data-driven mathematical model or metamodel that provides predictions along with their uncertainty Bartz-Beielstein et al. [2]. The metamodel replaces some of the calls to the expensive simulation and is a key ingredient to the optimization of costly functions. An acquisition criterion Wilson et al. [29] aggregates the spatial predictions and

Cuesta-Ramirez *et al. Adv. Model. and Simul. in Eng. Sci.*(2022)9:6

Page 2 of 29

uncertainties. The metamodel is trained from a reduced set of simulation data and the acquisition criterion is maximized to propose new configurations to be simulated at the next iteration. When the acquisition criterion is the expected improvement (EI), as first introduced in Mockus et al. [18], the BO algorithm is often called EGO (Efficient Global Optimization, Jones et al. [12]). EGO is currently a state-of-the-art approach to medium size, continuous and costly optimization problems, both from an empirical Le Riche and Picheny [14] and a theoretical point of view Vazquez and Bect [27].

However, in realistic settings, some of decision variables are categorical. In structural design for example, the type of material, the number of components, the choice between alternative technologies lead to discrete variables with no obvious distance between them. The combination of continuous and categorical variables is called a mixed optimization problem.

In non-costly cases, mixed optimization problems can be approached by Mixed-Integer NonLinear Programming Belotti et al. [4] (when the discrete variables are integers), by sampling based techniques such as evolutionary optimization Cao et al. [6], Emmerich et al. [8], Ocenasek and Schwarz [20] or by alternating mixed programming Audet and Dennis Jr [1].

When the objective function is costly, mixed optimization problems remain challenging and a topic for research. It is customary to replace some of the calls to the original objective function by calls to a (meta)model of it. Bartz-Beielstein and Zaefferer [3] provide an overview of metamodels that have or can be used in optimization when the variables are continuous or discrete. Bayesian optimization methods, which rely on metamodels to save computations, have already been extended to mixed problems. It was made possible by the realization that GP kernels (covariance functions) in mixed variables can be created by composing continuous and discrete kernels. The acquisition function is defined over the same space as the objective function. Therefore maximizing the acquisition function is also a mixed variables problem.

To the best of our knowledge, the first EGO-like algorithm for mixed variables has been proposed in Hutter et al. [11]. In this article, the mixed kernel is a product of continuous and discrete Gaussian kernels, and random forests constitute an alternative choice of mixed metamodel. More precisely, the discrete kernel is a Gaussian of integer or hamming (also known as Gower) distance for ordinal or nominal variables, respectively. In Hutter et al. [11], the expected improvement is first optimized with a multi-start local search for both the continuous and discrete variables (thus a neighborhood for the discrete variables is defined) which is then complemented by a random search. This work was continued with the REMBO method in Wang et al. [28], where a random linear embedding is introduced to tackle high-dimensional problems. Discrete variables were relaxed into continuous variables thanks to a mapping function. The optimization of the acquisition function was made with a combination of the DIRECT and CMA-ES continuous global optimizers. Both Hutter et al. [11] and Wang et al. [28] have been motivated by applications to the automatic configuration of algorithms. The goal of reaching very high dimensions (millions) probably forced the authors to use isotropic kernels as a way to keep the number of hyper-parameters low (only one length-scale for all dimensions).

A Bayesian mixed optimizer is presented in Pelamatti et al. [21]. The GP kernels are products of continuous and discrete kernels. Different discrete kernels are compared, namely the homo- and hetero-scedastic hypersphere decomposition and the compound

Cuesta-Ramirez *et al. Adv. Model. and Simul. in Eng. Sci.*(2022)9:6

Page 3 of 29

symmetric kernels. The optimization of the acquisition function is performed with a genetic algorithm in mixed variables. A similar BO with mixed kernel is described in Zuniga and Sinoquet [33], but the expected improvement is optimized with the mixed version of the MADS algorithm Audet and Dennis Jr [1] and the neighborhood of the categorical variables is defined through a probabilistic model.

Random forests can replace the kriging model in BO with mixed inputs as they natively have a measure of prediction uncertainty. Such an implementation, first done in Hutter et al. [11], is part of the `mlrMBO` R package Bischl et al. [5], in conjunction with several acquisition criteria that can be optimized with a "focus-search" algorithm. The focus-search algorithm hierarchically samples the search space of the chosen acquisition criterion.

Recent developments in metamodels involving mixed variables show that it is possible to map the categorical variables into quantitative non-observed *latent variables* that are then considered as continuous Zhang et al. [31]. Whenever it is possible to write a model of the studied system, quantitative latent variables exist that describe the effects of the categorical variables. Typically, there are more latent variables than categorical variables. The existence of continuous latent variables can sometimes be established from the physics of the considered phenomena, e.g. in material science Zhang et al. [32]. In structural mechanics for example, if the categorical variable describes the shape and the material of an element load in flexion, its bending moment of inertia is a candidate latent variable. Latent variables can emulate the properties of the original categorical variables, in particular within the metamodel, and open the way to reasonings with continous quantities: the kernels of the Gaussian processes can be taken as continuous, gradients and neighborhoods are naturally defined during the optimization. On the contrary, categorical variables and their inherent lack of distance definition is the cause of complications in the kernel definition and in the optimization.

This article presents a new Bayesian optimization algorithm for mixed variables called LV-EGO (for Latent Variable EGO). Our contribution with respect to Zhang et al. [32] is that the continuity of the latent variables is also taken advantage of during the optimization of the acquisition criterion. This implies that categorical variables must be recovered from the continuous latent variables proposed by the optimizer, which creates a new "pre-image" problem.

"Problem statement and background" introduces the problem and the principles of Bayesian optimization. In "EGO with latent variables", several variants of LV-EGO are described. They differ in the handling of the relationship between the categorical and the latent variables: the "vanilla" LV-EGO just recovers categorical variables after the optimization while augmented Lagrangian versions account for the link during the optimization through constraints. "Description of the numerical experiments" presents a set of benchmarks comparing our method to other state-of-the-art techniques. One of the benchmarks is a beam design problem and gives the opportunity to discuss the interpretation of the latent variables. Finally, "Conclusions and perspectives" offers conclusions and perspectives to this work.

Problem statement and background

We consider the problem of minimizing a function $y(x, u)$ depending on a vector of continuous variables $x = (x_1, \ldots, x_{n_c})$ and a vector of discrete variables $u = (u_1, \ldots, u_{n_d})$,

Cuesta-Ramirez *et al. Adv. Model. and Simul. in Eng. Sci.*(2022)9:6

Page 4 of 29

where each u_i has m_i levels encoded $1, \ldots, m_i$. We denote $\mathcal{X}$ the domain of definition for the continuous inputs, typically, after rescaling, the hypercubic domain $[0, 1]_c^n$. Similarly, we denote $\mathcal{U} = \prod_{j=1}^{n_d} \{1, \ldots, m_j\}$ the domain of definition for the discrete inputs. $\mathcal{X} \times \mathcal{U}$ is the set of the mixed optimization variables.

We focus on costly functions, meaning that each evaluation of y is time-consuming, and we aim at minimizing y with a tiny budget of evaluations. In this context, minimizing directly y is hardly possible. An alternative is to use Bayesian optimization (BO). In BO approaches, there are two main ingredients: a Gaussian process (GP) serving as a fast proxy, often called metamodel, built from the current learning set, and a sampling criterion, often called acquisition criterion, used to update the learning set with a new data point computed with y. A famous acquisition criterion is the expected improvement (EI). In that case, the BO approach is often called Efficient Global Optimization (EGO) algorithm.

To be more precise, let $(X, U) = \{(x, u)^{(1)}, \ldots, (x, u)^{(t)}\} \in (\mathcal{X} \times \mathcal{U})^t$ be a design of experiments (DoE), and $y_i = y(x^{(i)}, u^{(i)})$ be the corresponding function evaluations ($i = 1, \ldots, t$). Let $y_{\min} = \min(y_1, \ldots, y_t)$ be the current minimum. Let us now assume that y is a particular realization of the GP Y defined on $\mathcal{X} \times \mathcal{U}$. In that case, the EI criterion is defined by

$$\text{EI}(x, u) = \text{E}\left[\max(y_{\min} - Y^t(w), 0)\right], \ (x, u) \in (\mathcal{X} \times \mathcal{U}),$$

where Y^t is the conditional GP knowing the observations:

$$Y^t := Y \mid \{Y((x, u)^{(1)}) = y_1, \ldots, Y((x, u)^{(t)}) = y_t\}.$$

Notice that $\text{EI}(x, u)$ is large when exploiting interesting area, that is to say when there is a good chance that $Y^t(x, u)$ is smaller than $y_{\min}$. This may occur when $\text{E}[Y^t(x, u)]$ is close to $y_{\min}$, or when exploring unvisited areas, i.e. when the variance of $Y^t(x, u)$ is large compared to $(\text{E}[Y^t(x, u)] - y_{\min})^2$. The idea of EGO is to evaluate y at a new point maximizing the EI criterion until a stopping criterion is reached. See Algorithm 1 for a synthetic description of the EGO algorithm when the stopping criterion is a maximum number of evaluations of y, noted *budget*. A maximum budget is the logical stopping criterion in our context of costly optimization. Other stopping conditions are possible in the form of lower bounds on the acquisition criteria (expected improvement, knowledge gradient Frazier [9],...) i.e., minimal measures of progress below which the search should stop. In line 9, the solution returned by the algorithm is the best point of the last DoE, (X, U).

Algorithm 1 EGO algorithm on a mixed space

1: Generate the initial DoE of size N_{DoE}, (X, U), and calculate $Y = (y_1, \ldots, y_{N_{\text{DoE}}})$, $t \leftarrow N_{\text{DoE}}$.
2: **while** $t \leq$ budget **do**
3:     Estimate the GP Y^t from the learning set formed by (X, U) and Y.
4:     Look for the current minimum $y_{\min}$ and maximize $(x, u) \mapsto \text{EI}(x, u)$ on $\mathcal{X} \times \mathcal{U}$: $(x^{t+1}, u^{t+1}) \in \text{argmax}_{(x,u) \in \mathcal{X} \times \mathcal{U}} \text{EI}(x, u)$.
5:     Evaluate y at (x^{t+1}, u^{t+1}), $y^{t+1} = y(x^{t+1}, u^{t+1})$.
6:     Update the learning set: $(X, U) \leftarrow (X, U) \cup (x^{t+1}, u^{t+1})$, $Y \leftarrow Y \cup \{y^{t+1}\}$.
7:     $t \leftarrow t + 1$
8: **end while**
9: $(x^\star, u^\star) = \arg\min_{(x,u) \in (X,U)} y(x, u)$, $y^\star = y(x^\star, u^\star)$
10: **return** $(x^\star, u^\star, y^\star)$

Cuesta-Ramirez *et al. Adv. Model. and Simul. in Eng. Sci.*(2022)9:6

Page 5 of 29

This EGO algorithm has been intensively studied to minimize nonlinear functions that are expensive to be evaluated in the case $\mathcal{U} = \emptyset$, i.e. when all input variables are continuous (see Le Riche and Picheny [14] for numerical illustrations of its efficiency). The application of this algorithm in the presence of categorical variables is much less documented (see e.g. Pelamatti et al. [21], Zuniga and Sinoquet [33]), which can be explained by two main difficulties. The first one is related to the difficult estimation of covariance kernels on mixed spaces. Indeed, multi-dimensional covariance functions are often built by combination of one-dimensional ones. Therefore, covariance functions on mixed spaces can be obtained by combining covariance functions on $\mathcal{X}$ and $\mathcal{U}$:

$$\text{Cov}(Y(x, u), Y(x', u')) = k_1^x(x_1, x_1') * \cdots * k_{n_c}^x(x_{n_c}, x_{n_c}') * k_1^u(u_1, u_1') * \cdots * k_{n_d}^u(u_{n_d}, u_{n_d}'),$$
(1.1)

where $k_1^x, \ldots, k_{n_c}^x, k_1^u, \ldots, k_{n_d}^u$ are covariance functions and $*$ is an operation that preserves positive definiteness, such as sum or product. If we focus on the single categorical variable u_j with levels $1, \ldots, m_j$, we can identify the covariance function k_j^u to a $(m_j \times m_j)$-dimensional positive semidefinite matrix $\mathbf{T}$, such that for all $1 \leq k, \ell \leq m_j$,

$$(\mathbf{T})_{k\ell} = k_j^u(k, \ell).$$
(1.2)

This means that $\sum_{j=1}^{n_d} m_j(m_j + 1)/2$ coefficients need to be estimated to determine a covariance on $\mathcal{U}$ in the general case. That number can be large when m is large, which very often makes this estimation very difficult in practice. Furthermore, the optimization problem is often harder than the box-constrained one met with continuous variables. Indeed it is either constrained by the positive definiteness of $\mathbf{T}$, which is non-linear, or defined on a manifold if $\mathbf{T}$ is parameterized in spherical coordinates. We refer to Roustant et al. [25] for more details and other parsimonious representations of k_j^u, which can reduce but not totally fix these issues. The second reason that can explain the few number of direct applications of EGO algorithm on mixed space is related to the difficult maximization of the expected improvement, i.e. the search of the new input points where to call the function y, which are solutions of:

$$\max_{x, u \in \mathcal{X} \times \mathcal{U}} \text{EI}(x, u).$$
(1.3)

Indeed, classical optimization algorithms on continuous spaces usually try to exploit information related to the gradient of the function to be maximized, as well as notions of proximity in the space of the inputs. However, these two notions are difficult to exploit when dealing with categorical inputs, i.e. without any a priori ordering between the input instances. To circumvent this difficulty, a naive approach of resolution would consist in no longer considering a single maximization problem on $\mathcal{X} \times \mathcal{U}$, but the resolution in parallel of $\prod_{j=1}^{n_d} m_j$ maximization problems on $\mathcal{X}$, i.e. one problem per combination of instances of the categorical inputs u. Such an approach is not tractable when the number of optimization problems to be solved becomes large, which has motivated the definition of heuristics, such as evolutionary algorithms Li et al. [15], Cao et al. [6], Lin et al. [16], which seek to concentrate the searches only on the interesting instances of u. These approaches still rely on a large number of calls to the function to be optimized, and their convergence is not always easy to quantify.

