Operations Research Forum (2023) 4:12
https://doi.org/10.1007/s43069-022-00180-6

ORIGINAL RESEARCH

A General Mathematical Framework for Constrained Mixed-variable Blackbox Optimization Problems with Meta and Categorical Variables

Charles Audet[1] · Edward Hallé-Hannan[1] · Sébastien Le Digabel[1]

Received: 4 April 2022 / Accepted: 22 November 2022 / Published online: 23 February 2023

Abstract
A mathematical framework for modelling constrained mixed-variable optimization problems is presented in a blackbox optimization context. The framework introduces a new notation and allows solution strategies. The notation framework allows meta and categorical variables to be explicitly and efficiently modelled, which facilitates the solution of such problems. The new term meta variables is used to describe variables that influence which variables are included or excluded: meta variables may affect the number of variables and constraints. The flexibility of the solution strategies supports the main blackbox mixed-variable optimization approaches: direct search methods and surrogate-based methods (Bayesian optimization). The notation system and solution strategies are illustrated through an example of a hyperparameter optimization problem from the machine learning community.

Keywords Blackbox optimization · Derivative-free optimization · Mixed-variable optimization · Categorical variables · Meta variables

1 Introduction

This work considers a general constrained optimization problem

✉ Sébastien Le Digabel
sebastien.le.digabel@gerad.ca
https://www.gerad.ca/Sebastien.Le.Digabel

Charles Audet
https://www.gerad.ca/Charles.Audet

Edward Hallé-Hannan
edward.halle-hannan@polymtl.ca

[1] GERAD and Department of Mathematics and Industrial Engineering, Polytechnique Montréal, Montreal, Canada

$$\min_{x \in \Omega \subseteq \mathcal{X}} f(x), \tag{1}$$

where $f : \mathcal{X} \to \overline{\mathbb{R}}$ (with $\overline{\mathbb{R}} = \mathbb{R} \cup \{+\infty\}$) is the objective function, $\mathcal{X}$ is the domain of the objective function f, $x \in \mathcal{X}$ is a point and $\Omega \subseteq \mathcal{X}$ is the feasible set defined by the constraints of the problem.

1.1 Blackbox Optimization

In blackbox optimization (BBO), the objective and constraint functions are assumed to be blackboxes. In [1], a blackbox function is defined as: "*any process that when provided an input, returns an output, but the inner working of the process are not analytically available*". For example, an aircraft drag evaluation could require using a finite element method that takes some design variables, such as a the material and dimension of a wing, and outputs a drag value. In this example, the inner working of the process are too complicated to allow a closed-form analytical formulation: the finite element method is a blackbox function.

In BBO, the objective and constraint functions can only provide information through evaluations. For instance, the only information that the objective function f (a blackbox) can provide is the mapping of a point $x \in \mathcal{X}$ to its image $f(x) \in \overline{\mathbb{R}}$ through a given process, which is typically the execution of a time-consuming computer program. In [2], the authors provide examples : "*automotive valve train design* [3] *requires seconds, sample size identification for bioequivalence studies in the pharmaceutical studies industry* [4] *requires minutes, hyperparameter optimization* [5] *requires hours, and airfoil trailing-edge noise reduction* [6] *requires days of CPU time for each execution of the associated computer code or simulation*".

In a minimization optimization context, allowing the objective function f to take the value $+\infty$ is a convenient way to flag out and eliminate trial points outside of the domain $\mathcal{X}$ as well as to reject points for which the blackbox unexpectedly failed to return a value.

Consequently, derivatives are often inaccessible or are too computationally expensive to compute. Thus, in general, traditional derivative-based optimization methods cannot be applied to blackbox functions [1]. This work uses the following terminology [1]: "*BBO is the study of design and analysis of algorithms that assume the objective and/or constraint functions are given by blackboxes*".

1.2 Class of Problems

This work proposes a general framework to model a wide class of optimization problems with a broad type of variables. In addition to continuous and integer variables, the class may include categorical (or qualitative) variables as well as a special type of variables whose values determine if some other variables are included or excluded in the optimization problem. In this work, these special type of variables are called meta variables. These variables may alter the number of variables and constraints of the problem. Meta variables are a cornerstone of this work and are thoroughly defined in Section 3.

Real applications of this class of mixed-variable problems have been studied in the literature [2]. In [7, 8], a thermal insulation problem involves a meta variable that determines the number of heat intercept, where each added heat intercept involves new design variables. In [9], the optimal design of a magnetic resonance imaging device contains a meta variable that controls the number of magnets in the apparatus, which influences the number of variables and constraints the optimization problem. In deep learning [10], determining hyperparameters that maximize the performance of the model includes a meta variable representing the number of hidden layers that affects the number of variables associated to the units (or neurons) of the architecture of the model.

The presence of meta and categorical variables makes this class of mixed-variable optimization problems notoriously difficult to tackle. Meta variables are fundamentally difficult to model and treat, as they may alter the number of variables and constraints. Categorical variables are also difficult to treat since they take discrete values from sets that do not contain any intrinsic metric of distance between the elements and cannot be easily relaxed. For example, in the aircraft design problem, a possible categorical variable could be the material composition of a specific part. Moreover, the class of problems includes problems that may also contain continuous or integer variables. The different variable types are detailed in Section 3.1.

1.3 Literature Review

A first framework to treat mixed-variable optimization problems in a context of blackbox optimization is detailed in [11]. The methodology is based on the general pattern search algorithm (GPS) and the variables are partitioned into two components: discrete and continuous. The discrete component contains both the quantitative and the qualitative discrete variables, *i.e.*, integer variables in $\mathbb{Z}$ as well as categorical variables. The continuous component contains the continuous variables. Two main ideas emerged from this article. First, the continuous space, in which classical continuous blackbox optimization methods can be applied, are generated after fixing the discrete component. Thus, for a fixed discrete component, a continuous space is generated and explored.

Second, the exploration of the discrete variables space is being done by defining a set of neighbors function $\mathcal{N}$, which is an additional structure to the domain $\mathcal{X}$, such that $\mathcal{N}(x)$ is a set of neighbors of $x \in \mathcal{X}$. With this additional structure a local minimizer $x_\star$ is defined so that $x_\star$ minimizes the objective function f with respect to the set of neighbors (discrete part) and the continuous space. From the contributions of [11], a practical application of a thermal insulation optimization problem is treated and optimized [8]. In [12], the filter method is added to the methodology proposed in [8, 11]. This addition enables the methodology to treat general nonlinear constraints. In [13], the methodology based on GPS in [8, 11, 12] is extended to the mesh adaptive direct search (MADS) algorithm [14]. A rigorous convergence analysis based on [12] was improved by using the Clarke generalized derivatives on the continuous space. In [15], the MADS algorithm is equipped with a granular mesh called GMesh, which allows the discretization of granular and continuous variables simultaneously. Granular variables are quantitative variables with a controlled number of decimals. In particular, GMesh enables to treat integer-continuous problems with the MADS algorithm since integer variables are a special type of granular variables without decimals.

An important contribution from [9, 16] is the introduction of dimensional variables. These variables affect the number of variables, the number of constraints and the structure of the optimization problem. A point x is partitioned into three components: a dimensional component, a discrete component and a continuous component. The set of discrete variables, where the discrete component belongs, is generated from a fixed dimensional component. Additionally, the continuous space is generated from both a fixed dimensional component and a fixed discrete component. From the partition of a point x, a domain and a feasible set are implicitly presented in the formulation of an optimization problem. The present work importantly relies on the contributions from [9, 16].

A categorical kernel function is defined in [17] with the aim of tackling mixed-variable optimization problems with a surrogate approach based on radial basis functions (RBF). The categorical kernel function measures the number of disagreement between two categorical components, where a disagreement is counted when a specific variable of the two compared components is not the same. The surrogate is built upon a composed kernel such that the RBF, centered at some interpolation points, are shifted by the number of categorical disagreements between the fitted point and the interpolation points. The criterion to determine which point is evaluated by the objective function is based on [18]. In essence, the criterion has a high value for points that are distant from the previous evaluations (exploration) or points that have promising surrogate-value (intensification).

Bayesian optimization (BO) has undergone significant development with the recent advent of machine learning. Nowadays, the emerging scientific literature is mainly related to BO based on Gaussian processes (GP), which serves as probabilistic distribution surrogates [19]. The success of BO is explained by an acquisition function that selects which candidate point is to be evaluated. The acquisition function defines a less costly optimization problem with the surrogate. For continuous problems, a well documented acquisition function is the expected improvement (EI) [20], which provides candidate points in unexplored regions (exploration) and candidate points in promising regions (intensification): algorithms that applies an EI function on a GP are often referred to as efficient global optimization (EGO) algorithms [20]. Historically, BO based on GPs was used to tackle continuous blackbox optimization problems. Hence, in practice the integer and categorical variables (one-hot encoded) are often relaxed as continuous variables and rounded afterwards [21]. This naive approach, used in some modern blackbox solvers, often leads to failure such as a mismatch between the points provided by an acquisition function and where the true evaluation takes place, as well as reevaluating some points [21]. Moreover, an important number of additional variables may be generated by the one-hot encoding of categorical variables. In [22], continuous-categorical optimization problems are modelled with GPs, where a GP surrogate is characterized by a kernel composed of tensor products and additions of one-dimensional kernels: an one-dimensional kernel per variable. The one-dimensional kernel of a given categorical variable is a $C \times C$ matrix in which an element of the matrix is a correlation measure between two categories of the categorical variable. The matrix-kernels for the ordinal and nominal categorical variables are distinguished. In [23], the BO framework is extended to tackle mixed-variable optimization problems with continuous, discrete (categorical and integer) and dimensional variables, such as defined in [9, 16]. Again, the GP surrogate is characterized by a composed kernel built upon products and additions of one-dimensional kernels, each specified by the type of its corresponding variable. Moreover, two approaches are proposed in [23]: multiple surrogates, one surrogate per dimensional component (set of dimensional variables), which separates the main problems into subproblems and a single surrogate with a composed kernel built upon on all variables, including dimensional variables.