Because mixed optimization problems are difficult, an alternative approach is proposed in the rest of this paper. It is based on the possibility to relax the discrete variables into con-

tinuous latent variables, therefore benefiting from the more efficient search mechanisms that exist in continuous spaces (e.g. gradients).

EGO with latent variables

Latent variables

For an easier handling of categorical inputs, it was proposed in Zhang et al. [31] to replace each categorical input u_j by a vector of $q_j \geq 1$ continuous inputs with values in $\mathbb{R}^{q_j}$, noted ℓ_j. To give an intuition of the underlying idea in the automotive domain, a category of lubricant may be determined by physical continuous features such as boiling temperature, viscosity, etc that act as latent variables. In structural mechanics, the shape of a load carrying structure, which is categorical, has underlying continuous flexural and membrane moments that drive its behavior. This amounts to associating to the Gaussian process (GP) Y a new GP $\widetilde{Y}$, such that for each instance u of the categorical inputs there exists a particular value of $\ell := (\ell_1, \ldots, \ell_{n_d}) \in \mathcal{L} \subset \mathbb{R}^{q_1} \times \cdots \times \mathbb{R}^{q_{n_d}}$, which is called *latent variable*, allowing us to write:

$$Y(x, u) \stackrel{\text{in law}}{=} \widetilde{Y}(x, \ell), \ x \in \mathcal{X}. \tag{2.1}$$

An important point is that the values of ℓ are unobserved and therefore $\widetilde{Y}$ is unknown. Nevertheless, in order to replace the EI maximization problem on $\mathcal{X} \times \mathcal{U}$ by a new optimization problem on $\mathcal{X} \times \mathcal{L}$, a precise knowledge of $\widetilde{Y}$ is not necessary. Indeed, assuming that kernels for mixed inputs are built by combining 1-dimensional ones as in (1.1), it is sufficient to identify the mappings ϕ_j from $\{1, \ldots, m_j\}$ to $\mathbb{R}^{q_j}$ to each variable u_j such that

$$k_j^u(u_j, u_j') \approx k_j(\phi_j(u_j), \phi_j(u_j')), \tag{2.2}$$

where k_j is a continuous kernel on $\mathbb{R}^{q_j} \times \mathbb{R}^{q_j}$. Thus, it is not so much the values of $\phi_j(u_j)$ that are important, but their relative positions in $\mathbb{R}^{q_j}$ in order to allow a reasonable reconstruction of the dependency structure between $Y(x, u)$ and $Y(x', u')$.

According to the works achieved in Zhang et al. [31], it appears that interesting mappings can be obtained by likelihood maximization and that relatively small values of q_j can give a satisfying reconstruction. Following their recommendations, q_j can be chosen equal to 1 if $m_j \leq 3$ and to 2 otherwise, which will be the values chosen in the rest of this paper. We denote by $n_\ell = \sum_{j=1}^{n_d} q_j$ the total number of latent variables. Following Roustant et al. [25], the continuous kernel k_j associated to the latent variables was chosen as the dot product kernel $k_j(t, t') = \langle t, t' \rangle$. The corresponding covariance matrix is then low-rank, and provided better performances than the Gaussian kernel in the examples considered in the latter reference.

This new parametrization leads us to the following adaptation of the EI maximization problem defined by Eq. (1.3), which we name *acquisition problem* as it allows to acquire a new point to evaluate:

$$\max_{x, \ell \in \mathcal{X} \times \mathcal{L} \subset \mathbb{R}^{n_c + n_\ell}} \text{EI}^{(t)}(x, \ell)$$
$$\text{such that } \exists u \in \mathcal{U} \text{ with } \ell = \phi^{(t)}(u). \tag{2.3}$$

Here, $\text{EI}^{(t)}(x, \ell)$ is the expected improvement associated with GP $\widetilde{Y}$ at iteration t, $\phi^{(t)} = (\phi_1^{(t)}, \ldots, \phi_{n_d}^{(t)})$ is the vector-valued mapping from $\prod_{j=1}^{n_d}\{1, \ldots, m_j\}$ to $\mathbb{R}^{q_1} \times \cdots \times \mathbb{R}^{q_{n_d}}$ at iteration t, and the constraint on the values of ℓ is driven by the fact that the values of the latent variables at the new point have to remain compatible with the current mapping functions.

We follow two paths to solve this acquisition problem. In the vanilla LV-EGO approach, which will be described soon, the EI maximization and the latent-discrete compatibility constraint are addressed one after each other. Alternatively, with the augmented Lagrangian approaches, which will be described in "LV-EGO algorithms with Augmented Lagrangian", the full constrained optimization problem is treated.

The vanilla LV-EGO algorithm

At each iteration, the vanilla LV-EGO algorithm first maximizes EI in a relaxed, fully continuous, formulation where the discrete variables are replaced by relaxed continuous latent variables. Then, a pre-image problem is solved where EI is maximized over the discrete variables only, the continuous variables being fixed at their value of the relaxed problem. The LV-EGO methodology is summarized in Algorithm 2.

Algorithm 2 Vanilla LV-EGO with mixed inputs

1: **Generate** the initial DoE of size N_{DoE}: X, U
2: **Costly function evaluations** $y(x^i, u^i)$, $i = 1, \ldots, N_{\text{DoE}}$, $t \leftarrow N_{\text{DoE}}$
3: **while** $t \leq$ budget **do**
4:    **Estimate** the latent variable mappings $\phi^{(t)}$ and the parameters of the continuous GP $\widetilde{Y}$.
5:    **Perform** one EGO iteration in the *relaxed continuous* space :
   $(x^{t+1}, \ell^{t+1}) = \arg \max_{x, \ell \in \mathcal{X} \times \mathcal{L} \subset \mathbb{R}^{n_c + n_\ell}} \text{EI}^{(t)}(x, \ell)$.
6:    **Recover** the *discrete pre-image* component u^{t+1} as: $u^{t+1} = \arg \max_{u \in \mathcal{U}} \text{EI}^{(t)}(x^{t+1}, \phi^{(t)}(u))$.
7:    **Update** the DoE with (x^{t+1}, u^{t+1}) with output value $y(x^{t+1}, u^{t+1})$.
8:    $t \leftarrow t + 1$
9: **end while**
10: **Return** $(x^\star, u^\star) = \arg \min_{x^t, u^t \in (X, U)} y(x^t, u^t)$

The main difference with the generic Bayesian algorithm 1 is the new discrete pre-image problem in line 6. Notice that the pre-image is formulated in terms of the EI objective, as opposed to a more arbitrary distance like $\|\ell^{t+1} - \phi^{(t)}(u)\|$. Solving the pre-image in terms of the iterative figure of merit, the expected improvement, is meant to provide a gain in efficiency with respect to a pre-image minimizing an Euclidean distance between the map of a discrete level and the latent variables. In the particular situation where the latent variable coincides with the image of a discrete level, $\ell^{t+1} = \phi^{(t)}(u^{t+1})$, both approaches yield the same result since ℓ^{t+1} is a maximizer of EI (see line 5 of Algorithm 2). In terms of implementation, the EI maximization (line 5) is done with the COBYLA algorithm, a gradient free non-linear optimization technique Powell [23]. Since COBYLA is a local optimizer and the EI is a multimodal function, the maximization is repeated (10 times, which is more than the maximum dimension of the test cases studied in this article and more than the default—3—of the `kergp` package) from randomly chosen initial points and the best result is kept. An exhaustive search is carried out for the EI maximization of the pre-image problem (line 6).

A comparison of the numerical complexities of the vanilla LV-EGO (Algorithm 2) and the generic EGO (Algorithm 1) shows that the cost of the latent variables is limited. Let us consider that the discrete space can be searched essentially by enumeration in $\mathcal{O}(\text{card}\,\mathcal{U}) = \mathcal{O}(\prod_{i=1}^{n_d} m_i)$ operations (where m_i is the number of levels per discrete variable) while a

Cuesta-Ramirez *et al. Adv. Model. and Simul. in Eng. Sci.(2022)9:6*

Page 8 of 29

continuous space can be searched more efficiently in linear time. At each iteration, the Bayesian algorithms of this paper have three steps: first a GP is learned, then an acquisition criterion (EI for now and an augmented Lagrangian later) is maximized and finally a pre-image problem is solved. In the vanilla LV-EGO algorithm, these steps take place at lines 4, 5 and 6 of Algorithm 2, respectively. Table 1 summarizes the number of operations per step. The number of operations for learning the GPs is proportional to the cube of the number of points evaluated (t) because of the inversions of the covariance matrices, times the number of (continuous) parameters of the GP for the likelihood maximization.

The two other steps, the acquisition and the pre-image, imply predictions by the GP in t^2 operations times a number of operations that depends on the specific algorithm. Comparing in Table 1 the column of the generic EGO with that of the vanilla LV-EGO, and assuming that for all i $m_i = m$ to keep the discussion simple, it can be seen that the latent variables induce a slight extra cost to be learnt. When $q = 2$, which is our default here, this extra cost is $n_d \times m_i \times t^3$ operations. Setting $q = 1$ would not add any cost to the learning. An advantage, which comes from the sequential resolution of the mixed problem, occurs in the maximization of the acquisition criterion when $n_c + q \times n_d \times m < m^{n_d} \times n_c$, at the cost of an additional pre-image problem to solve. Thus, LV-EGO will be faster than a mixed EGO once the latent variables are estimated if $m^{n_d} + n_c + q \times m \times n_d < m^{n_d} \times n_c$, which happens frequently (take for example $n_c = 4, n_d = 2, m = 10, q = 2$).

LV-EGO algorithms with Augmented Lagrangian

A possible pitfall of the vanilla LV-EGO detailed in Algorithm is that the link between the discrete variables u and their relaxed continuous counterparts ℓ is lost when maximizing $EI^{(t)}(x, \ell)$ in line 5. Recovering it during the discrete pre-image problem where x is fixed to a value optimal in the relaxed formulation but possibly non-optimal with respect to the mixed problem (1.3) may yield a sub-optimal solution. For this reason, we now propose LV-EGO algorithms that account for the discreteness constraint during the optimization using augmented Lagrangians.

In that prospect, notice that problem (2.3) can be approximated as an optimization problem with an inequality constraint:

$$
\min_{x, \ell \in \mathcal{X} \times \mathcal{L} \subset \mathbb{R}^{n_c + n_\ell}} f^{(t)}(x, \ell) := -\log(1 + EI^{(t)}(x, \ell))
$$

$$
\text{such that } g^{(t)}(\ell) := \min_{u \in \mathcal{U}} \| \ell - \phi^{(t)}(u) \| - \epsilon \leq 0
\tag{2.4}
$$

where ϵ is a small positive relaxation constant and $\|\cdot\|$ the Euclidean norm. In this reformulation, called *relaxed acquisition problem*, notice the log scaling of the EI which does not change the solution but improves the conditioning of the problem. Two values of ϵ will be discussed in the sequel, $\epsilon = 0$ in which case the constraint becomes an equality constraint, $\min_{u \in \mathcal{U}} \| \ell - \phi^{(t)}(u) \| = 0$, and $\epsilon > 0$ but small which corresponds to a relaxation of the equality. In the sequel, ϵ is normalized with respect to the size of the vector of latent variables and set to $\epsilon = 0.01$.