In [24], the authors combined the user-defined set of neighbors in order to tackle categorical variables in an EGO subproblem. More precisely, a user-defined set of neighbors is randomly defined with a discrete probability distribution based on a GP. Thus, the randomly user-defined set of neighbors serves as a randomized categorical exploration strategy for the EGO subproblem.

Covariance functions (kernels) are fundamentally difficult to defined on categorical sets since the distance between two categories (levels) is not defined. To tackle this difficulty, the authors in [25] proposed to map the categories of each categorical variable to a set of quantitative values that represents some underlying latent unobservable quantitative variable. More precisely, the categories of each categorical variable are mapped to a 2D continuous space: for a given categorical variable, the categories are compared into a 2D space. The quantitative values in the vectors does not have any intrinsic meaning. However, the distance between the values encapsulates some information, since the categories are mapped among themselves in a correlated manner. Mathematically, the mapping is done via a maximum likelihood estimation (MLE) procedure that fits the best multivariate Gaussian distribution of some data. A GP model is then constructed on continuous variables and latent variables. Furthermore, the authors in [26] formalized a pre-image problem with a constraint that recovers a categorical component from a vector of continuous latent variables. More technically, a continuous EGO problem is formulated as an augmented Lagrangian with a retrieving constraint on the continuous latent variables.

1.4 Motivation, Contributions and Structure of the Work

The compact and general formulation of Problem (1) does not explicitly model mixed-variable problems. Hence, in order to efficiently model and tackle these problems, the formulation must be further detailed with a focus on treating the categorical and, more particularly, meta variables.

To the best of the authors' knowledge, no similar work has rigorously and explicitly formulated Problem (1) for mixed-variable problems. The core aspect and main contributions of this work are to formally define the domain $\mathcal{X}$ of the objective function, the feasible set Ω and a point x for mixed-variable problems. These definitions have many implications in the mathematical framework that consists of a notation system and solution strategies. The notation framework rigorously models constrained mixed-variable problems in an efficient and unambiguous manner, as well as shines the light on some algorithmic subtleties in the solution of these problems. The present work also formalizes solution strategies present in the literature and tackles these problems by being fully compatible with two of the main mixed-variable blackbox optimization approaches: direct search methods are formalized as subproblems strategies and surrogate-based Bayesian optimization approaches are formalized as auxiliary problem strategies. Hence, another main contribution of this work is the formalization of direct search methods and surrogate Bayesian optimization approaches, for mixed-variable problems with meta and categorical variables, into a single and general framework.

The present work does not present any computational experiments, as it focuses on the presentation of the framework. The essence of the present work is similar to the well-known surrogate management framework [27], which was proposed with very few experiments. Computational experiments will be carried out in future work.

The document is organized as follows. First, an example of a deep learning mixed-variable optimization problem is described in Section 2. The example is used throughout the paper to facilitate understanding. Second, the notation system is exhaustively detailed in Section 3. The notation partitions variables in different types, classifies constraint functions, and formally presents their domain and the feasible set. Finally, solution strategies are presented in Section 4 from the framework perspective.

2 Hyperparameter Multilayer Perceptron Example

In order to illustrate the mathematical framework, a simplified constrained hyperparameter optimization (HPO) problem on a multilayer perceptron (MLP) is detailed throughout the document. The goal of the detailed problem is to model a simple constrained mixed-variable blackbox optimization problem.

2.1 Basics of the MLP

No prerequisites in machine learning or deep learning are necessary since the example is treated as a blackbox problem. However, the example is more relevant with some background in machine learning or deep learning. Hence, an overview of the basic concepts of supervised learning are briefly explained and illustrated on the MLP example in this section. For more details, see Chapters 5 and 6 in [28] that provide an excellent and concise introduction to this topic.

In the example, the MLP is a model designed to perform regression for inputs with $p \in \mathbb{N}$ continuous features. More precisely, the MLP is a regression model $\hat{h} : \mathbb{R}^p \to \mathbb{R}$ that approximates a nonlinear function $h : \mathbb{R}^p \to \mathbb{R}$, such that

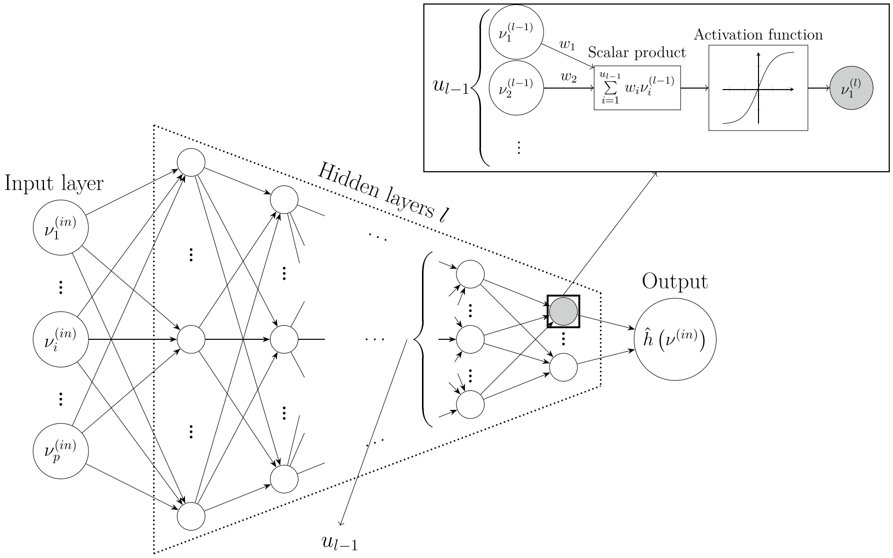


Fig. 1 An example of a MLP

$\nu^{(in)} = (\nu_1, \nu_2, \dots, \nu_p) \in \mathbb{R}^p$ is an input data, $\hat{h}\left(\nu^{(in)}\right) \in \mathbb{R}$ is a predicted output and $h\left(\nu^{(in)}\right)$ is the corresponding true output. The example is established in a supervised learning context, since the true output is assumed to be known.

In order to respect the dimensions of the domain and the codomain of the function h, the architecture of the model must have $u_{in} = p$ units in the input layer and $u_{out} = 1$ unit in the output layer, as illustrated in Fig. 1.

The regression model $\hat{h}$ in Fig. 1 is an example of a MLP neural network with l hidden layers. The mapping, commonly called feed-forward, of the model $\hat{h} : \mathbb{R}^p \to \mathbb{R}$ is illustrated in Fig. 1: starting from the input layer, the input $\nu^{(in)}$ leads to the output layer $\hat{h}\left(\nu^{(in)}\right)$ through the hidden layers from left to right. The transition from one layer to a subsequent one is shown for the first node in the l-th layer (see the rectangle at the top of Fig. 1): the value $\nu_1^{(l)}$ (grey node) is determined by a scalar product between the units of the $l-1$-th layer and the weights (edges) w_i, followed by a mapping of the resulting scalar product with a nonlinear function called the activation function. The activation function enables to approximate nonlinear functions.

The model fitting is commonly referred to as the training. It consists of adjusting the weights of the model to minimize a loss function from the training dataset. A loss function quantifies a score difference between a prediction $\hat{h}\left(\nu^{(in)}\right)$ and its corresponding true value $h\left(\nu^{(in)}\right)$. After the training process is completed, the performance of the model can be tested on another dataset, commonly called the test dataset.

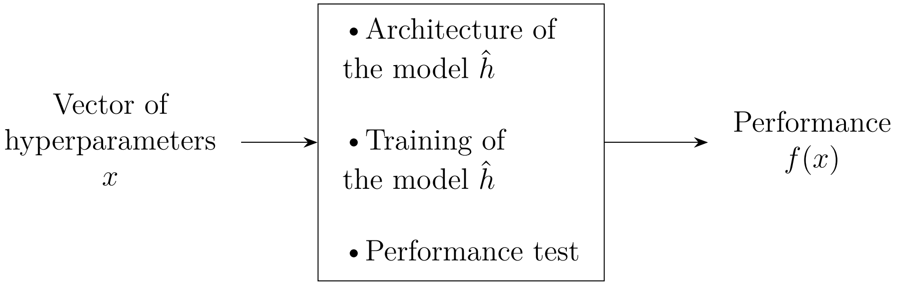

Vector of hyperparameters x
• Architecture of the model $\hat{h}$
• Training of the model $\hat{h}$
• Performance test
Performance $f(x)$

Fig. 2 Objective function of the HPO problem

2.2 Constrained HPO of the MLP

As any deep neural network, a MLP model is characterized by its hyperparameters. In deep learning, a hyperparameter is a parameter whose value affects the architecture (units or layers) of the model or controls the training process (activation function, optimizer or learning rate). The hyperparameters must not be confused with the parameters (weights represented as edges in Fig. 1) of the model. Hyperparameters are not internally adjusted with the training dataset, but they rather control minimization process of the loss function or the number of parameters of the model. The performance of a MLP is highly sensible to the choice of hyperparameters. Finding good hyperparameters is a difficult task since the training, validation and performance test is done for a fixed set of hyperparameters (vector of hyperparameters) as illustrated in Fig. 2.

In Fig. 2, the objective function f of the HPO problem is schematized. Computation of the objective function f requires the training and testing of a deep neural network model for a vector of hyperparameters x as an input. The goal is to find the set of hyperparameters x that maximizes a performance score $f(x)$, which is usually a precision score of accuracy on a untested data set. Although the internal mechanisms of the MLP are known (see Section 2.1), the HPO problem is assumed to be a blackbox optimization problem. The internal adjustment of the weights (training), for a given vector of hyperparameters x, is performed by an algorithm called backpropagation, which computes the gradient of the loss function with respect to the weights of the model. In practice, the number of parameters is in the order of millions or billions. Thus, the number of parameters combined with the backpropagation algorithm makes it almost impossible to formulate an analytical expression of f. For a more in-depth description of the HPO see [10].