The constrained optimization problem (2.4) is solved through an augmented Lagrangian approach Minoux, [17], Nocedal and Wright [19]. The augmented Lagrangian is that of Rockafellar [24] which, specified for Problem (2.4), is,

$$
L_A^{(t)}(x, \ell; \lambda, \rho) = \begin{cases} f^{(t)}(x, \ell) - \frac{\lambda^2}{2\rho}, & \text{if } g^{(t)}(\ell) \leq \frac{-\lambda}{\rho}, \\ f^{(t)}(x, \ell) + \lambda g^{(t)}(\ell) + \frac{\rho}{2} g^{(t)}(\ell)^2, & \text{otherwise.} \end{cases}
\tag{2.5}
$$

Table 1 Numerical complexities of the algorithms compared at each iteration (for a given t)

	Fixed space search M	Vanilla LV-EGO V	LV-EGO-gA	LV-EGO-IA
	Alg. (1)	Alg. (2)	Alg. (3+4)	Alg. (3+5)
GP learning	$(n_c + \sum_{i=1}^{n_d} m_i) \times t^3$	$(n_c + q \times \sum_{i=1}^{n_d} m_i) \times t^3$	$(n_c + q \times \sum_{i=1}^{n_d} m_i) \times t^3$	$(n_c + q \times \sum_{i=1}^{n_d} m_i) \times t^3$
Max acquisition	$(\prod_{i=1}^{n_d} m_i) \times n_c \times t^2$	$(n_c + q \times \sum_{i=1}^{n_d} m_i) \times t^2$	$(N'_{DoE} + n_c + q \times \sum_{i=1}^{n_d} m_i) \times t^2$	$(n_c + q \times \sum_{i=1}^{n_d} m_i) \times t^2$
Pre-image	0	$(\prod_{i=1}^{n_d} m_i) \times t^2$	$(\prod_{i=1}^{n_d} m_i) \times t^2$	$(\prod_{i=1}^{n_d} m_i) \times t^2$

Cuesta-Ramirez *et al. Adv. Model. and Simul. in Eng. Sci.*(2022)9:6

Page 10 of 29

When $\epsilon = 0$, the constraint $g^{(t)}(\ell) \leq 0$ becomes an equality constraint, $g^{(t)}(\ell) = 0$. In this case, the augmented Lagrangian connected to that of Rockaffelar is that of Hestenes [10] and takes the form

$$L_A^{(t)}(x, \ell; \lambda, \rho) = f^{(t)}(x, \ell) + \lambda g^{(t)}(\ell) + \frac{\rho}{2} g^{(t)}(\ell)^2 \tag{2.6}$$

Complementary explanations about the augmented Lagrangians are given in Appendix .

Augmented Lagrangians require to specify the values of the Lagrange multiplier, λ, and of the penalty parameter, ρ. The general principle to fix them is to calculate the generalized Lagrange multiplier with a dual formulation Minoux [17]: the dual function $D^{(t)}$ is maximized with respect to the multiplier λ while the penalty parameter ρ should take the smallest value that allows one to find feasible solutions,

$$\rho_t = \arg \min_{\rho \geq 0} \rho \ \text{ such that } \ g(\ell^t) \leq 0$$

$$\text{where } \lambda_t = \arg \max_{\lambda \geq 0} D^{(t)}(\lambda, \rho),$$

$$D^{(t)}(\lambda, \rho) = \min_{x, \ell \in \mathcal{X} \times \mathcal{L} \subset \mathbb{R}^{n_c + n_\ell}} L_A^{(t)}(x, \ell; \lambda, \rho), \tag{2.7}$$

$$\text{and } (x^t, \ell^t) \in \arg \min_{x, \ell \in \mathcal{X} \times \mathcal{L} \subset \mathbb{R}^{n_c + n_\ell}} L_A^{(t)}(x, \ell; \lambda, \rho).$$

There are two logics to solve Problem (2.7), both of which have been investigated in this study. Following an idea presented in Le Riche and Guyon [13] for classical Lagrangians, we first propose to approximate the dual function $D()$ as the lower front of the augmented Lagrangians of a finite set of calculated points. The approximated dual is

$$\widehat{D}(\lambda, \rho) = \min_{(x, \ell) \in (X', L')} L_A^{(t)}(x, \ell; \lambda, \rho) \tag{2.8}$$

where (X', L') is a DoE that should not be mistaken for (X, U), the DoE of the original expensive problem. $(\lambda_t, \rho_t, x^t, \ell^t)$ comes from solving Problem (2.7) with minimizations over the finite set (X', L') instead of the initial $\mathcal{X} \times \mathcal{L}$. The functions in Problem (2.4) are not costly, (X', L') can be quite large. This approach is called *global dual* as a global approximation to the dual function is built and maximized. It applies to very general functions, e.g., non differentiable functions. Another advantage of this approach is to allow large changes in the dual space. Figure 10 provides an illustration of the approximated dual function and the effect of ρ on the dual problem. The sketch is done for an inequality constraint, yet it also stands with marginal changes for an equality (cf. Appendix and the caption to the Figure). Under the non-restrictive hypothesis that there is a ρ beyond which the solution to the primal problem (2.4) maximizes the dual function, maximizing the dual function preserves the global aspect of the search. However, the accuracy of the obtained (λ_t, ρ_t)'s will depend on the DoE. Because there is only one constraint in the current problem and evaluating it does not require calling the costly function, the maximization on λ and ρ is done by enumeration on a 100×20 grid and (X', L') is a 100 LHS sample.

The other path to updating the multiplier is to progressively change them based on the minimizers of the augmented Lagrangian at the current step. This updating can be seen as a step in the dual space which makes it general, although it is usually proved by analogy with the Karush Kuhn and Tucker optimality conditions Nocedal and Wright [19] which add unnecessary conditions (like differentiability), cf. Appendix . Let (x^t, ℓ^t) be a solution to

$$\min_{x, \ell \in \mathcal{X} \times \mathcal{L} \subset \mathbb{R}^{n_c + n_\ell}} L_A^{(t)}(x, \ell; \lambda_t, \rho_t) \tag{2.9}$$

The update formula reads

$$\lambda_{t+1} = \lambda_t + \rho_t \left(g^{(t)}(\ell^t) + \max(0, \frac{-\lambda_t}{\rho_t} - g^{(t)}(\ell^t)) \right) \qquad (2.10)$$

As in Picheny et al. [22], the penalty parameter ρ is simply increased if the constraint is not satisfied,

$$\rho_{t+1} = \begin{cases} \rho_t & \text{if} \quad g^{(t)}(\ell^t) \leq 0 \\ 2\rho_t & \text{otherwise} \end{cases} \qquad (2.11)$$

The update scheme based on Eqs. (2.10) and (2.11) is called *local dual* as a local step in the dual (λ, ρ) space is taken.

Algorithm 3 Augmented Lagrangian Latent Variables EGO with global or local dual scheme (ALV-EGO-g or ALV-EGO-l)

 1: **generate** the initial DoE of size N_{DoE} for (X, U)
 2: **costly function evaluations** $y(x^i, u^i)$, $i = 1, \ldots, N_{\text{DoE}}$, $t \leftarrow N_{\text{DoE}}$
 3: **initialize** budget, ϵ
 4: **while** $t \leq$ budget **do**
 5:     **estimate** the latent variables $\phi^{(t)}$ and the GP parameters from current DoE.
 6:     {approximately solve the relaxed acquisition problem (2.4) with $f^{(t)}(\cdot) = -\log(1+\text{EI}^{(t)}(\cdot))$}
        $(x^{t+1}, \ell^{t+1}) = \arg\min_{x, \ell} f^{(t)}(x, \ell)$ s.t. $g^{(t)}(\ell^{t+1}) = \min_{u \in \mathcal{U}} \|\ell - \phi^{(t)}(u)\| - \epsilon \leq 0$,
        ALV-EGO-g variant: with the global dual scheme, cf. Algorithm 4
        ALV-EGO-l variant: with the local dual scheme, cf. Algorithm 5
 7:     **recover** the *discrete pre-image* component u^{t+1} as: $u^{t+1} = \arg\max_{u \in \mathcal{U}} \text{EI}^{(t)}(x^{t+1}, \phi^{(t)}(u))$

 8:     **update DoE**: add (x^{t+1}, u^{t+1}) and its costly evaluation $y(x^{t+1}, u^{t+1})$ to the DoE (X, U).
 9:     $t \leftarrow t + 1$
10: **end while**
11: **return** $(x^\star, u^\star) = \arg\min_{(X, U)} y(x, u)$

Algorithm 3 gathers all these changes and is called ALV-EGO. The essential difference between this ALV-EGO algorithm and the vanilla counterpart (Algorithm 2) is that the EI maximization step is constrained so that the link between the discrete variables and the relaxed latent variables (hence the continuous x) is not lost and left to the pre-image step. The coupling between the continuous and the discrete variables is better accounted for. However, a pre-image step (line 7) is still necessary to fully recover a discrete solution in cases when the constraint is relaxed ($\epsilon > 0$). In ALV-EGO like in the vanilla LV-EGO, there are $q = 2$ continuous latent variable per discrete variable.

The global and local dual schemes are further detailed in Algorithms 4 and 5. The continuous minimizations of the Augmented Lagrangians once the Lagrange multipliers are set are always done with 10 random restarts of the COBYLA algorithm Powell [23]. They occur in Algorithm 4, line 4 and Algorithm 5 line 5. To allow comparisons, this implementation is identical to the EI maximization of the vanilla LV-EGO (step 5 of Algorithm 2).

Algorithm 4 Global dual scheme (makes ALV-EGO-g when used in Algorithm 3)

Ensure: An estimation of the solution to the relaxed acquisition problem (2.4)

Require: $f^{(t)}()$, an objective function, $g^{(t)}()$, a constraint

　　$N'_{\text{DoE}}, N_\lambda, N_\rho > 0$

1: Calculate a DoE $(X', L') \in (\mathcal{X}, \mathcal{L})^{N'_{\text{DoE}}}$.

　　Half of the points are feasible by *i)* sampling a $u \in \mathcal{U}$ and *ii)* setting $\ell' = \phi^{(t)}(u)$

2: Create a grid of Lagrange multipliers and penalty parameters, $(\boldsymbol{\lambda}, \boldsymbol{\rho}) = \{\lambda_1, \ldots, \lambda_{N_\lambda}\} \times \{\rho_1, \ldots, \rho_{N_\rho}\}$, with $\lambda_i \geq 0$ and $\rho_j \geq 0$ for all i, j

3: Approximately solve the dual problem by enumeration:

　　ρ_t smallest $\rho \in \boldsymbol{\rho}$ that yields a feasible solution, $g(\ell^t) \leq 0$ where

　　　　$(\lambda_t, x', \ell') = \arg\max_{\lambda \in \boldsymbol{\lambda}} \min_{(x,\ell) \in (X', L')} L_A^{(t)}(x, \ell; \lambda, \rho)$

4: Fine tune the next candidate: $(x^{t+1}, \ell^{t+1}) = \arg\min_{(x,\ell) \in (\mathcal{X}, \mathcal{L})} L_A^{(t)}(x, \ell; \lambda_t, \rho_t)$

5: **return** x^{t+1}, ℓ^{t+1}

Algorithm 5 Local dual scheme (makes in ALV-EGO-l when used in Algorithm 3)

Ensure: An estimation of the solution to the relaxed acquisition problem (2.4)

Require: $f^{(t)}()$, an objective function, $g^{(t)}()$, a constraint

　　initial values of the Lagrange multiplier and penalty, $\lambda_{N_{\text{DoE}}} = 0$ and $\rho_{N_{\text{DoE}}} = 1$, t

1: **if** $t > N_{\text{DoE}}$ **then**

2:　　{when $t = N_{\text{DoE}}$ the initial $\lambda_{N_{\text{DoE}}}, \rho_{N_{\text{DoE}}}$ are used}

　　　Update λ according to Eq. (2.10)

　　　$\lambda_t = \lambda_{t-1} + \rho_{t-1}\left(g^{(t-1)}(\ell^t) + \max(0, \frac{-\lambda_{t-1}}{\rho_{t-1}} - g^{(t-1)}(\ell^t))\right)$

3:　　Update ρ according to Eq. (2.11)

　　　$\rho_t = \rho_{t-1}$ if $g^{(t-1)}(\ell^t) \leq 0$, $2\rho_{t-1}$ otherwise

4: **end if**

5: $(x^{t+1}, \ell^{t+1}) = \arg\min_{(x,\ell) \in (\mathcal{X}, \mathcal{L})} L_A^{(t)}(x, \ell; \lambda_t, \rho_t)$

6: **return** x^{t+1}, ℓ^{t+1}

While the local update of λ and ρ might seem less robust, it is the most common implementation and it might be sufficient for the constrained EI maximization. Indeed, between two iterations, the EI changes only locally around the current iterate. Providing the latent mapping functions do not change too much, a local update of λ and ρ seems appropriate. The numerical complexity of the ALV-EGO-g and -l algorithms is essentially the same as that of the vanilla LV-EGO, cf. Table 1. The global dual scheme has a slight extra-cost because of the search for the Lagrange multiplier and penalty parameter that require N'_{DoE} extra GP predictions.
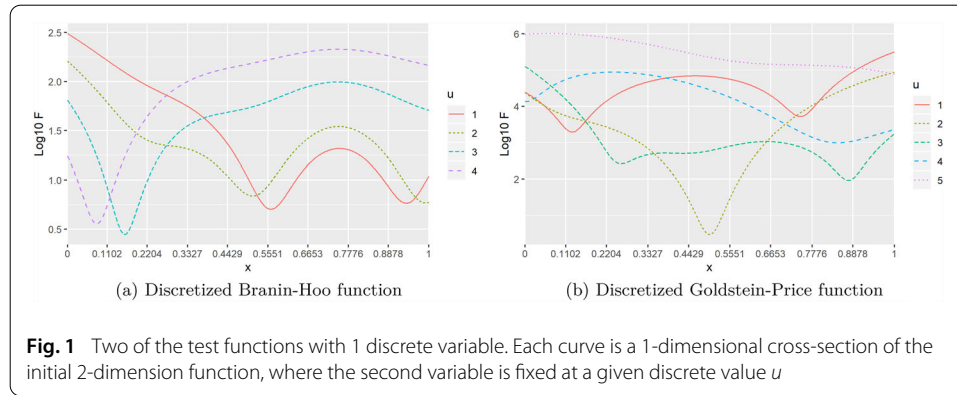
Eventually, four variants of ALV-EGO are considered, ALV-EGO-ge or -gi or -le or -li where g stands for global, l for local, e for equality ($\epsilon = 0$) and i for inequality ($\epsilon > 0$).