The hyperparameters of the MLP example are described in Table 1. Some important hyperparameters are internationally left out, such as the mini-batch size or the dropout, in order to restrict the number of variables to be modeled and keep the presentation concise.

In Table 1, the index i in u_i represents the i-th hidden layer. The number of units in the hidden layers are grouped in the vector $u(l) = (u_1, u_2, \dots, u_l)$, where l is the number of hidden layers. The situation where there are no hidden layer

Table 1 Hyperparameters of the MLP

Hyperparameter	Variable	Domain
Learning rate	r	]0, 1[
Activation function	a	{ReLU, Sigmoid}
# of hidden layers	l	$\{0, 1, \dots, l^{\max}\}$
# of units hidden layer i	u_i	$\{u_i^{\min}, u_i^{\min}+1, \dots, u_i^{\max}\}$
Optimizer	o	{Adam, ASGD}
if $o =$ ASGD		
decay	λ	]0, 1[
power update	α	]0, 1[
averaging start	t_0	]1E3, 1E8[
if $o =$ Adam		
running average 1	β_1	]0, 1[
running average 2	β_2	]0, 1[
numerical stability	ϵ	]0, 1[

is modeled by setting $l = 0$. In that case, the variables u_i are said to be excluded, which signifies that the variables u_i are not part of the optimization problem. The terminology of included and excluded is further detailed in Sections 3.1.1 and 3.1.2.

The optimizer o is an important hyperparameter since it is a gradient descent method, based on the backpropagation algorithm, that iteratively adjusts the weights of the model. Moreover, depending on the choice of the optimizer, different hyperparameters are involved. Indeed, in Table 1 the optimizers do not share the same continuous hyperparameters. A given optimizer leads to different variables in the problem. For example, the variable decay λ is only included in the problem when $o =$ ASGD. This consideration is important and will be discussed throughout the document, but notably in Section 3.1.1 that focuses on meta variables. Related to the optimizer is the learning rate r which is the stepsize in the gradient descent method.

Finally, constraints on the hyperparameters are imposed to illustrate the notation (see Section 3.4). The first constraint of the problem imposes that $r \geq \alpha l$, where r is the learning rate, l is the number of hidden layers and $\alpha \in \mathbb{R}^+$ is a tunable-scalar (not a hyperparameter). The constraint $r \geq \alpha l$ models a trade-off between training time and performance of the model. In general, higher values of l are directly associated to longer training times, since the number of parameters in the model grows with the number of hidden layers l. Hence, the constraint $r \geq \alpha l$ forces that larger values of l are associated with larger learning rates r, which allows the model to learn faster but at the cost of some performance gains by fine-tuning the model parameters. The other constraints are $u_i \leq u_{i-1}$ $\forall i \in \{2, 3, \dots, l\}$ and they impose that the number of units is monotone decreasing in order to reduce the training time of the model. The dotted trapezoidal architecture of the MLP in Fig. 1 is based on the constraints on the units u_i.

3 Notation Framework

This section contains the fundamental mathematical definitions that allow modelling mixed-variable problems. In Section 3.1, the mathematical objects that define the variables (point and components) are described. Subsequently, the domain $\mathcal{X}$ is detailed in Section 3.2. Then, the feasible set $\Omega \subseteq \mathcal{X}$ is precised in Section 3.3. Finally, the content in Sections 3.1 to 3.3 is discussed within the MLP example in Section 3.4.

3.1 Variables and Components of a Point

The goal of an optimization algorithm is to find a feasible point $x_\star$ that minimizes the objective function f. In a mixed-variable optimization context, it is necessary to formally define how a point is partitioned into different components.

Definition 1 (Components of a point). A point $x = (x^{\mathrm{m}}, x^{\mathrm{cat}}, x^{\mathrm{qnt}})$ is partitioned into three components:

- a meta component x^{m};
- a categorical component $x^{\mathrm{cat}} = (x^{\mathrm{nom}}, x^{\mathrm{ord}})$, which itself is partitioned into the unordered categorical (nominal) component x^{nom} and the ordered categorical (ordinal) component x^{ord};
- a quantitative component $x^{\mathrm{qnt}} = (x^{\mathrm{int}}, x^{\mathrm{con}})$, which itself is partitioned into the integer component x^{int} and the continuous component x^{con}.

For each $t \in \{\mathrm{m}, \mathrm{cat}, \mathrm{nom}, \mathrm{ord}, \mathrm{qnt}, \mathrm{int}, \mathrm{con}\}$, the component x^t is a vector containing $n^t \in \mathbb{N}$ variables of type t:

$$x^t = (x^t_1, x^t_2, \dots, x^t_{n^t}). \tag{2}$$

The integer and continuous components are contained in the quantitative component x^{qnt} for several reasons. In practice these variables are generally optimized with well known methods. Moreover, some blackbox optimization algorithms have the ability to simultaneously optimize integer and continuous variables. Thus, it is convenient to group these variables to lighten the notation. However, the quantitative component $x^{\mathrm{qnt}} = (x^{\mathrm{int}}, x^{\mathrm{con}})$ can easily be partitioned into its two components if necessary.

The meta, quantitative and categorical components, as well as their corresponding variables, are respectively discussed in Sections 3.1.1, 3.1.3 and 3.1.4. Additionally, the motivations behind the compact partition $x = (x^{\mathrm{m}}, x^{\mathrm{cat}}, x^{\mathrm{qnt}})$ and the complete partition $x = (x^{\mathrm{m}}, x^{\mathrm{nom}}, x^{\mathrm{ord}}, x^{\mathrm{int}}, x^{\mathrm{con}})$ are discussed and illustrated in Section 3.1.5. Finally, in Section 3.1.2, the roles of variables and constraints are introduced in order to define more clearly the domain $\mathcal{X}$ in Section 3.2.

3.1.1 Decree Property, Meta Variables and Meta Component

Meta variables are a cornerstone of this work. To formally define them, the decree property is introduced.

Definition 2 (Decree property and meta variables). The decree property is attributed to variables whose values determine if other variable(s) and/or constraint(s) are included or excluded from the optimization problem.

Variables possessing the decree property are called *meta variables.*

In the MLP example from Section 2, the variable decay λ is included when $o = \text{ASGD}$, otherwise it is excluded. An included constraint is a constraint function that defines the feasible set Ω that contains feasible solutions, whereas an excluded constraint has no impact on the feasible set Ω. In the MLP example, if the number of hidden layer is $l = 0$, then the constraints $u_i \leq u_{i-1}\ \forall i \in \{2, 3, \dots, l\}$ are excluded.

In addition to being meta, a meta variable also has a common variable type, such as categorical (cat), integer (int) or continuous (con). For example, in the MLP example, the number of hidden layers l is a meta variable since it decrees the units u_i in the hidden layers $i \in \{1, 2, \dots, l^{\max}\}$. More precisely, for a given $l \in \{0, 1, \dots, l^{\max}\}$, $u_1, u_2, \dots, u_l$ are included variables and $u_{l+1}, u_{l+2}, \dots, u_{l^{\max}}$ are excluded. Moreover, l is also an integer variable, since its domain is $\{0, 1, \dots, l^{\max}\}$. The number of hidden layers l is a meta-integer variable.

On the one hand, meta variables may affect the number of variables (dimension) as well as the number of constraints included in the problem. In the MLP example, the number of hidden layers l affect the dimension and the number of constraints of the problem. Indeed, l affect the number of variables (dimension) since it determines the number of variables associated to the units u_i in the hidden layers that are included: precisely, this number of variables is $\left|\{u_1, u_2, \dots, u_l\}\right|$. Moreover, l also decrees the corresponding constraints $u_i \leq u_{i-1}\ \forall i \in \{2, 3, \dots, l\}$, thus affecting the number of constraints.

On the other hand, meta variables do not necessarily affect the dimension of the problem or the number of constraints. Indeed, in the MLP example, both optimizers ASGD and Adam from Table 1 decree three different continuous hyperparameters, specific to each optimizer. The dimension nor the number of constraints is affected by the choice of the optimizer.

In that regard, meta variables are a generalization of the strictly discrete dimensional variables defined in [9, 16]. There are several reasons to generalize the dimensional variables into the new meta variables.

1. Meta variables do not necessarily affect the dimension (*e.g.*, the optimizer o), in comparison to dimensional variables. This is an important justification, especially since the optimization of hyperparameters (see Section 2) in deep learning is, in effect, one of the most important industrial and academic mixed-variable blackbox optimization problem.

2. Meta variables can be of any type $t \in \{\text{cat}, \text{nom}, \text{ord}, \text{qnt}, \text{int}, \text{con}\}$ and are not strictly discrete (categorical or integer). For example, a problem could contain a continuous variable frequency that takes its value in the visible spectrum (continuous domain). The visible spectrum could be partitioned into the three intervals that represent the red-blue-green colors. The frequency could decree some variable(s) or constraint(s) depending in which interval (color) it belongs to. In that particular example, the frequency is a meta-continuous variable.
3. The terminology *dimensional* is used in physical sciences and engineering to describe quantities such as the velocity, mass and time. Many of blackbox mixed-variable optimization problems come from these disciplines, hence it is also strategic to avoid the term *dimensional* for a variable type.

Finally, the meta variables are contained in the meta component

$$x^{\text{m}} = \left(x_1^{\text{m}}, x_2^{\text{m}}, \dots, x_{n^{\text{m}}}^{\text{m}}\right), \tag{3}$$

where $n^{\text{m}} \in \mathbb{N}$ is the number of meta variables and x_j^{m} is a meta variable. For short, the decree propriety is attributed to the meta component x^{m}, since it contains the meta variables (more details are given in Section 3.1.2): this is a natural extension of Definition 2.

3.1.2 Roles of Variables

Some variable(s) are included or excluded depending on the value of a meta variable. This consideration leads to the following definition.

Definition 3 (Decreed variable). A variable of type $t \in \{\text{cat}, \text{nom}, \text{ord}, \text{qnt}, \text{int}, \text{con}\}$ is a decreed variable if its inclusion or exclusion is determined by values of a meta variable.