Description of the numerical experiments

Algorithms tested

The various algorithms tested are summarized in the Table 2 which provides their names, the type of formulation for the mixed variables, the type of metamodel, the acquisition criterion and the technique to optimize the acquisition criterion. The two possible formulations for the mixed variables are either by searching in a mixed space (MS) or by a formulation in latent variables (LV). All Gaussian processes (GPs) are built with the `kerpg`

Table 2 Summary of the 9 algorithms tested: name, space over which it is defined (mixed versus continuous with latent variables), metamodel used, acquisition criterion, optimizer of the acquisition criterion

Name	Formulation	Metamodel	Acq. crit.	Optimizer of the acq. crit.
LV-EGO	LV	GP	EI	Restarted COBYLA
LV-RFO	LV	`randomForest` toolbox	EI	Focus-search (from `mlrMBO`)
ALV-EGO-ge or -gi	LV	GP	EI	DoE (for λ_t and ρ_t) and restarted COBYLA
ALV-EGO-le or -li	LV	GP	EI	Restarted COBYLA
MS-RFO	MS	`randomForest` toolbox	EI	Focus-search (from `mlrMBO`)
MS-ES	MS	none	$-y(x, u)$	Evolution strategy (from Li et al. [15] in `CEGO` implementation Zaefferer [30])
MS-MKES	MS	GP (sym. compound disc. kernel)	EI	Evolution strategy (from Li et al. [15] in `CEGO` implementation Zaefferer [30])

(a) Discretized Branin-Hoo function   (b) Discretized Goldstein-Price function

Fig. 1 Two of the test functions with 1 discrete variable. Each curve is a 1-dimensional cross-section of the initial 2-dimension function, where the second variable is fixed at a given discrete value u

package Deville et al. [7]. The meaning of the acronyms is: LV-EGO, Latent Variables EGO; LV-RFO, Latent Variables Random Forest Optimization ; ALV-EGO-ge/-gi/-le/-li, Augmented Lagrangian Latent Variables global/local dual scheme with equality/inequality pre-image constraints; MS-RFO, Mixed Space search with Random Forest Optimization; MS-ES, Mixed Space search with Evolution Strategy; MS-MKES, Mixed Space search with Mixed Kriging metamodel and Evolution Strategy.

The different algorithms will be tested on the suite of test problems described hereafter.

Test cases

There are 3 analytical test cases and a beam bending problem. The analytical test cases have all been designed by discretizing some of the variables of classical multimodal continuous test functions. The following notation is introduced to describe the discretization: if the continuous variable x_i is discretized with u_j that takes values in $\{1, \ldots, m_j\}$, then $u_j(k) = \beta$ means $x_i = \beta$ when $u_j = k$, β a scalar, $1 \leq k \leq m_j$.

Test case 1: discretized Branin function. We modified the 2 dimensional *Branin-Hoo* function whose expression is

$$y(x_1, x_2) = (x'_2 - bx'^2_1 + cx'_1 - r)^2 + s(1 - t)\cos(x'_1) + s,$$
$$x' = x'^{\min} + (x'^{\max} - x'^{\min}) \times x$$

where $b = 5/(4\pi^2)$, $c = 5/\pi$, $r = 6$, $s = 10$, $t = 1/(8\pi)$, $x'^{\min} = [-5; 0]$, $x'^{\max} = [10; 15]$ by keeping x_1 continuous in $[0; 1]$ and making x_2 discrete with 4 levels $\{u(1) = 0; u(2) = 0.333; u(3) = 0.666; u(4) = 1\}$. The discretized Branin, which was already used in Zhang et al. [32], has several local minima as shown in Figure 1a.

The global optimum is located at $(x_1^{\star}, u^{\star}) = (0.182; u(3))$ with $y(x_1^{\star}, u^{\star}) = 2.791$.

Test case 2: discretized Goldstein function. As a second test case, the continuous Goldstein function

$$
\begin{aligned}
y(x_1, x_2) &= [1 + (x'_1 + x'_2 + 1)^2(19 - 14x'_1 + 3x_1'^2 - 14x'_2 + 6x'_1 x'_2 + 3x_2'^2)] \\
&\times [30 + (2x'_1 - 3x'_2)^2(18 - 32x'_1 + 12x_1'^2 + 48x'_2 - 36x'_1 x'_2 + 27x_2'^2)], \\
x' &= x'^{\min} + (x'^{\max} - x'^{\min}) \times x \quad, \quad x'^{\min} = [-2, -2], \ x'^{\max} = [2, 2]
\end{aligned}
$$

is partly discretized by replacing x_2 by u with 5 levels $\{u(1) = 0; u(2) = 1/2; u(3) = 1/2; u(4) = 3/4; u(5) = 1\}$. The discretized Goldstein, which has also been studied in Zhang et al. [32], is drawn in Figure 1b. It has several local optima. The global optimum is located at $(x_1^{\star}, u^{\star}) = (0.5; u(2))$ with $y(x_1^{\star}, u^{\star}) = 3$.

Test case 3: discretized Hartman function. Two variables are discretized in the 6 dimensional Hartman function,

$$
y(x) = -\sum_{i=1}^{4} \alpha_i \exp\left(-\sum_{j=1}^{d} A_{ij}(x_j - P_{ij})\right),
$$

where $x \in [0, 1]^d$, $d = 6$, $\alpha = [1, 1.2, 3, 3.2]^{\top}$ and

$$
A = \begin{pmatrix} 10 & 3 & 17 & 3.5 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{pmatrix}, \ P = 10^{-4} \begin{pmatrix} 1312 & 1696 & 5569 & 124 & 8283 & 5886 \\ 2329 & 4135 & 8307 & 3736 & 1004 & 9991 \\ 2348 & 1451 & 3522 & 2883 & 3047 & 6650 \\ 4047 & 8828 & 8732 & 5743 & 1091 & 381 \end{pmatrix}.
$$

The variables x_5 and x_6 are discretized with 5 and 4 levels respectively such that $\{u_1(1) = 0.350; u_1(2) = 0.257; u_1(3) = 0.477; u_1(4) = 0.312; u_1(5) = 0.657\}$ and $\{u_2(1) = 0.150; u_2(2) = 0.657; u_2(3) = 0.512; u_2(4) = 0.741\}$. Again, there are multiple local minima and the global optimum is located at $(x^{\star}, u^{\star}) = (0.202; 0.150; 0.477; 0.275; u_1(4), u_2(2))$ with $y(x^{\star}, u^{\star}) = -3.322$.

Euler-Bernoulli beam bending problem. This test case corresponds to an horizontal beam that is clamped at one end and subject to a vertical force at the other end. If the length of the beam is sufficiently long compared to the dimensions of its cross section, and if it is operating within its linear elastic range, the final beam deflection y (to be minimized) is expressed as

$$
D(L, S, \tilde{I}) = \frac{PL^3}{3 E S^2 \tilde{I}} \tag{3.1}
$$

where $P = 600N$ is the vertical load, $E = 600GPa$ is the Young's modulus, $L \in [10, 20]$ is the horizontal length of the beam, $S \in [1, 2]$ is the cross-section area and $\tilde{I} = I/S^2$, $\in \{\tilde{I}(1), \tilde{I}(2), \ldots, \tilde{I}(12)\}$ is the normalized moment of inertia that can explicitly be derived for a given catalog of beam profiles. The 12 levels of the normalized moment of inertia are

$$
\tilde{I} = \{0.083; 0.139; 0.380; 0.080; 0.133; 0.363; 0.086; 0.136; 0.360; 0.092; 0.138; 0.369\}.
$$

Table 3 Dimensions and DoE size of the test cases

	n_c	n_d	m_i	N_{DoE}
Branin-Hoo	1	1	4	16
Goldstein	2	1	5	40
Hartmann	4	2	{5,4}	160
Beam Bending	2	1	12	96

$$\tag{3.2}$$

We are interested in finding the best compromise between a minimization of the vertical deflection and the total weight, as expressed in the objective

$$y(x_1, x_2, u_1) = D(L, S, \tilde{I}) + \alpha L S, \tag{3.3}$$

$$\text{where } L = 10 + 10 \times x_1, \ S = 1 + x_2, \ u_1 = \tilde{I}, \tag{3.4}$$

$$\text{and } (x_1, x_2) \in [0, 1]^2. \tag{3.5}$$

Here α is the weight balancing the two effects in the objective function. It is chosen as $\alpha = 60$ so that y has several local minima and only one global minimum. This global solution is $(x_1^\star, x_2^\star, u_1^\star) = (0; 0.43; \tilde{I}(3))$ with output $y^\star = 1.287385 \times 10^3$.

Experiments setup and metrics

The optimization of each pair of algorithm and test case are repeated 50 times from different initial DoEs. The DoEs are generated by minimax Latin Hypercube Sampling. The size of the DoEs is $N_{\mathrm{DoE}} = 4 \times n_c \times n_d \times \max(m_i)$ and a budget of $N_{\mathrm{DoE}} + 50$ evaluations of the true objective function. Remember that the true objective function is supposed to be computationally intensive although it is not in these experiments so that runs can be repeated. The evolution strategies are stopped after $N_{\mathrm{DoE}} + 50$ evaluations of the true function, like the other algorithms.

The internal local optimizer, COBYLA, is restarted 5 times during the likelihood maximization and 10 times during the maximization of the acquisition criterion. The focus-search algorithm has a sample size of 1000 with 5 boundary reduction iterations and 3 multi-starts, for a total of 3000 calls to the acquisition criterion.

A summary of the dimensions involved in the different examples is given in Table 3.

Results and discussion

The results are provided with 4 main metrics. The performance of an algorithm is classically described by the median objective function over the 50 repeated runs, calculated at each iteration. The associated measure of dispersion of the performance is the interquartile over the repetitions as a function of the iteration. To discriminate between methods that are rapid but provide rough solutions from the ones that take more time but yield better solutions, the two other metrics are based on the definition of targets. For each test case, a target is a given quantile of all the objectives functions found by all the algorithms throughout all the repetitions. A 10% target is difficult, while a 50% target is the median performance. The third metric is the iteration number at which the median objective function of a given algorithm reaches a given target. The fourth metric is the success rate (given a target), which is the percentage of the runs that do better than the target. The metrics associated to the quantile targets have the advantage that they are normalized with

respect to the test cases: thanks to the quantiles, the definitions of an easy, a median or a hard target stands accross the different functions to optimize. The target-based metrics will later be averaged over the different test cases.

Let us now review the performances of the algorithms on each test case.

Analytical test functions

Branin function Figure 2 presents the results for the Branin function with the four metrics. On the top left plot, showing the median value for the objective function, it is clear that the two methods that rely on the random forest metamodel (MS-RFO and LV-RFO) are overtaken by all other methods. This indicates that, whether in the mixed or in the latent-augmented space, random forests do not represent sufficiently well the Branin function in comparison to Gaussian processes. Looking at Fig. 2b, it is observed that the fast methods typically have the lowest spread in performance and vice versa. This is expected as non converging runs may yield a wide range of performances. All methods involving the discrete constraint (i.e., the augmented Lagrangians) managed to improve over the LV-EGO performance; and including a mixed metamodel increased significantly the success rate and the median solution for the evolutionary strategy.
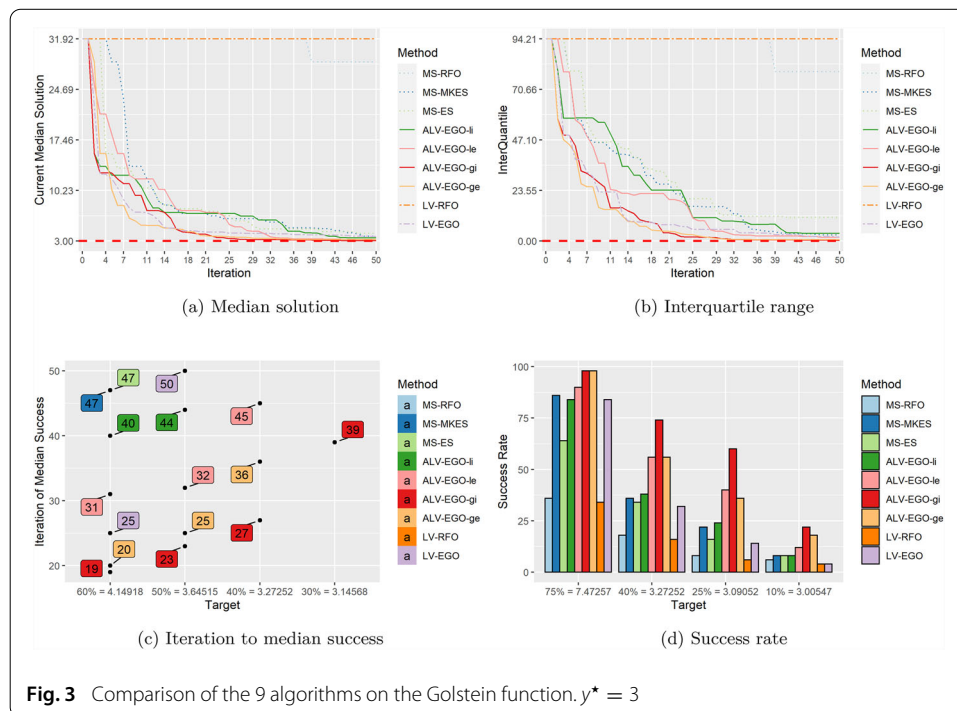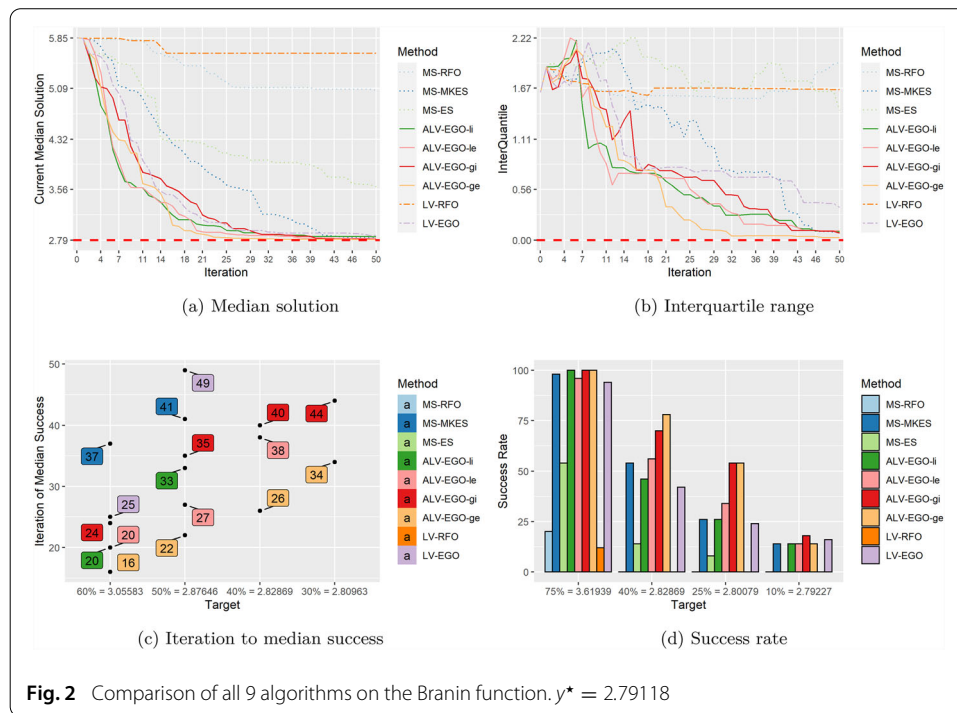
Regarding the success rate on Fig. 2d, the methods MS-MKES, LV-EGO, ALV-EGO-li, -le, -ge and -gi were the most prominent, the latter being capable to reach success rates of about 20% for a 10% target. Notice that all these methods contain Gaussian processes. Indeed, the Branin function is easy to represent by a GP whether continuous or mixed. In the same vein, MS-MKES which differs from MS-ES by the use of a GP, clearly benefits from that metamodel.

All ALV- methods, which account for the discrete constraint, obtained the best median performances. ALV-EGO-ge in particular found all targets, in the median sense, earlier than the other algorithms as can be seen from Fig. 2c.

A last comment is necessary regarding the bottom of Fig. 2: the plot on the left describes the median performance (in terms of targets reached) while the right plot counts the success rate at reaching a target over all runs. Therefore, some targets are reached on the right by some of the runs of a given algorithm, while they are never atteined on the left by the median of the same algorithm. This comment stands accross all test cases.

Goldstein function The experiments done with the Goldstein test function are summed up in Fig. 3. Like with the Branin function, algorithms relying on random forests (LV-RFO and MS-RFO) showed both poor performance (top left plot). The associated high constant interquartile (top right) is that of the best points in the initial designs, which remains unchanged since no better point is found by these algorithms.

Considering the success rates for all targets (bottom plots), it is seen that accounting for the discreteness through a constraint (which is the distinctive feature of ALV- methods) is useful with the Goldstein function: like with Branin, ALV-EGO-gi is the best performer, but the other ALV- follow and outperform LV-EGO. All ALV- strategies almost reach the absolute target of percentile 25% with a rate of 25% or higher. The comparison of the plots Fig. 3c, d also shows that, behind the ALV- methods, LV-EGO has a good median performance (cf. Fig. 3c) but more of the MS-MKES searches manage to find difficult targets (the 25% and 10% quantiles).

Fig. 2 Comparison of all 9 algorithms on the Branin function. $y^\star = 2.79118$

Fig. 3 Comparison of the 9 algorithms on the Golstein function. $y^\star = 3$

Hartmann function Results on the Hartmann function which has 4 continuous and 2 discrete variables, with a total of 9 discrete levels, will be impacted by the sensitivity of the algorithms to an increase in dimension. These results are reported in Fig. 4.

LV-EGO stands out as the best method with respect to all criteria for Hartmann. The next two best methods are LV-RFO and ALV-EGO-gi, followed by MS-RFO and ALV-
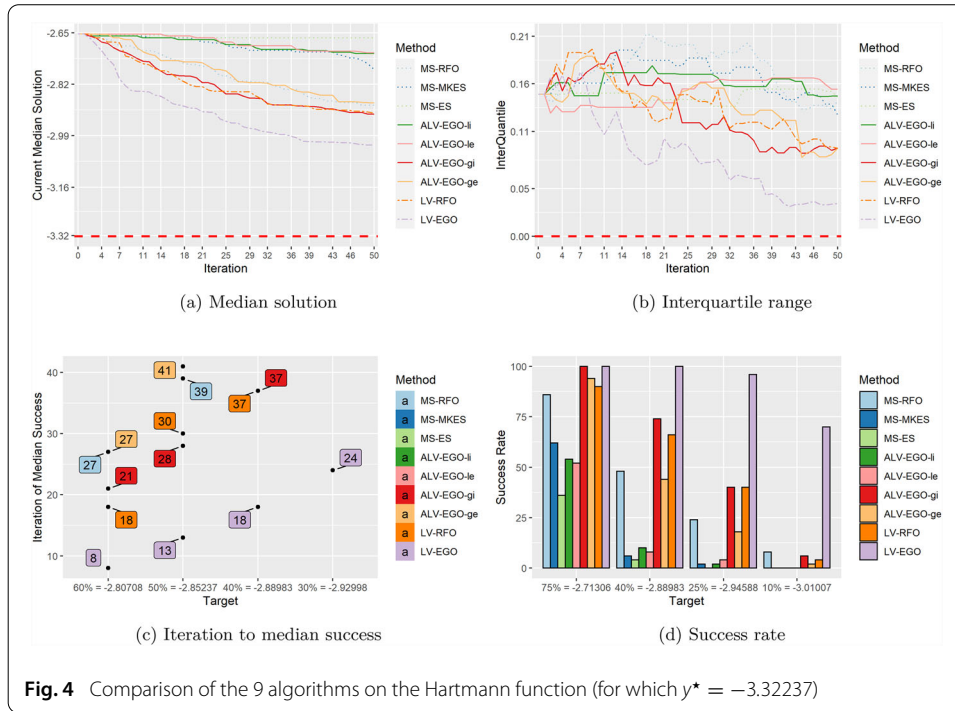
Fig. 4 Comparison of the 9 algorithms on the Hartmann function (for which $y^\star = -3.32237$)

EGO-ge. This time, LV-RFO and MS-RFO, which both rely on random forests, belong to the efficient methods: random forests gain in relative performance with respect to the GPs when the dimension and the size of the initial DoE increase. For Hartmann, LV-EGO consistently outperforms the ALV- implementations. The importance of keeping the coupling between discrete and latent variables during the optimization seems less crucial, and even somewhat detrimental, in the Hartmann case. We think that this is due to the very tight budget (50 iterations after the initial DoE) which does not allow the convergence of the optimizers, as can be seen in the Plot 4a where the global optimum is not reached. Because the optimum is not really found, constraints on discreteness are superfluous and their handling through the pre-image problem is sufficient. As in the other test cases, MS-ES was slower than the other methods.

Beam bending application

Optimization results Figure 5 summarizes the 4 comparison metrics of all 9 algorithms in the bended beam test case. The ranking of the algorithms is similar to that obtained with the Branin and Goldstein functions. LV-EGO has the best convergence both in terms of median speed (cf. plots of the left column) and accuracy (bottom right plot). ALV-EGO-gi is the second most efficient method followed by ALV-EGO-ge. Again, the algorithms that resort to random forests, LV-RFO and MS-RFO, are the slowest and most inaccurate. They share this counter-performance with MS-ES.

Latent variables in the beam application The beam subject to a bending load is a test case that allows to interpret the latent variables. Indeed, the normalized moment of inertia, $\tilde{I}$, is a candidate latent variable once it is allowed to take continuous values as it determines, with the continuous cross-section S and the length L, the output (the penalized beam deflection) y in Eq. (3.5). The levels of $\tilde{I}$ (given in Eq. (3.2)) correspond to 3 increasingly
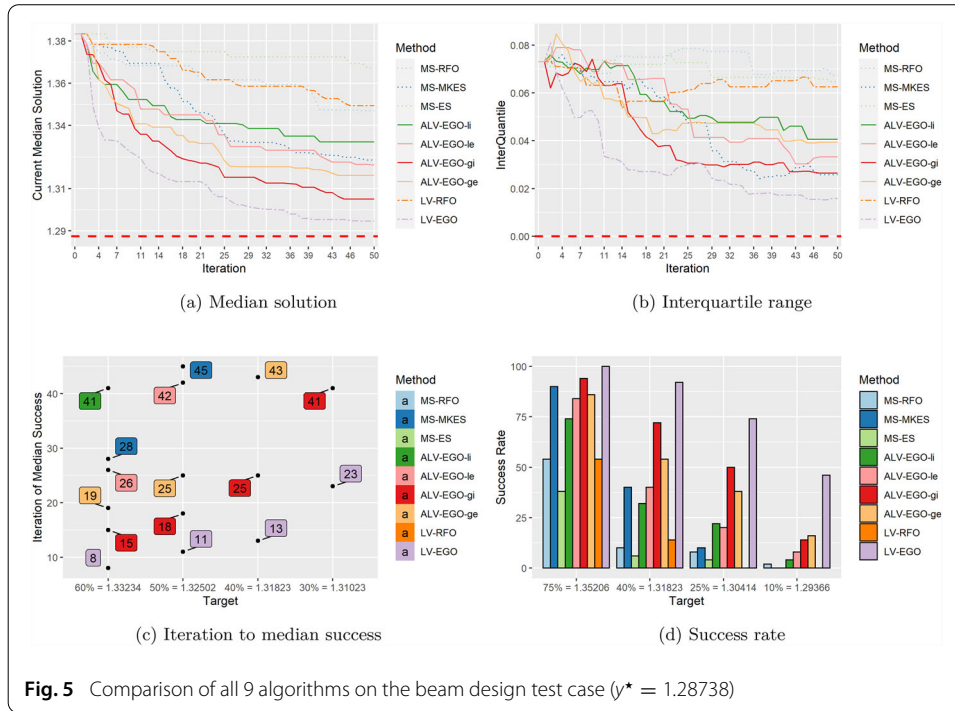
Cuesta-Ramirez *et al. Adv. Model. and Simul. in Eng. Sci.*(2022)9:6