Definition 3 above ensures that a decreed variable cannot be a meta variable, since the type t belongs to $\{\text{cat}, \text{qnt}, \text{nom}, \text{ord}, \text{int}, \text{con}\}$. Therefore, a meta variable cannot decree another meta variable: this modeling choice has been made for the following reason. Problems in which a meta variable can be decreed are more general than the class of problems described in Section 1.2, however they are uncommon in practice, they make the notation considerably more cumbersome and they are not the target problems of this work.

Furthermore, there may be some variables that are not decreed and do not decree other variables. For example, in the MLP example, the learning rate r, is always included, is not a meta variable and is not decreed (see Table 1). This remark leads the following definition.

Definition 4 (Neutral variable). A variable of type $t \in \{\text{cat}, \text{nom}, \text{ord}, \text{qnt}, \text{int}, \text{con}\}$ is a neutral variable if it is always included in the problem.

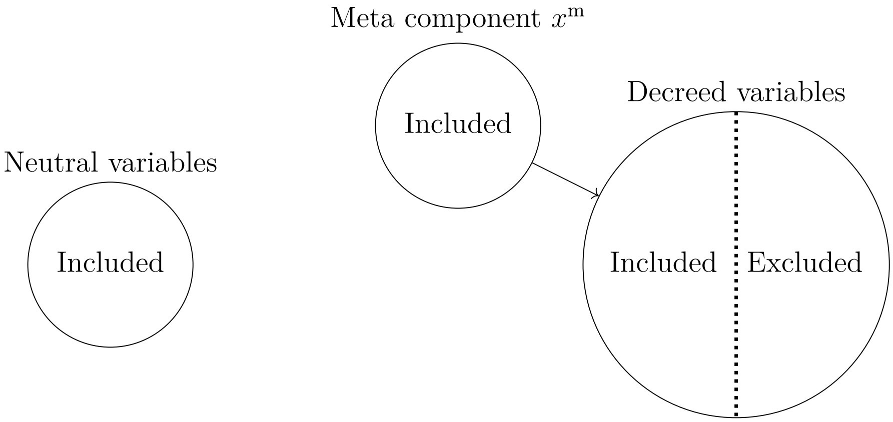


Fig. 3 Roles of variables

Definition 4 ensures that a neutral variable cannot be a meta variable. Hence, neutral variables are variables that are always included but not meta. Incidentally, note that meta variables are also always included.

The definitions for meta, decreed and neutral variables are all related to the decree property: meta variables have it; decreed variables are, by implication, affected by it and neutral variables are unaffected by it. To summarize and conceptualize this, the *role of a variable* is defined.

Definition 5 (Role of a variable). The role of a variable represents its relation to the decree property. A variable is attributed a single role amongst meta (as in Definition 2), decreed (as in Definition 3) or neutral (as in Definition 4).

The role of a variable differs from its variable type. In addition to its type $t \in \{\mathrm{m}, \mathrm{cat}, \mathrm{nom}, \mathrm{ord}, \mathrm{qnt}, \mathrm{int}, \mathrm{con}\}$, each variable takes a single role amongst meta, decreed or neutral. By definition, meta variables are attributed the role meta. Interrelations between meta, decreed and neutral variables are schematized in Fig. 3.

Figure 3 illustrates many points. First, excluded variables are necessarily decreed. Second, neutral variables are disjoint from the other variables, which implies that they are unaffected by the meta variables. Third, meta variables are always included. Additionally, the arrow in Fig. 3 symbolizes the decree property. Finally, the meta component x^{m} decrees all decreed variables.

In essence, the roles of variables consist of additional terminologies that help elucidate some subtleties of the mathematical framework. The roles of the variables are particularly useful for defining the domain $\mathcal{X}$ in Section 3.2.

3.1.3 Categorical Component

The categorical component x^{cat} contains qualitative variables, known as categorical variables, that are not meta: a categorical variable may be decreed or neutral. Categorical variables are discrete variables that take qualitative values called categories. More precisely, a categorical variable x_j^{cat} has j categories such that $x_j^{\text{cat}} \in L_j = \{l_1, l_2, \dots, l_j\}$, where l_i is the i-th category for $0 \leq i \leq j$.

A categorical variable belong to either an unordered or ordered set. A nominal (unordered categorical) variable belongs to an unordered set. For example, the blood-type of a given person $x_j^{\text{nom}} \in \{\text{O-}, \text{O+}, \dots, \text{AB+}\}$ is a nominal variable since there is no intrinsic ordering between the blood type categories. The nominal variables are contained in the unordered categorical component x^{nom}.

Subsequently, an ordinal (ordered categorical) variable belongs to an ordered set, in which the elements (categories) of the set are naturally ordered[1]. For example, the education level of a person $x_j^{\text{ord}} \in \{\text{“less than HS”}, \text{“HS”}, \dots, \text{“MSc”}, \text{“PhD”}\}$, where HS signifies high school, is an ordinal variable, since the categories are naturally ordered, *i.e.*, “less than HS” $\leq$ “HS” $\leq \dots \leq$ “MSc” $\leq$ “PhD”. The ordinal variables are contained in the categorical ordered component x^{ord}. Although the ordinal variables belong to ordered sets, distances between the ordinal variables are inherently unknown: “[...] *there is an ordering between the values, but no metric notion is appropriate*” [29].

Note that a binary variable may be: a nominal variable, e.g., $x_j^{\text{nom}} \in \{\text{True}, \text{False}\}$; an ordinal variable, e.g. $x_j^{\text{ord}} \in \{\text{small}, \text{tall}\}$ or an integer variable, e.g. $x_j^{\text{int}} \in \{0, 1\}$. A modeling choice is made in this regard, however a binary variable should be typed into to either nominal (nom), ordinal (ord) or integer (int) based on its nature.

The categorical component $x^{\text{cat}} = (x^{\text{nom}}, x^{\text{ord}})$ is composed of the nominal component x^{nom} and the ordinal component x^{ord}, which are not meta.

In some cases, it might be beneficial to exploit the order of an ordinal variable, motivating the partition of the categorical component into nominal and ordinal components. For instance, [22] used different kernels for ordinal and nominal components. Moreover, a direct search exploration strategy could be generically implemented with a previous and next element mechanism for an ordinal set.

In previous work [8, 11, 12], meta variables were included in the categorical variables; it is an important distinction from this work.

3.1.4 Quantitative Component

The quantitative x^{qnt} component contains discrete and continuous quantitative variables that are not meta variables: a quantitative variable may be decreed or neutral.

[1] Formally, a (partially) ordered set is a set X equipped with a binary relation (partial order) $\leq$ that satisfies the properties of reflexivity, transitivity and antisymmetry for any pair $x, y \in X$. Additionally, an order $\leq$ is total if the comparability property ($x, y \in S$ are either $x \leq y$ or $y \leq x$) is met, which means that any two elements are comparable (a totally ordered set).

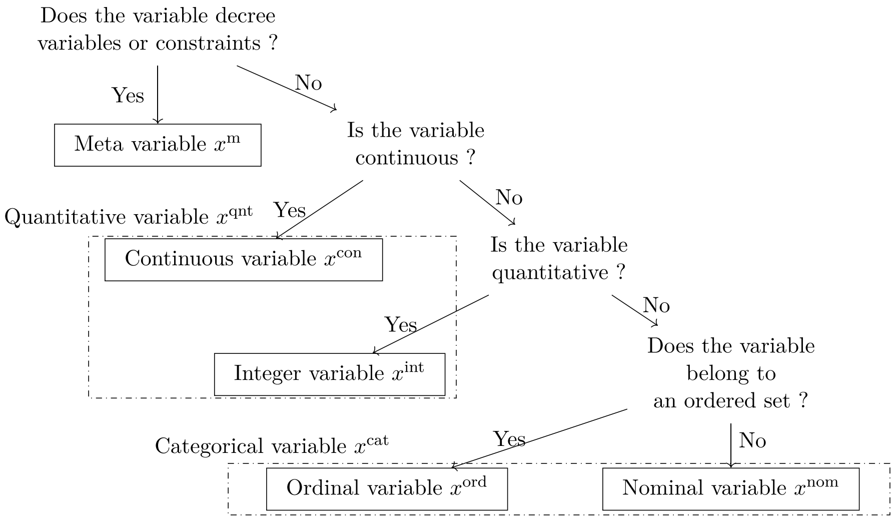

Does the variable decree variables or constraints ?
Yes
No
Meta variable x^{m}
Is the variable continuous ?
Quantitative variable x^{qnt}
Yes
No
Continuous variable x^{con}
Is the variable quantitative ?
Yes
No
Integer variable x^{int}
Does the variable belong to an ordered set ?
Categorical variable x^{cat}
Yes
No
Ordinal variable x^{ord}
Nominal variable x^{nom}

Fig. 4 Variable type classification tree chart

Formally, the quantitative component x^{qnt} contains variables that belong to intrinsically ordered sets for which a metric of distance is intuitively definable. Simply put, the quantitative component x^{qnt} contains the integer and the continuous variables.

The integer component x^{int} exclusively contains discrete quantitative variables, called integer variables, that are not meta. Unlike the categorical variables, integer variables are always ordered and belong to sets with appropriate metric notions [29]. The decision to separate the discrete variables into the categorical component and the integer component differs from some of the current literature. Indeed, in [9, 13, 23] the discrete component contains both the categorical and the integer variables. Thus, in [9, 13, 23], categorical and integer variables are not clearly distinguished: some useful mathematical properties of the integer variables might not be exploited at their fullest. In that regard, integer programming is a well developed optimization field that exploits the properties of the integer variables. In practice, this strengthens the separation of the integer variables from the categorical ones, since integer programming techniques could be implemented in the algorithmic framework to treat the integers variables.

The continuous component x^{con} contains continuous variables that are not meta. Continuous variables have many properties that are generally exploited in a context of blackbox optimization.