Page 19 of 29

Fig. 5 Comparison of all 9 algorithms on the beam design test case ($y^\star = 1.28738$)
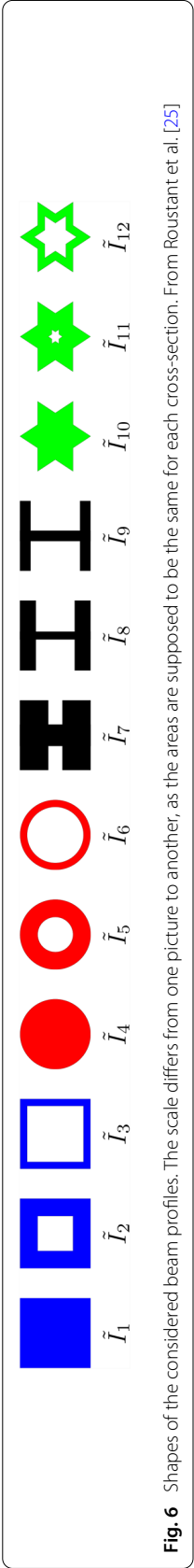
hollow profiles of 4 shapes, as illustrated in Fig. 6. Because a relaxed $\tilde{I}$ is a possible latent variable, it is expected that the latent variables $\phi^{(t)}$ learned from the data will be grouped in the same way as $\tilde{I}$. Looking at $\tilde{I}$ values and at Fig. 6, we thus expect, in the image space defined by latent variables, three groups of levels: those corresponding to solid forms (levels $\{1, 4, 7, 10\}$), medium-hollow forms (levels $\{2, 5, 8, 11\}$) and hollow forms (levels $\{3, 6, 9, 12\}$).
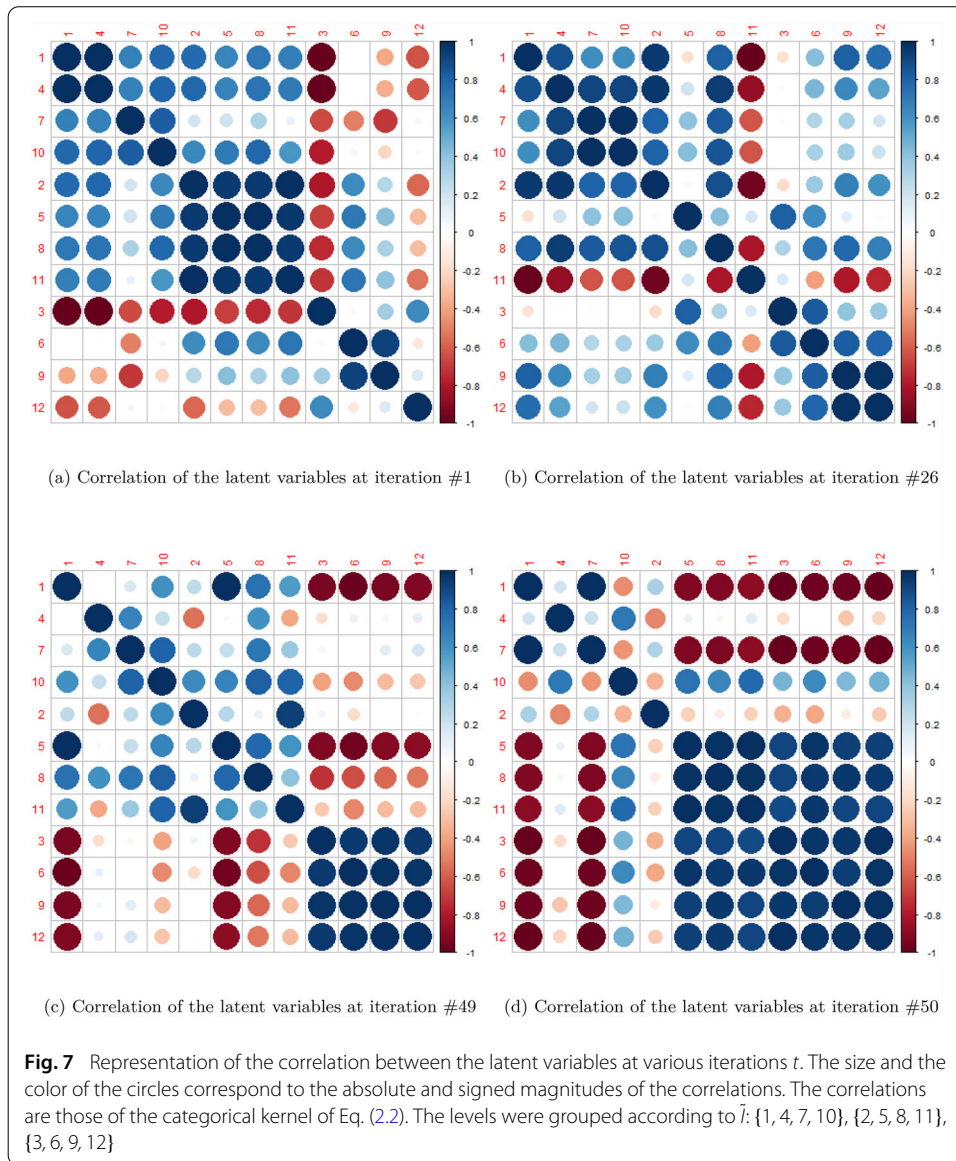
For the sake of interpretation, we select 1 run that found the global optimum with the Vanilla LV-EGO algorithm. In Fig. 7, we represent in a color scale the estimated correlation matrix corresponding to the categorical kernel of Eq. (2.2), at iterations $[1; 26; 49; 50]$. At the beginning of the optimization, at iteration 1, we can see a block-structure which corresponds quite well to the three groups of forms described above. This structure becomes less clear for the next iterations of the LV-EGO algorithm. This may be explained by the fact that the algorithm creates an unbalanced design, with more points in the promising areas according to the optimizers, so that all levels are no longer properly represented.

Summary and discussion

The results of all the previous test cases which are measured through targets can be averaged. For example, the success rate of an algorithm at 25% difficulty is the average of the rates for the 25% quantiles of all test cases. The average results are presented in Fig. 8.

The three leading algorithms out of the 9 tested are ALV-EGO-gi, -ge and LV-EGO. Among them, LV-EGO is slightly better at locating difficult targets (10% quantile) while ALV-EGO-gi (closely followed by ALV-EGO-ge) is more robust at locating 50% targets as can be seen from the median success plot in Fig. 8a. All three algorithms have in common to use latent variables. In particular, these algorithms outperformed MS-MKES which benefits from a Gaussian process but works only in the mixed space, i.e., MS-MKES does

Cuesta-Ramirez *et al. Adv. Model. and Simul. in Eng. Sci.*(2022)9:6

Page 20 of 29

Fig. 6 Shapes of the considered beam profiles. The scale differs from one picture to another, as the areas are supposed to be the same for each cross-section. From Roustant et al. [25]

Cuesta-Ramirez *et al. Adv. Model. and Simul. in Eng. Sci.*(2022)9:6

Page 21 of 29

(a) Correlation of the latent variables at iteration #1

(b) Correlation of the latent variables at iteration #26

(c) Correlation of the latent variables at iteration #49

(d) Correlation of the latent variables at iteration #50

Fig. 7 Representation of the correlation between the latent variables at various iterations t. The size and the color of the circles correspond to the absolute and signed magnitudes of the correlations. The correlations are those of the categorical kernel of Eq. (2.2). The levels were grouped according to $\tilde{l}$: {1, 4, 7, 10}, {2, 5, 8, 11}, {3, 6, 9, 12}
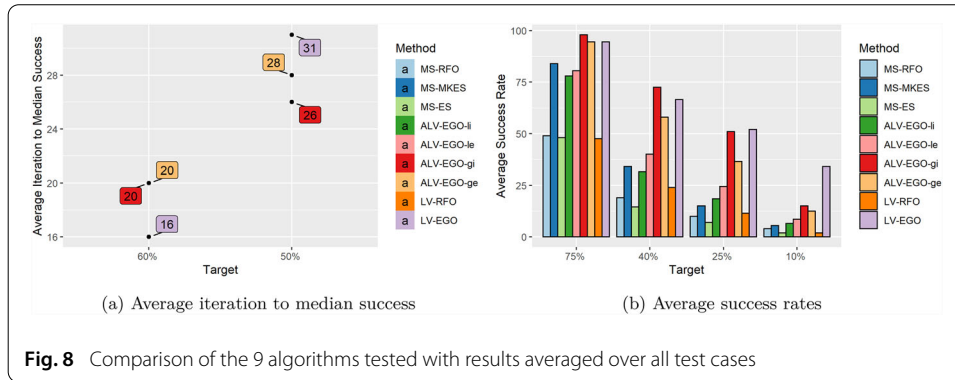
not imply latent variables. This shows that latent variables are useful to speed up a Bayesian search for mixed problems.

No clear advantage, on the average, was found for accounting for the discrete nature of the variables through constraints: LV-EGO, which ignores the link between latent variables and the discrete variables until the pre-image problem, is competitive with the best of the augmented Lagrangian ALV-EGO algorithms. We hypothesize that the constraint on latent variables, by creating disconnected feasibility islands around $\phi^{(t)}(u)$, $u \in \mathcal{U}$, makes the optimization of the acquisition criterion almost as difficult to solve as it originally was in the mixed space, therefore not allowing to fully benefit from the continuity of the $\mathcal{X} \times \mathcal{L}$ space.

In our tests, the global updating of the Lagrange multipliers was always preferable to the local counterparts, ALV-EGO-gi and -ge eclipsing ALV-EGO-li and -le.

(a) Average iteration to median success　　　　(b) Average success rates

Fig. 8 Comparison of the 9 algorithms tested with results averaged over all test cases

The ALV-EGO-gi approach, where the discrete constraint is relaxed and turned into an inequality (Eq. (2.4)), works better on the average than ALV-EGO-ge where the constraint is an equality. This illustrates the positive effect of the relaxation ϵ, that softens the phenomenon we mentionned above where the feasible domain is broken into disconnected regions.

MS-ES is consistently less efficient than the other algorithms. It was expected, because there is no metamodel to save calls to the function. Furthermore, the sampling is done in the mixed space. The optimizers based on random forests have also rather poor average performances, to the exception of the 6 dimensional Hartmann function. We believe the random forests need a sufficiently large initial DoE (which happened with a higher dimension) to fruitfully guide the search.
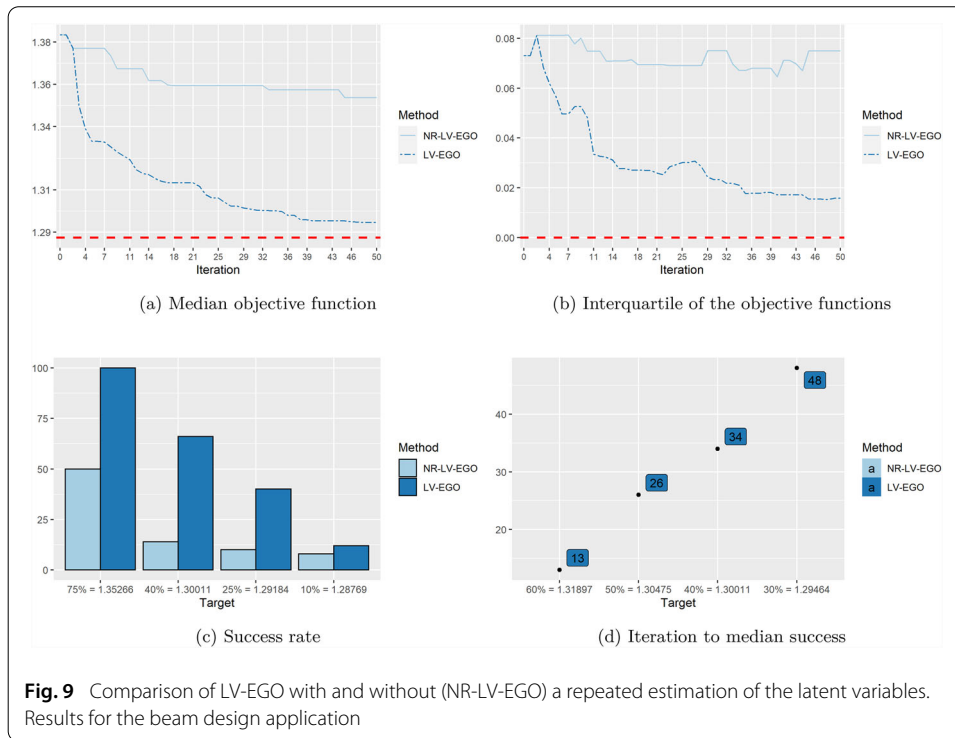
As a final comment, we discuss the necessity of re-estimating the latent variables at each iteration. The estimation of the latent variables has an important numerical cost of about $qt^3 \sum_{i=1}^{n_d} m_i$ operations at each iteration t (cf. Table 1). It was repeated at each iteration in the algorithms with latent variables considered so far. In the experiment reported in Figure 9, a version of the LV-EGO algorithm is considered where the latent variables are estimated once only, with the initial DoE, yielding the NR-LV-EGO algorithm (for Non Repeated estimation of $\phi()$).

As can be seen in Fig. 9 when comparing LV-EGO with NR-LV-EGO, the re-estimation of the latent variables at each iteration, as implemented in the LV-EGO algorithm and its ALV-EGO variants, considerably improves its performance. An accompanying result is the visualization of the correlation matrix of the discrete variable provided in Fig. 7, where one notices that the correlation (hence the latent variables) evolves in time. Our experiments indicate that this evolution is beneficial to the optimization efficiency.

Conclusions and perspectives

This work has investigated five Bayesian optimization approaches to small and medium size mixed problems that hinged on latent variables. They differed in the way the coupling between the discrete variables and their relaxed pendants, the latent variables, is implemented.

Algorithms involving latent variables were compared to other algorithms directly working in the mixed space and were found to consistently outperform them. LV-EGO and ALV-EGO-gi were more efficient (in terms of calls to the true objective function) than MS-MKES which also benefits from the Gaussian process. These first results show that

(a) Median objective function

(b) Interquartile of the objective functions

(c) Success rate

(d) Iteration to median success

Fig. 9 Comparison of LV-EGO with and without (NR-LV-EGO) a repeated estimation of the latent variables. Results for the beam design application

latent variables provide a flexible way to handle mixed problems where the total number of levels and of variables is less or equal to about 10 variables and 10 levels in total.

Accounting for the discrete nature of some variables through a constraint during the relaxed optimization with augmented Lagrangians was not clearly found to further increase the performance of the search as LV-EGO competed equally and even sometimes outperformed the ALV versions of the algorithms. It was also observed that expressing the discreteness as an inequality constraint by adding a tolerance was a better option than expressing it as an equality. The global updating strategy of the Lagrange multipliers, which to the best of our knowledge is original, improved over the more common local updating schemes. Finally, the random forests metamodels did not do as well as the Gaussian processes, whether in their continuous or mixed forms, within the Bayesian optimization algorithm.

Our study needs to be completed in three ways. To fully leverage on the continuous latent space, the gradient of the acquisition function should be analytically calculated and used to guide its maximization. The implementation we proposed creates more latent variables than there are discrete levels, which limits its application to about 10 levels. This limitation can be overcome with under-parameterized kernels based on groups Roustant et al. [25] or warping techniques Deville et al. ?[7]. Mixed Bayesian optimization through latent variables would also gain in credibility if the convergence results of EGO were generalized to it.

Cuesta-Ramirez *et al. Adv. Model. and Simul. in Eng. Sci.* (2022) 9:6

Page 24 of 29

Complements on the augmented Lagrangians

Case of an equality constraint

Let us first consider an optimization problem with an equality constraint,

$$
\begin{cases}
\min_{x \in \mathcal{X}} f(x) \\
\text{such that } h(x) = 0
\end{cases}
\tag{A.1}
$$

At this point, $f()$ and $h()$ are very general functions on a d-dimensional general set $\mathcal{X}$. We only require that $\mathcal{X}$ is not empty, that $f()$ and $h()$ are bounded, and that there is at least one solution to (A.1), $x^\star \in \mathcal{X}$, which can be attained. $f()$ and $h()$ are not necessarily continuous, a fortiori not necessarily differentiable. With respect to the main body of the article, the notations are simplified in this Section: $\mathcal{X}$ stands for the cartesian product of $\mathcal{X}$ and $\mathcal{L}$, $f(x)$ generalizes $-\log(1 + \mathrm{EI}^{(t)}(x, \ell))$ and $h(x)$ corresponds to $g^{(t)}(\ell)$ when $\epsilon = 0$. Note that $g^{(t)}()$, being made of the minimum distance to a discrete set of points (cf. Eq. (2.4)), is not differentiable. $g^{(t)}()$ is the only constraint in the article. This appendix considers one constraint too, but all the results given readily generalize to many constraints by replacing the products by vector scalar products.

Problem (A.1) can be equivalently reformulated as

$$
\begin{cases}
\min_{x \in \mathcal{X}} f(x) + \frac{1}{2}\rho h^2(x) \\
\text{such that } h(x) = 0
\end{cases}
\tag{A.2}
$$

where $\rho \geq 0$ is a penalty parameter. The two above formulations have the same solution $x^\star$ and the same value of optimal objective function since $x^\star$ is feasible, $h(x^\star) = 0$, therefore $f(x^\star) = f(x^\star) + \frac{1}{2}\rho h^2(x^\star)$. However, as proved in Minoux [17] and sketched in Fig. 10, there is always a positive lower bound on the penalty parameters, $\rho \geq \rho^\star \geq 0$, such that Problem (A.2) can be equivalently solved through the dual formulation,

$$
\begin{aligned}
&\max_{\lambda \in \mathbb{R}} D(\lambda, \rho) \\
&\text{where } D(\lambda, \rho) = \min_{x \in \mathcal{X}} L_A(x; \lambda, \rho) \\
&\text{and } L_A(x; \lambda, \rho) = f(x) + \lambda h(x) + \frac{1}{2}\rho h^2(x)
\end{aligned}
\tag{A.3}
$$

In this way, the augmented Lagrangian of Hestenes [10] is the classical Lagrangian of the penalized problem (A.2). We write $\lambda^\star, \rho^\star$ a solution to (A.3). $D(\lambda, \rho)$ is the lower front of all augmented Lagrangians for varying x at a given λ, ρ. The "global dual" update of (λ, ρ) comes from the resolution of (A.3) where the set $\mathcal{X}$ is approximated by the finite subset of samples X.

Let us denote

$$
x(\lambda, \rho) = \arg \min_{x \in \mathcal{X}} L_A(x; \lambda, \rho)
\tag{A.4}
$$

a solution at given multiplier and penalty parameter. The function $D(\lambda, \rho)$ is concave in λ and ρ and $h(x(\lambda, \rho))$ is a subgradient with respect to λ Minoux [17]. This is at the root of updating strategies that we called "local dual" earlier and which consist in a gradient step in the dual space,

$$
\lambda_{t+1} = \lambda_t + \alpha \partial_\lambda D(\lambda_t, \rho_t) = \lambda_t + \alpha h(x(\lambda_t, \rho_t)) \quad ,
\tag{A.5}
$$

where $\alpha > 0$ is a step size factor.

More specific update strategies such as those given in Nocedal and Wright [19], Picheny et al. [22] stem from the Karush Kuhn and Tucker (KKT) optimality conditions and require the additional assumption that $\mathcal{X} \in \mathbb{R}^d$ and $f()$ and $h()$ are differentiable. At $x^\star$, since $h(x^\star) = 0$ and λ^{KKT} being the KKT multiplier[1], one has

$$\nabla f(x^\star) + \rho h(x^\star) \nabla h(x^\star) + \lambda^{KKT} \nabla h(x^\star) = 0$$
$$\Rightarrow \qquad \nabla f(x^\star) + \lambda^{KKT} \nabla h(x^\star) = 0 \qquad (A.6)$$

At iteration t, the necessary conditions for $x^t = x(\lambda_t, \rho_t)$ to be the minimum of $L_A(; \lambda_t, \rho_t)$ are

$$\nabla f(x^t) + \rho_t h(x^t) \nabla h(x^t) + \lambda_t \nabla h(x^t) = \nabla f(x^t) + (\rho_t h(x^t) + \lambda_t) \nabla h(x^t) = 0 \qquad (A.7)$$

Comparing Eqs. (A.6) and (A.7), x^t can be driven to $x^\star$ if

$$\lambda_{t+1} = \lambda_t + \rho_t h(x^t) \qquad (A.8)$$

The updates (A.5) and (A.8) have the same form, (A.8) is more restrictive since the KKT conditions must apply but the step size is known.

The equality constraint of the article (Eq. (2.4) with $\epsilon = 0$) is a minimum over distances. It has the additional feature that it is always positive or null, $\forall x \in \mathcal{X}$, $h(x) \geq 0$. Because of this, if h is locally differentiable around $x^\star$, $\nabla h(x^\star) = 0$ since h has a minimum at $x^\star$. The constraint qualification condition is not satisfied ($\nabla h(x^\star)$ does not span a non-empty set) and the KKT conditions do not apply. Another consequence is that the optimal Lagrange multiplier must be positive and the search for λ can be written $\max_{\lambda \geq 0} D(\lambda, \rho)$ in Problem (A.3), as in Problem (2.7).

Proof Assume ρ is large enough for Problem (A.2) to have a saddle point at its optimum, $f(x^\star) \leq f(x) + \rho/2 h^2(x) + \lambda^\star h(x)$, $\forall x$ where $\lambda^\star$ is the optimum Lagrange multiplier. Since the optimization problem has an active constraint, there is a point x^I that is infeasible, $h(x^I) > 0$, and has a better objective function than the feasible solution (otherwise the constraint is useless), $f(x^I) + \frac{\rho}{2} h^2(x^I) \leq f(x^\star)$. If the optimum Lagrange multiplier is negative, $\lambda^\star < 0, f(x^I) + \frac{\rho}{2} h^2(x^I) + \lambda^\star h(x^I) < f(x^\star)$ which contradicts the fact that $x^\star$ is a solution to the dual problem.   □
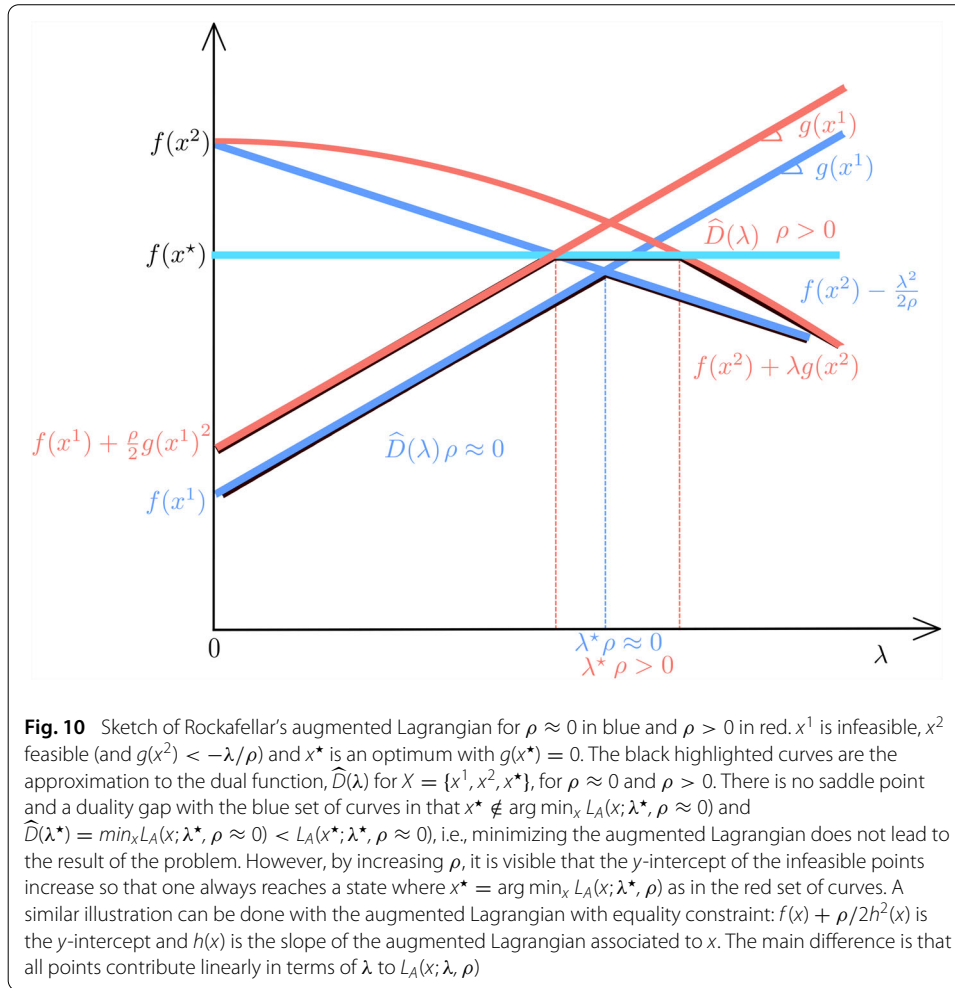
Inequality constraint

When $\epsilon > 0$, Problem (2.4) has an inequality constraint which we rewrite here more simply,

$$\begin{cases} \min_{x \in \mathcal{X}} f(x) \\ \text{such that } g(x) \leq 0 \end{cases} \qquad (A.9)$$

The considerations on augmented Lagragian done above for equality constraints readily extend to inequality constraints by introducing a slack variable,

---

[1] The Lagrange multiplier that maximizes the dual function is equal to the KKT multiplier only when the functions are differentiable, the constraints qualification conditions apply, and there is a saddle point i.e., $min_x max_\lambda L_A(x; \lambda, \rho) = max_\lambda min_x L_A(x; \lambda, \rho)$.