3.1.5 Variable Type Classification

Figure 4 shows a tree chart that classifies a variable by their type in the proposed mathematical framework. The first question identifies meta variables, the second determines the continuous variables, the third distinguishes integer from categorical variables and the last one separates ordinal from nominal variables. The first question also imply that continuous, integer, ordinal and nominal variables are not meta variables. The doted box in the middle illustrates that quantitative variables contains the continuous and integer variables, whereas the doted box in the bottom exhibits that two types of categorical variable, which are ordinal and nominal variable.

The full partition of a point $x = (x^{\mathrm{m}}, x^{\mathrm{nom}}, x^{\mathrm{ord}}, x^{\mathrm{int}}, x^{\mathrm{con}})$, displayed in Fig. 4, offers flexibility and extracts most mathematical information accessible to facilitate the optimization process: the modelling choices for the partitions are motivated by these considerations. The compact partition $x = (x^{\mathrm{m}}, x^{\mathrm{cat}}, x^{\mathrm{qnt}})$ implicitly contains the same information and flexibility of the full partition. However, the compact partition alleviates the notation, which is why it is mostly used throughout this work.

3.2 Domain

At this stage, the variables have been: 1) classified into different types; 2) organized into components, which forms a partition of a point x; 3) attributed roles. The next step is to define the domain $\mathcal{X}$ of the objective function $f : \mathcal{X} \to \overline{\mathbb{R}}$.

Definition 6 (Domain). The domain of objective function is defined by:

$$\mathcal{X} = \left\{ (x^{\mathrm{m}}, x^{\mathrm{cat}}, x^{\mathrm{qnt}}) \ : \begin{array}{l} x^{\mathrm{m}} \in \mathcal{X}^{\mathrm{m}}, \\ x^{\mathrm{cat}} \in \mathcal{X}^{\mathrm{cat}}(x^{\mathrm{m}}), \\ x^{\mathrm{qnt}} \in \mathcal{X}^{\mathrm{qnt}}(x^{\mathrm{m}}) \end{array} \right\} \tag{4}$$

where $\mathcal{X}^{\mathrm{m}} \subseteq \mathbb{M}^{n^{\mathrm{m}}}$ is the meta set, $\mathcal{X}^{\mathrm{cat}}(x^{\mathrm{m}}) \subseteq \mathbb{Z}^{n^{\mathrm{cat}}(x^{\mathrm{m}})}$ is the parametrized categorical set and $\mathcal{X}^{\mathrm{qnt}}(x^{\mathrm{m}}) \subseteq \mathbb{Z}^{n^{\mathrm{int}}(x^{\mathrm{m}})} \times \mathbb{R}^{n^{\mathrm{con}}(x^{\mathrm{m}})}$ is the parametrized quantitative set.

The meta set $\mathcal{X}^{\mathrm{m}} \subseteq \mathbb{M}^{n^{\mathrm{m}}}$, the parametrized categorical set $\mathcal{X}^{\mathrm{cat}}(x^{\mathrm{m}}) \subseteq \mathbb{Z}^{n^{\mathrm{cat}}(x^{\mathrm{m}})}$ and the parametrized standard set $\mathcal{X}^{\mathrm{qnt}}(x^{\mathrm{m}}) \subseteq \mathbb{Z}^{n^{\mathrm{int}}(x^{\mathrm{m}})} \times \mathbb{R}^{n^{\mathrm{con}}(x^{\mathrm{m}})}$ are detailed in Sections 3.2.2, 3.2.3 and 3.2.4, respectively.

The dependencies of the parametrized categorical set $\mathcal{X}^{\mathrm{cat}}(x^{\mathrm{m}})$ and parametrized quantitative set $\mathcal{X}^{\mathrm{qnt}}(x^{\mathrm{m}})$ are defined through a parametrization with respect to the meta component x^{m}. These parametrizations are a direct consequence of the decree property of the meta component x^{m}.

Definition 7 (Parametrized set). A parametrized set $\mathcal{X}^{t}(x^{\mathrm{m}})$ of type $t \in \{\mathrm{cat}, \mathrm{nom}, \mathrm{ord}, \mathrm{qnt}, \mathrm{int}, \mathrm{con}\}$ is the set that contains all the components of type t, such that

$$\mathcal{X}^{t}(x^{\mathrm{m}}) = \left\{ x^{t} = \left(x^{t}_{1}, x^{t}_{2}, \dots, x^{t}_{n^{t}(x^{\mathrm{m}})} \right) \ : \ x^{t}_{i} \in S^{t}_{i} \text{ is an included variable } \forall i \in I^{t}(x^{\mathrm{m}}) \right\} \tag{5}$$

where S^{t}_{i} is the domain of the included variable x^{t}_{i} and $I^{t}(x^{\mathrm{m}}) = \{1, 2, \dots, n^{t}(x^{\mathrm{m}})\}$ is the set of indices of the included variables x^{t}_{i}, which are either neutral or decreed by the meta component $x^{\mathrm{m}} \in \mathcal{X}^{\mathrm{m}}$.

From Definition 7, it follows that a component $x^{t} \in \mathcal{X}^{t}(x^{\mathrm{m}})$ contains only the included variables of type t. The excluded variables are not contained in the component $x^t \in \mathcal{X}^t(x^{\mathrm{m}})$. Recall that excluded variables are necessarily decreed variables, whereas included variables may be neutral or decreed. Hence, in the component $x^t \in \mathcal{X}^t(x^{\mathrm{m}})$, some included variables contained may be decreed by the meta component $x^{\mathrm{m}} \in \mathcal{X}^{\mathrm{m}}$, which justifies the parametrization of the set $\mathcal{X}^t(x^{\mathrm{m}})$.

Two additional remarks follow. First, meta variables are always included, thus the meta set $\mathcal{X}^{\mathrm{m}}$ has no dependency. Secondly, a parametrized set $\mathcal{X}^t(x^{\mathrm{m}})$ is a subset of the set that contains all possible components $\mathcal{X}^t$, such that

$$\mathcal{X}^t(x^{\mathrm{m}}) \subseteq \mathcal{X}^t = \bigcup_{x^{\mathrm{m}} \in \mathcal{X}^{\mathrm{m}}} \mathcal{X}^t(x^{\mathrm{m}}), \tag{6}$$

where $t \in \{\mathrm{cat}, \mathrm{nom}, \mathrm{ord}, \mathrm{qnt}, \mathrm{int}, \mathrm{con}\}$. A component y^t is said to be incompatible with the meta component x^{m}, if $y^t \in \mathcal{X}^t$ and $y^t \notin \mathcal{X}^t(x^{\mathrm{m}})$. The set $\mathcal{X}^t$ contains all possible components, including incompatibles ones. The compatible and incompatible components are further discussed in Section 3.2.1.

In the MLP example, a continuous component y^{con} that contains the decay λ (see Table 1) is incompatible with the meta component $x^{\mathrm{m}} = (l, \mathrm{Adam})$. Indeed, if $o = \mathrm{Adam}$ and y^{con} is a continuous component that contains the decay, then $y^{\mathrm{con}} \in \mathcal{X}^{\mathrm{con}}$ and $y^{\mathrm{con}} \notin \mathcal{X}^{\mathrm{con}}(l, \mathrm{Adam})$.

Moreover, if all variables of type $t \in \{\mathrm{cat}, \mathrm{nom}, \mathrm{ord}, \mathrm{qnt}, \mathrm{int}, \mathrm{con}\}$ are neutral variables, then no parametrization is necessary, and therefore $\mathcal{X}^t(x^{\mathrm{m}}) = \mathcal{X}^t$.

The possibility of having incompatible components justifies why the domain $\mathcal{X}$ from Definition 6 is formulated with a categorical parametrized set $\mathcal{X}^{\mathrm{cat}}(x^{\mathrm{m}})$ and a quantitative parametrized set $\mathcal{X}^{\mathrm{qnt}}(x^{\mathrm{m}})$ rather than the categorical set $\mathcal{X}^{\mathrm{cat}}$ and a quantitative set $\mathcal{X}^{\mathrm{qnt}}$. Indeed, for a given meta component $x^{\mathrm{m}} \in \mathcal{X}^{\mathrm{m}}$, the categorical and quantitative components reside in their parametrized sets, such that $x^{\mathrm{cat}} \in \mathcal{X}^{\mathrm{cat}}(x^{\mathrm{m}})$ and $x^{\mathrm{qnt}} \in \mathcal{X}^{\mathrm{qnt}}(x^{\mathrm{m}})$, in order to take into account that some categorical or quantitative variables may be decreed by the given meta component $x^{\mathrm{m}} \in \mathcal{X}^{\mathrm{m}}$.

Moreover, the meta component x^{m} may affect the dimension $n^t(x^{\mathrm{m}})$ of the component $x^t \in \mathcal{X}^t(x^{\mathrm{m}})$. Indeed, some included variables of type t contained in the component $x^t \in \mathcal{X}^t(x^{\mathrm{m}})$ may be decreed by the meta component x^{m}, thus the number of included variables in this component may vary with the meta component x^{m}. In simpler terms, the dimension of the component x^t may vary with the meta component x^{m}. Hence, the dimension of the component x^t is a function $n^t : \mathcal{X}^{\mathrm{m}} \to \mathbb{N}$. Notably in the MLP example in Table 1, the number of hidden layers l decrees the number of units u_i in the hidden layers, which affects the number of integer variables. Thus, the dimension of the integer component $x^{\mathrm{int}} \in \mathcal{X}^{\mathrm{int}}(x^{\mathrm{m}})$ is determined by the meta component x^{m}.

3.2.1 Partition of Components into Roles

Following the discussion of a parametrized set $\mathcal{X}^t(x^{\mathrm{m}})$ in Section 3.2, it may be necessary or simply useful in some cases to explicitly distinguish the neutral and decreed-included variables of a given type $t \in \{\mathrm{cat}, \mathrm{nom}, \mathrm{ord}, \mathrm{qnt}, \mathrm{int}, \mathrm{con}\}$. This section serves as a complementary discussion of the domain $\mathcal{X}$ and the parametrized set $\mathcal{X}^t(x^{\mathrm{m}})$.