Fig. 10 Sketch of Rockafellar's augmented Lagrangian for $\rho \approx 0$ in blue and $\rho > 0$ in red. x^1 is infeasible, x^2 feasible (and $g(x^2) < -\lambda/\rho$) and $x^\star$ is an optimum with $g(x^\star) = 0$. The black highlighted curves are the approximation to the dual function, $\widehat{D}(\lambda)$ for $X = \{x^1, x^2, x^\star\}$, for $\rho \approx 0$ and $\rho > 0$. There is no saddle point and a duality gap with the blue set of curves in that $x^\star \notin \arg\min_x L_A(x; \lambda^\star, \rho \approx 0)$ and $\widehat{D}(\lambda^\star) = \min_x L_A(x; \lambda^\star, \rho \approx 0) < L_A(x^\star; \lambda^\star, \rho \approx 0)$, i.e., minimizing the augmented Lagrangian does not lead to the result of the problem. However, by increasing ρ, it is visible that the y-intercept of the infeasible points increase so that one always reaches a state where $x^\star = \arg\min_x L_A(x; \lambda^\star, \rho)$ as in the red set of curves. A similar illustration can be done with the augmented Lagrangian with equality constraint: $f(x) + \rho/2 h^2(x)$ is the y-intercept and $h(x)$ is the slope of the augmented Lagrangian associated to x. The main difference is that all points contribute linearly in terms of λ to $L_A(x; \lambda, \rho)$

$$\begin{cases} \min_{x,s\in\mathcal{X}\times\mathbb{R}} f(x) \\ \text{such that } g(x) + s^2 = 0 \end{cases} \tag{A.10}$$

and the expression for the augmented Lagrangian (A.3) becomes

$$L_A(x, s; \lambda, \rho = f(x) + \lambda(g(x) + s^2) + \frac{1}{2}\rho(g(x) + s^2)^2 \tag{A.11}$$

The minimization of $L_A()$ on the slack variable s can be done analytically:

$$\frac{\partial L_A(x, s; \lambda, \rho)}{\partial s} = 0 \iff s^2 = -\frac{\lambda}{\rho} - g(x)$$

Since s^2 needs to be positive, all cases are summed up in

$$s^2 = \max\left(0, -\frac{\lambda}{\rho} - g(x)\right) \tag{A.12}$$

Reinjecting the expression of s^2 into the augmented Lagrangian yields

$$L_A(x; \lambda, \rho) = f(x) + \frac{1}{2\rho}\left[(\max(0, \lambda + \rho g(x)))^2 - \lambda^2\right] \tag{A.13}$$

which is equivalent to the expression of Rockafellar with the 2 cases given in Eq. (2.5) (recall $-\log(1 + EI)$ is $f(x)$).

The update equations for λ are the same as those for the equality case where the slack variable s^2 takes its optimal value. On the one hand, it is possible to solve the approximated dual problem as in (2.8). On the other hand, a step along a subgradient in the dual space can be taken,

$$
\begin{aligned}
\lambda_{t+1} &= \lambda_t + \alpha(g(x^t) + s_t^2) \\
\Rightarrow \lambda_{t+1} &= \lambda_t + \alpha\left(g(x^t) + \max(0, -\frac{\lambda}{\rho} - g(x^t))\right)
\end{aligned}
\tag{A.14}
$$

where α is again a positive step factor. It has the same form as Eq. (2.10). The update (2.10) is fully recovered from the KKT conditions as above for equalities, (A.8),

$$
\begin{aligned}
\lambda_{t+1} &= \lambda_t + \rho(g(x^t) + s_t^2) \\
\Rightarrow \lambda_{t+1} &= \lambda_t + \rho\left(g(x^t) + \max(0, -\frac{\lambda}{\rho} - g(x^t))\right)
\end{aligned}
\tag{A.15}
$$

Equations (A.14) and (A.15) are the same but in the latest the step factor α is known and equal to ρ, which comes at the additional expense of the KKT validity conditions.

Abbreviations
ALV: Augmented Lagrangian latent Variable; DoE: Design of Experiment; $D()$, $\widehat{D}()$: dual and approximate dual functions; MLE: Maximum Likelihood Estimation; ϵ: relaxation constant for the discreteness constraint; E[·];mathematical expectation EGO: Efficient Global Optimization algorithm; EI(), EI(x,ℓ), EI$^{(t)}$(x, ℓ): expected improvement (at point (x, ℓ) and iteration t); ES: Evolution Strategy optimization algorithm; ϕ , $\phi^{(t)}$: Vector of latent mapping functions stemming from MLE maximization (at iteration t), $\phi^{(t)} : u \in \mathcal{U} \to \phi^{(t)}(u) \in \mathcal{L}$; $f()$, $f^{(t)}()$: Non-costly objective function to minimize, based on the GP, typically $-\log(1 + \text{EI}^{(t)}())$; $g()$, $g^{(t)}()$: inequality (≤ 0) or equality constraint function; GP: Gaussian process; λ, λ_t: Lagrange multiplier; ℓ : vector of relaxed latent variables. They take a value in $\mathcal{L}$; LV: Latent Variable; MK: Mixed Kriging, a GP indexed on mixed variables; MS: Mixed Space formulation (as opposed to relaxed with latent variables); m, m_j: number of levels for all discrete variables or for the discrete variable u_j; n_c: number of continuous variables; n_d: number of discrete variables; n_ℓ: total number of latent variables for all discrete variables, in this article $= n_d q$; q: number of latent variables per discrete variable, in this article $= 2$; ρ, ρ_t: penalty parameter; RFO: Random Forest Optimization algorithm; t, $^{(t)}$: number of calls to the expensive objective function, superscript for functions redefined at each iteration (depending on the GP); u: vector of discrete (ordinal or nominal) variables, $\in \mathcal{U}$; U: set of the discrete part of already evaluated points, $\in \mathcal{U}^t$; x: vector of continuous variables, $\in \mathcal{X} \subset \mathbb{R}^{n_c}$; X: set of the continuous part of already evaluated points, $\in \mathcal{X}^t \subset \mathbb{R}^{n_c t}$; $y(,)$: "costly" objective function to minimize, typically based on a numerical simulation; Y: current set of outputs of the evaluated points, $\in \mathbb{R}^t$.

Author contributions
The kernels with latent variables were developed jointly by OR, GP and JC-R. The Bayesian optimization formulation was developed jointly by RLR, JC-R, OR, GP and CD. The augmented Lagrangians schemes were developped jointly by RLR and JC-R. The test cases were proposed by GP, CD, AG and JC-R. JC-R did the code implementation. All authors reviewed the manuscript. All authors read and approved the final manuscript.

Availibility of data and materials
The source code of this work will be made available upon request to the corresponding author.

Declarations

Competing interests
The authors declare no competing interests.

Author details
[1]CEA, LETI, Univ. Grenoble Alpes, 38000 Grenoble, France, [2]INSA Toulouse, Toulouse, France, [3]LIMOS (CNRS, Mines Saint-Etienne, UCA), Saint-Etienne, France, [4]COSYS, Université Gustave Eiffel, 77420 Champs-sur-Marne, France, [5]CEA-DAM, Paris, France.

References
1.　Audet C, Dennis Jr John E. Pattern search algorithms for mixed variable programming. SIAM J Optim. 2001;11(3):573–94.

2. Bartz-Beielstein T, Filipič B, Korošec P, Talbi EG. High-Performance Simulation-Based Optimization. Studies in Computational Intelligence. Springer International Publishing, 2019. ISBN 9783030187644. https://books.google.fr/books?id=8yGbDwAAQBAJ.
3. Bartz-Beielstein T, Zaefferer Martin. Model-based methods for continuous and discrete global optimization. Appl Soft Comput. 2017;55:154–67.
4. Belotti P, Kirches C, Leyffer S, Linderoth J, Luedtke J, Mahajan Ashutosh. Mixed-integer nonlinear optimization. Acta Numer. 2013;22:1–131.
5. Bischl B, Richter J, Bossek J, Horn D, Thomas J, Lang M. mlrMBO: A modular framework for model-based optimization of expensive black-box functions, 2018.
6. Cao YJ, Jiang L, Wu QH. An evolutionary programming approach to mixed-variable optimization problems. Appl Math Model. 2000;24(12):931–42.
7. Deville Y, Ginsbourger D, Roustant O, Durrande N. kergp. https://cran.r-project.org/package=kergp, 2017–2021.
8. Emmerich Michael, Zhang A, Li R, Flesch I, Lucas Peter J. Mixed-integer Bayesian optimization utilizing a-priori knowledge on parameter dependences. J Phys Chem A. 2008;65–72.
9. Frazier Peter I. A Tutorial on Bayesian Optimization. *arXiv e-prints*, page arXiv:1807.02811, July 2018.
10. Hestenes Magnus R. Multiplier and gradient methods. J Optim Theory Appl. 1969;4(5):303–20.
11. Hutter F, Hoos HH, Leyton-Brown K. Sequential model-based optimization for general algorithm configuration. In: International Conference on Learning and Intelligent Optimization, pp. 507–523. Springer, 2011.
12. Jones DR, Schonlau M, Welch WJ. Efficient global optimization of expensive black-box functions. J Glob Optim. 1998;13(4):455–92. https://doi.org/10.1023/A:1008306431147 (**ISSN 1573-2916**).
13. Le Riche R, Frédéric G. Dual evolutionary optimization. Lecture Notes in Computer Science, (2310): 281–294. selected papers of the 5th Int. Evolution Artificielle Conf; 2002.
14. Le Riche R, Picheny V. Revisiting bayesian optimization in the light of the coco benchmark. Struct MultiDiscip Optim. 2021. to appear.
15. Li R, Emmerich MTM, Eggermont J, Bäck T, Schütz M, Dijkstra J, Reiber Johan HC. Mixed integer evolution strategies for parameter optimization. Evol Comput. 2013;21(1):29–64.
16. Lin Y, Liu Y, Chen W-N, Zhang J. A hybrid differential evolution algorithm for mixed-variable optimization problems. Inform Sci. 2018;466:170–188. ISSN 00200255. https://doi.org/10.1016/j.ins.2018.07.035. URL https://linkinghub.elsevier.com/retrieve/pii/S0020025516318163.
17. Minoux M. Mathematical programming: theory and algorithms. Wiley: A Wiley-Interscience publication; 1986.
18. Mockus J, Tiesis V, Zilinskas Antanas. The application of Bayesian methods for seeking the extremum. Towards Glob Optim. 1978;2(117–129):2.
19. Nocedal J, Wright SJ. Numerical optimization. Springer series in operations research. Springer, New York, 2nd edn, 2006. ISBN 978-0-387-30303-1. OCLC: ocm68629100.
20. Ocenasek J, Schwarz J. Estimation of distribution algorithm for mixed continuous-discrete optimization problems. In: 2nd Euro-International Symposium on Computational Intelligence. pp. 227–232. IOS Press Kosice, Slovakia, 2002.
21. Pelamatti J, Brevault L, Balesdent M, Talbi E-G, Guerin Y. Efficient global optimization of constrained mixed variable problems. J Glob Optim. 2019;73(3):583–613. https://doi.org/10.1007/s10898-018-0715-1 (**ISSN 0925-5001, 1573-2916**).
22. Picheny V, Gramacy RB, Wild S, Le Digabel S. Bayesian optimization under mixed constraints with a slack-variable augmented Lagrangian. In Lee D, Sugiyama M, Luxburg, Guyon I, Garnett R. editors, Advances in Neural Information Processing Systems, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/file/31839b036f63806cba3f47b93af8ccb5-Paper.pdf.
23. Powell MJD. A direct search optimization method that models the objective and constraint functions by linear interpolation, pp. 51–67. Netherlands, Dordrecht: Springer; 1994. ISBN 978-94-015-8330-5. https://doi.org/10.1007/978-94-015-8330-5_4.
24. Rockafellar TR. Lagrange multipliers and optimality. SIAM Rev. 1993;35(2):183–238. URL http://www.jstor.org/stable/2133143.
25. Roustant O, Padonou E, Deville Y, Clément A, Perrin G, Giorla J, Wynn Henry. Group kernels for gaussian process metamodels with categorical inputs. SIAM/ASA J Uncertainty Quantif. 2020;8(2):775–806. https://doi.org/10.1137/18M1209386.
26. Thi HAL, Le HM, Dinh TP. Optimization of complex systems: theory, models, algorithms and applications. In: Advances in intelligent systems and computing. Springer International Publishing, 2019. ISBN 9783030218034. URL https://books.google.fr/books?id=R46dDwAAQBAJ.
27. Vazquez E, Bect J. Convergence properties of the expected improvement algorithm with fixed mean and covariance functions. J Statis Plan Inference, 2010;140(11):3088–3095. ISSN 03783758. https://doi.org/10.1016/j.jspi.2010.04.018. URL https://linkinghub.elsevier.com/retrieve/pii/S0378375810001850.
28. Wang Z, Hutter F, Zoghi M, Matheson D, de Feitas Nando. Bayesian optimization in a billion dimensions via random embeddings. J Artif Intell Res. 2016;55:361–87.
29. Wilson JT, Hutter F, Deisenroth MP. Maximizing acquisition functions for bayesian optimization. In Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS'18, page 9906-9917, Red Hook, NY, USA, 2018. Curran Associates Inc.
30. Zaefferer M. CEGO. https://cran.r-project.org/package=CEGO, 2014–2021.
31. Zhang Y, Tao S, Chen W, Apley DW. A latent variable approach to Gaussian Process modeling with qualitative and quantitative factors. Technometrics, 2019;1–12. ISSN 0040-1706, 1537-2723. https://doi.org/10.1080/00401706.2019.1638834.
32. Zhang Y, Apley DW, Chen W. Bayesian optimization for materials design with mixed quantitative and qualitative variables. Sci Rep. 2020;10(1). ISSN 2045-2322. https://doi.org/10.1038/s41598-020-60652-9. URL http://www.nature.com/articles/s41598-020-60652-9.

Cuesta-Ramirez *et al. Adv. Model. and Simul. in Eng. Sci.*(2022)9:6   Page 29 of 29

33. Zuniga MM, Sinoquet Delphine. Global optimization for mixed categorical-continuous variables based on gaussian process models with a randomized categorical space exploration step. INFOR Inform Syst Opera Res. 2020;58(2):310–41. https://doi.org/10.1080/03155986.2020.1730677.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.