A component $x^t \in \mathcal{X}^t(x^{\mathrm{m}})$ of type $t \in \{\mathrm{cat}, \mathrm{nom}, \mathrm{ord}, \mathrm{qnt}, \mathrm{int}, \mathrm{con}\}$ may be partitioned into its neutral and decreed-included variables partitioned such that

$$x^t = (x^t_{\mathrm{neu}}, x^t_{\mathrm{dec}}) \in \mathcal{X}^t(x^{\mathrm{m}}), \tag{7}$$

where x^t_{neu} is the neutral-t component which contains the neutral variables of type t and x^t_{dec} is the decreed-included-t component which contains the decreed-included variables of type t for a given meta component $x^{\mathrm{m}} \in \mathcal{X}^{\mathrm{m}}$. Based on the partition in Eq. (7), a parametrized set $\mathcal{X}^t(x^{\mathrm{m}})$ of type $t \in \{\mathrm{cat}, \mathrm{nom}, \mathrm{ord}, \mathrm{qnt}, \mathrm{int}, \mathrm{con}\}$ may be formulated as a Cartesian product such that

$$\mathcal{X}^t(x^{\mathrm{m}}) = \mathcal{X}^t_{\mathrm{neu}} \times \mathcal{X}^t_{\mathrm{dec}}(x^{\mathrm{m}}), \tag{8}$$

where $\mathcal{X}^t_{\mathrm{neu}}$ is the neutral-t set that contains all neutral-t components x^t_{neu} and $\mathcal{X}^t_{\mathrm{dec}}(x^{\mathrm{m}})$ is the decreed-included-t set that contains all decreed-included-t components x^t_{dec} for a given meta component $x^{\mathrm{m}} \in \mathcal{X}^{\mathrm{m}}$. Definition 6 may be further detailed with Eq. (8), such that

$$\begin{aligned} \mathcal{X} = \Big\{ \ (x^{\mathrm{m}}, x^{\mathrm{cat}}, x^{\mathrm{qnt}}) \quad : \ & x^{\mathrm{m}} \in \mathcal{X}^{\mathrm{m}}, \\ & x^{\mathrm{cat}} = (x^{\mathrm{cat}}_{\mathrm{neu}}, x^{\mathrm{cat}}_{\mathrm{dec}}) \in \mathcal{X}^{\mathrm{cat}}_{\mathrm{neu}} \times \mathcal{X}^{\mathrm{cat}}_{\mathrm{dec}}(x^{\mathrm{m}}), \\ & x^{\mathrm{qnt}} = (x^{\mathrm{qnt}}_{\mathrm{neu}}, x^{\mathrm{qnt}}_{\mathrm{dec}}) \in \mathcal{X}^{\mathrm{qnt}}_{\mathrm{neu}} \times \mathcal{X}^{\mathrm{qnt}}_{\mathrm{dec}}(x^{\mathrm{m}}) \ \Big\}. \end{aligned} \tag{9}$$

The flexibility and thoroughness of the framework, provided by the types and roles of the variables, is succinctly displayed in Eq. (9).

Moreover, the decreed-excluded variables of type t are contained in the decreed-excluded-t component $\overline{x}^t_{\mathrm{dec}}$, such that

$$(x^t_{\mathrm{neu}}, x^t_{\mathrm{dec}}, \overline{x}^t_{\mathrm{dec}}) \in \mathcal{X}^t. \tag{10}$$

As discussed in Section 3.2, the component $x^t \in \mathcal{X}^t(x^{\mathrm{m}})$ is compatible with the meta component $x^{\mathrm{m}} \in \mathcal{X}^{\mathrm{m}}$, whereas $(x^t_{\mathrm{neu}}, x^t_{\mathrm{dec}}, \overline{x}^t_{\mathrm{dec}}) \in \mathcal{X}^t$ is incompatible since it contains decreed-excluded variables.

The partition of components into roles is not put forward in the rest of the document. However, it is implicitly present and illustrates several algorithmic subtleties, including the importance of distinguishing neutral and decreed variables during the optimization process.

3.2.2 Meta Set

The number of variables in the meta component is denoted $n^{\mathrm{m}} \in \mathbb{N}$. The meta component x^{m} belongs to the meta set $\mathcal{X}^{\mathrm{m}} \subseteq \mathbb{M}^{n^{\mathrm{m}}}$, which contains all the meta component x^{m}. In comparison to a parametrized set, the meta set $\mathcal{X}^{\mathrm{m}}$ is static, since the meta variables are always included variables. This also implies that the meta component x^{m} has a fixed dimension $n^{\mathrm{m}} \in \mathbb{N}$. Moreover, the set $\mathbb{M}^{n^{\mathrm{m}}}$ is a mixed set consisting of Cartesian products, such that

$$\mathbb{M}^{n^{\mathrm{m}}} = \mathbb{Z}^{n^{\mathrm{m}}_{\mathrm{cat}}} \times \mathbb{Z}^{n^{\mathrm{m}}_{\mathrm{int}}} \times \mathbb{R}^{n^{\mathrm{m}}_{\mathrm{con}}}, \tag{11}$$

where $n^{\mathrm{m}} = n^{\mathrm{m}}_{\mathrm{cat}} + n^{\mathrm{m}}_{\mathrm{int}} + n^{\mathrm{m}}_{\mathrm{con}}$ is the number of meta variables, $n^{\mathrm{m}}_{\mathrm{cat}}$ is the number of meta-categorical variables, $n^{\mathrm{m}}_{\mathrm{int}}$ is the number of meta-integer variables and $n^{\mathrm{m}}_{\mathrm{con}}$ is the number of meta-continuous variables.

In particular, note that $\mathbb{M}^{n^{\mathrm{m}}} = \mathbb{Z}^{n^{\mathrm{m}}_{\mathrm{cat}}} \times \mathbb{Z}^{n^{\mathrm{m}}_{\mathrm{int}}} = \mathbb{Z}^{n^{\mathrm{m}}}$ in the case where meta variables are strictly discrete variables. Formally, in that case, the meta set $\mathcal{X}^{\mathrm{m}}$ is a countable set since it is a subset of $\mathbb{M}^{n^{\mathrm{m}}}$, which is a Cartesian product of two countable sets. This case is common in practice and this allows to formulate $\mathcal{X}$ more schematically in Section 3.2.5.

3.2.3 Parametrized Categorical Set

The categories of each categorical variable can be mapped with a bijection to a subset of $\mathbb{Z}$. Hence, without any loss of generality, the parametrized categorical set $\mathcal{X}^{\mathrm{cat}}(x^{\mathrm{m}})$ is considered to be a subset of $\mathbb{Z}^{n^{\mathrm{cat}}(x^{\mathrm{m}})}$. However, this bijection does not imply that a metric notion is appropriate [29]. In other words, this bijection is only useful in terms of algorithmic implementations.

Since the categorical variable x^{cat}_j takes values from the set $L_j = \{l_1, l_2, \dots, l_j\}$, the parametrized categorical set $\mathcal{X}^{\mathrm{cat}}(x^{\mathrm{m}})$ is defined as

$$\mathcal{X}^{\mathrm{cat}}(x^{\mathrm{m}}) = \prod_{j=1}^{n^{\mathrm{cat}}(x^{\mathrm{m}})} L_j = \prod_{j=1}^{n^{\mathrm{cat}}(x^{\mathrm{m}})} \{l_1, l_2, \dots, l_j\}. \tag{12}$$

It may also be expressed as the Cartesian product between the parametrized unordered and ordered sets

$$\mathcal{X}^{\mathrm{cat}}(x^{\mathrm{m}}) = \mathcal{X}^{\mathrm{nom}}(x^{\mathrm{m}}) \times \mathcal{X}^{\mathrm{ord}}(x^{\mathrm{m}}) = \prod_{i=1}^{n^{\mathrm{nom}}(x^{\mathrm{m}})} L_i \times \prod_{j=1}^{n^{\mathrm{ord}}(x^{\mathrm{m}})} L_j, \tag{13}$$

which outlines the distinction between nominal and ordinal variables.

3.2.4 Parametrized Quantitative Set

The parametrized quantitative set $\mathcal{X}^{\mathrm{qnt}}(x^{\mathrm{m}})$ is a compact notation that describes a direct Cartesian product of the parametrized integer and continuous sets:

$$\mathcal{X}^{\mathrm{qnt}}(x^{\mathrm{m}}) = \mathcal{X}^{\mathrm{int}}(x^{\mathrm{m}}) \times \mathcal{X}^{\mathrm{con}}(x^{\mathrm{m}}) \subseteq \mathbb{Z}^{n^{\mathrm{int}}(x^{\mathrm{m}})} \times \mathbb{R}^{n^{\mathrm{con}}(x^{\mathrm{m}})}, \tag{14}$$

where $\mathcal{X}^{\mathrm{int}}(x^{\mathrm{m}}) \subseteq \mathbb{Z}^{n^{\mathrm{int}}(x^{\mathrm{m}})}$ is the parametrized integer set and $\mathcal{X}^{\mathrm{con}}(x^{\mathrm{m}}) \subseteq \mathbb{R}^{n^{\mathrm{con}}(x^{\mathrm{m}})}$ is the parametrized continuous set.

Again, the compact notation for the parametrized quantitative set $\mathcal{X}^{\mathrm{qnt}}(x^{\mathrm{m}})$ is particularly interesting for algorithms that optimize simultaneously the integer and continuous variables is employed.

3.2.5 Alternative Formulation of the Domain

The domain $\mathcal{X}$, as formulated in Definition 6, allows meta variables to be either categorical, integer or continuous. However, Definition 6 offers little insight regarding the visualization and construction of the domain $\mathcal{X}$, especially regarding the parametrized categorical set $\mathcal{X}^{\text{cat}}(x^{\text{m}})$ and the parametrized quantitative set $\mathcal{X}^{\text{qnt}}(x^{\text{m}})$. In the case where meta variables are all discrete (categorical or integer), such that the meta set $\mathcal{X}^{\text{m}} \subseteq \mathbb{M}^{n^{\text{m}}}$ is countable, a more visual and algorithmic formulation of the domain, based on [12, 13], is proposed:

$$\mathcal{X} = \bigcup_{x^{\text{m}} \in \mathcal{X}^{\text{m}}} \left(\{x^{\text{m}}\} \times \bigcup_{x^{\text{cat}} \in \mathcal{X}^{\text{cat}}(x^{\text{m}})} \left(\{x^{\text{cat}}\} \times \mathcal{X}^{\text{qnt}}(x^{\text{m}}) \right) \right). \tag{15}$$

Following the same logic as in Eq. (15), the parametrized quantitative set $\mathcal{X}^{\text{qnt}}(x^{\text{m}})$ is formulated as:

$$\mathcal{X}^{\text{qnt}}(x^{\text{m}}) \;=\; \mathcal{X}^{\text{int}}(x^{\text{m}}) \times \mathcal{X}^{\text{con}}(x^{\text{m}}) \;=\; \bigcup_{x^{\text{int}} \in \mathcal{X}^{\text{int}}(x^{\text{m}})} \left(\{x^{\text{int}}\} \times \mathcal{X}^{\text{con}}(x^{\text{m}}) \right) \tag{16}$$

Schematically, the parametrized quantitative set $\mathcal{X}^{\text{qnt}}(x^{\text{m}})$ can be visualized as the union of multiple layers, where each layer is a Cartesian product of a parametrized continuous set $\mathcal{X}^{\text{con}}(x^{\text{m}})$ with an integer component $x^{\text{int}} \in \mathcal{X}^{\text{int}}(x^{\text{m}})$, which is illustrated in Fig. 5b. In Fig. 5b, each layer shares the same continuous set $\mathcal{X}^{\text{con}}(x^{\text{m}})$, whereas each layer has a distinct integer component $x^{\text{int}} \in \mathcal{X}^{\text{int}}(x^{\text{m}})$. The quantitative set $\mathcal{X}^{\text{qnt}}(x^{\text{m}})$ is represented as a box containing all the possible unions described in Eq. (16).

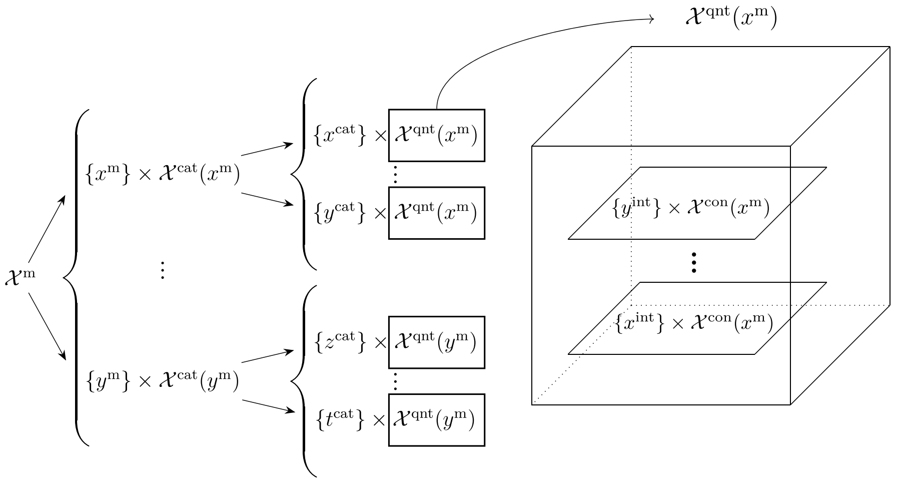
A visualization of the entire domain $\mathcal{X}$ can be built upon the abstraction of the quantitative set $\mathcal{X}^{\text{qnt}}(x^{\text{m}})$ illustrated in Fig. 5b. In Fig. 5a, the quantitative sets are represented as small rectangles, following the abstraction from Fig. 5b. The left-curly brackets represents the unions in the Eq. (15), from left to right. Furthermore, the formulation of the domain $\mathcal{X}$ in Eq. (15) may be understood and visualize as an explicit enumeration of all the possible points, similarly to a set of all possible components $\mathcal{X}^t$ in Eq. (6).

3.3 Feasible Set

Similarly to decreed variables, some constraints may be included or excluded of the problem. To model these constraints, the next definition, based on Definition 3 (decreed variable), is proposed.

Definition 8 (Decreed constraint). A constraint $c^{\text{m}} : \mathcal{X} \to \overline{\mathbb{R}}$ is a decreed constraint if its inclusion or exclusion is determined by values of a meta variable.

In the MLP example discussed in Section 2, each constraint $u_i \leq u_{i-1}\ \forall i \in \{2, 3, \ldots, l^{\max}\}$ is decreed by the number of hidden layers $l \in \{0, 1, \ldots, l^{\max}\}$.


$\mathcal{X}^{\mathrm{m}}$
$\{x^{\mathrm{m}}\} \times \mathcal{X}^{\mathrm{cat}}(x^{\mathrm{m}})$
$\{y^{\mathrm{m}}\} \times \mathcal{X}^{\mathrm{cat}}(y^{\mathrm{m}})$
$\{x^{\mathrm{cat}}\} \times \mathcal{X}^{\mathrm{qnt}}(x^{\mathrm{m}})$
$\{y^{\mathrm{cat}}\} \times \mathcal{X}^{\mathrm{qnt}}(x^{\mathrm{m}})$
$\{z^{\mathrm{cat}}\} \times \mathcal{X}^{\mathrm{qnt}}(y^{\mathrm{m}})$
$\{t^{\mathrm{cat}}\} \times \mathcal{X}^{\mathrm{qnt}}(y^{\mathrm{m}})$
$\mathcal{X}^{\mathrm{qnt}}(x^{\mathrm{m}})$
$\{y^{\mathrm{int}}\} \times \mathcal{X}^{\mathrm{con}}(x^{\mathrm{m}})$
$\{x^{\mathrm{int}}\} \times \mathcal{X}^{\mathrm{con}}(x^{\mathrm{m}})$

(a) Visualization of Equation (15) **(b)** Visualization of Equation (16)

Fig. 5 Visualization of the domain $\mathcal{X}$

The meta component x^{m}, which contains all the meta variables, decrees all the decreed constraint. In other words, the decreed constraints that are included in the problem are determine by the meta component x^{m} since it contains all the meta variables. From this remark, the set of decreed-included constraints is defined.

Definition 9 (Set of decreed-included constraints). The set of decreed-included constraints $C^{\mathrm{m}}(x^{\mathrm{m}})$ is the set that contains all the included constraints that are decreed by the meta component x^{m}.

Similarly to a parametrized set $\mathcal{X}^t(x^{\mathrm{m}})$ defined in Definition 7, the dependency of $C^{\mathrm{m}}(x^{\mathrm{m}})$ with x^{m} is defined through a parametrization with respect to the meta component x^{m}. Moreover, the set of decreed-included constraints $C^{\mathrm{m}}(x^{\mathrm{m}})$ is a subset of the set of decreed constraints C^{m}. A constraint $\hat{c}^{\mathrm{m}} \in C^{\mathrm{m}}$ is either included or excluded, whereas $c^{\mathrm{m}} \in C^{\mathrm{m}}(x^{\mathrm{m}})$ is an included constraint, decreed by the meta component x^{m} (more precisely, a meta variable contained in the meta component x^{m}). In the MLP example, the set of decreed constraints is

$$C^{\mathrm{m}} = \{u_i - u_{i-1} \leq 0 \ : \ \forall i \in \{2,3,\dots,l^{\max}\}\} \tag{17}$$

and the set of decreed-included constraints is

$$C^{\mathrm{m}}(x^{\mathrm{m}}) = C^{\mathrm{m}}(l,o) = \begin{cases} \emptyset, & \text{if } l \in \{0,1\} \\ \{u_i - u_{i-1} \leq 0 \ : \ \forall i \in \{2,3,\dots,l\}\} \subseteq C^{\mathrm{m}}, & \text{otherwise} \end{cases} \tag{18}$$

where $l \leq l^{\max}$.

Contrary to decreed constraints, some constraints are not decreed by the meta component x^{m}. Based on Definition 4 (neutral variable), the definition of neutral constraint is given.

Definition 10 (Neutral constraint). A constraint $c_j : \mathcal{X} \to \overline{\mathbb{R}}$ is a neutral constraint if it is always included.

In the MLP example, the neutral constraint is $c(x) = \alpha l - r \leq 0$, which is always included no matter the meta component x^{m}.

To define the feasible set Ω, the neutral constraints and decreed constraints are distinguished.

Definition 11 (Feasible set). The feasible set $\Omega \subseteq \mathcal{X}$ is the domain $\mathcal{X}$ defined by constraints:

$$\begin{aligned} \Omega = \Big\{ \ (x^{\mathrm{m}}, x^{\mathrm{cat}}, x^{\mathrm{qnt}}) \in \mathcal{X} \ \ &: c_i(x) \leq 0, \ \forall i \in \{1, 2, \dots, p\}, \\ &\quad c^{\mathrm{m}}(x) \leq 0, \ \forall c^{\mathrm{m}} \in C^{\mathrm{m}}(x^{\mathrm{m}}) \ \Big\} \end{aligned} \tag{19}$$

where c_i are the neutral constraints with $p \in \mathbb{N}$ and $C^{\mathrm{m}}(x^{\mathrm{m}})$ is the set of decreed-included constraints, which is parametrized with respect to meta component x^{m}. The number of included constraints that are decreed by the meta component x^{m} is simply $|C^{\mathrm{m}}(x^{\mathrm{m}})|$.

3.4 Mathematical Modeling of the MLP Example

Each hyperparameter is identified with its variable type and role in Table 2.

The following observations can be made. First, the number of units u_i in the hidden layers are typed as integer variables. Although they affect the network architecture, they are not meta variables because they do not decree other variables. More precisely, they do not affect the dimension of the integer component, since they do not decree any other hyperparameters. Second, the number of hidden layers l is a meta variable, since it decrees the units u_i and thus it affects the dimension of a component $x^{\mathrm{int}} \in \mathcal{X}^{\mathrm{int}}(x^{\mathrm{m}})$. Third, the activation function $a \in \{\mathrm{ReLU}, \mathrm{Sigmoid}\}$ is nominal variable, since it is a qualitative discrete variable that belongs to a set with no appropriate metric and no order. Fourth, the optimizer o is a meta-nominal variable as it decrees some continuous hyperparameters of the problem and takes a category in a set with no appropriate metric and no order.

3.4.1 Components and Sets

The meta set $\mathcal{X}^{\mathrm{m}}$ is the Cartesian product between the domains of the two meta variables, the number of hidden layers l and the optimizer o, thus the meta component x^{m} and the meta set $\mathcal{X}^{\mathrm{m}}$ are:

$$x^{\mathrm{m}} = (l, o) \in \mathcal{X}^{\mathrm{m}} = \{0, 1, \dots, l^{\max}\} \times \{\mathrm{Adam}, \mathrm{ASGD}\}. \tag{20}$$

Then, the only categorical variable is the activation function a, which is a neutral variable. Thus, $\mathcal{X}^{\mathrm{cat}}(x^{\mathrm{m}}) = \mathcal{X}^{\mathrm{cat}}$ in the example, since no parametrization of the

Table 2 Hyperparameters with their variable type and role

Hyperparameter	Variable	Domain	Type	Role
Learning rate	r	$]0,1[$	continuous	neutral
Activation function	a	{ReLU, Sigmoid}	nominal	neutral
# of hidden layers	l	$\{0,1,\dots,l^{\max}\}$	meta-integer	meta
# of units hidden layer i	u_i	$\{u_i^{\min}, u_i^{\min}+1,\dots,u_i^{\max}\}$	integer	decreed
Optimizer	o	{Adam, ASGD}	meta-nominal	meta
if $o=$ ASGD				
decay	λ	$]0,1[$	continuous	decreed
power update	α	$]0,1[$	continuous	decreed
averaging start	t_0	]1E3, 1E8[	continuous	decreed
if $o=$ Adam				
running average 1	β_1	$]0,1[$	continuous	decreed
running average 2	β_2	$]0,1[$	continuous	decreed
numerical stability	ϵ	$]0,1[$	continuous	decreed

categorical set is necessary. Following this, the categorical component x^{cat} and the categorical set $\mathcal{X}^{\text{cat}}$ are:

$$x^{\text{cat}} = a \in \mathcal{X}^{\text{cat}} = \{\text{ReLU}, \text{Sigmoid}\}. \tag{21}$$

Moreover, the integer component is directly the vector of units in the hidden layers, such that $x^{\text{int}} = u(l) = (u_1, u_2, \dots, u_l)$. All the integer variables are decreed by the meta component x^{m} and more specifically the number of hidden layers l. The integer component x^{int} and the parametrized integer set $\mathcal{X}^{\text{int}}$ are:

$$x^{\text{int}} = \begin{cases} \emptyset \text{ (excluded)}, & \text{if } l=0 \\ (u_1,u_2,\dots,u_l) \in \mathcal{X}^{\text{int}}(x^{\text{m}}) = \mathcal{X}^{\text{int}}(l) = \prod_{i=1}^{l} \{u_i^{\min}, u_i^{\min}+1,\dots,u_i^{\max}\} \subseteq \mathbb{N}^l, & \text{if } l \geq 1 \end{cases} \tag{22}$$

where $u_i^{\min}$ and $u_i^{\max}$ are respectively the minimum and the maximum of units allowed for each hidden layer $i \in \{1,2,\dots,l\}$, l is the number of hidden layers and $u(0)$ is an empty vector.

Finally, all continuous variables are decreed by the optimizer o, except for the learning rate r. Thus, the continuous component x^{con} is decreed by the meta component x^{m}, implying that the continuous set requires a parametrization. The continuous component x^{con} and the parametrized continuous set $\mathcal{X}^{\text{con}}(x^{\text{m}})$ are:

$$x^{\text{con}} \in \mathcal{X}^{\text{con}}(x^{\text{m}}) = \begin{cases} \mathcal{X}^{\text{con}}(\text{Adam}) =]0,1[^4 \subseteq \mathbb{R}^4, & \text{if o= Adam }, \\ \mathcal{X}^{\text{con}}(\text{ASGD}) =]0,1[^3 \times]1\text{E}3, 1\text{E}8[\subseteq \mathbb{R}^4, & \text{if o= ASGD }. \end{cases} \tag{23}$$

The continuous component $x^{\text{con}} \in \mathcal{X}^{\text{con}}$ contains neutral and decreed-included variables, hence it may be partitioned into its neutral and decreed-included variables. The neutral-continuous component $x^{\text{con}}_{\text{neu}}$ and the neutral-continuous set $\mathcal{X}^{\text{con}}_{\text{neu}}$ are:

$$x^{\mathrm{con}}_{\mathrm{neu}} = r \in \mathcal{X}^{\mathrm{con}}_{\mathrm{neu}} =]0, 1[\tag{24}$$

where r is the learning rate. Moreover, the decreed-included-continuous component $x^{\mathrm{con}}_{\mathrm{dec}}$ and the decreed-included-continuous set $\mathcal{X}^{\mathrm{con}}_{\mathrm{neu}}(x^{\mathrm{m}})$ are:

$$x^{\mathrm{con}}_{\mathrm{dec}} \in \mathcal{X}^{\mathrm{con}}_{\mathrm{dec}}(x^{\mathrm{m}}) = \begin{cases}]0, 1[^3 \subseteq \mathbb{R}^3, & \text{if o= Adam ,} \\ ]0, 1[^2 \times]1\mathrm{E}3, 1\mathrm{E}8[\subseteq \mathbb{R}^3, & \text{if o= ASGD .} \end{cases} \tag{25}$$

For the sake of simplicity, the domain of the units in the hidden layers u_i and the number of hidden layer l are set as: $u_i^{\min} = 100$ and $u_i^{\max} = 300$, $\forall i$ and $l \in \{2, 3\}$ in Table 1. With $l \in \{2, 3\}$, the meta set (20) can be explicit as:

$$\mathcal{X}^{\mathrm{m}} = \{(2, \mathrm{Adam}), (3, \mathrm{Adam}), (2, \mathrm{ASGD}), (3, \mathrm{ASGD})\}. \tag{26}$$

Moreover, the parametrized integer set $\mathcal{X}^{\mathrm{int}}(x^{\mathrm{m}})$ can also be explicit:

$$\mathcal{X}^{\mathrm{int}}(x^{\mathrm{m}}) = \{100, 101, \dots, 300\}^l = U^l = \begin{cases} \{100, 101, \dots, 300\}^2, & \text{if } l = 2 \\ \{100, 101, \dots, 300\}^3, & \text{if } l = 3. \end{cases} \tag{27}$$

The parametrized categorical and continuous sets remain unchanged.

3.4.2 Constraints

In the example there is a neutral constraint and decreed constraints. The neutral constraint can be easily expressed as $c(x) = c(l, r) = \alpha l - r \leq 0$. The set of decreed constraints is

$$C^{\mathrm{m}} = \left\{ u_i - u_{i-1} \leq 0 \ : \ i \in \{2, 3, \dots, l^{\max}\} \right\} \tag{28}$$

and the set of decreed-included constraints is

$$C^{\mathrm{m}}(x^{\mathrm{m}}) = C^{\mathrm{m}}(l) = \begin{cases} \emptyset \text{ (excluded)}, & \text{if } l < 2, \\ \left\{ u_i - u_{i-1} \leq 0 \ : \ i \in \{2, 3, \dots, l\}, \right\} \text{ (included)}, & \text{if } l \geq 2, \end{cases} \tag{29}$$

which can be further detailed since $l \in \{2, 3\}$

$$C^{\mathrm{m}}(2) = \{u_2 - u_1 \leq 0\}, \quad C^{\mathrm{m}}(3) = \{u_3 - u_2 \leq 0, \ u_2 - u_1 \leq 0\}. \tag{30}$$

In this particular example, the number of neutral constraint is $p = 1$ and the number of included constraints that decreed by the meta component is $|C(x^{\mathrm{m}})| = l - 1$.

3.4.3 Visualization of the Domain and the Feasible Set

In the MLP example, the meta set $\mathcal{X}^{\mathrm{m}}$ is countable (see Eq. 26). Hence, the alternative formulation of the domain $\mathcal{X}$ in Eq. (15) and the feasible set Ω in Definition 19 of the MLP example may be visualized in Fig. 6, in which $\mathcal{X}^{\mathrm{con}}_{\mathrm{A}} = \mathcal{X}^{\mathrm{con}}(\mathrm{Adam})$, $\mathcal{X}^{\mathrm{con}}_{\mathrm{B}} = \mathcal{X}^{\mathrm{con}}(\mathrm{ASGD})$ and $U^l = \{100, 101, \dots, 300\}^l$ (domain of the units u_i).

The upper part of Fig. 6 (above the dotted line) represents the alternative formulation of the domain $\mathcal{X}$ in Eq. (15). The parametrized quantitative sets $\mathcal{X}^{\text{qnt}}(x^{\text{m}})$ are illustrated as rectangles and the unions from left to right in Eq. (15) are viewed from top to bottom in Fig. 6. Moreover, the parametrized quantitative sets are expressed explicitly, such that $\mathcal{X}^{\text{qnt}}(x^{\text{m}}) = \mathcal{X}^{\text{qnt}}(l, o) = \mathcal{X}^{\text{int}}(l) \times \mathcal{X}^{\text{con}}(o) = U^l \times]0,1[\times \mathcal{X}_i^{\text{con}}$, where $i \in \{A, B\}$. The lower part of Fig. 6 schematizes the constraints. The included constraints decreed by a meta component x^{m}, are contained in the set of decreed-included constraints $C^{\text{m}}(x^{\text{m}})$. The neutral constraint is always included and unaffected by the meta component x^{m}, hence it is not assign to a specific meta component x^{m} comparatively to decreed constraints: this representation shows the neutral aspect of neutral constraints. Altogether, the upper and lower parts Fig. 6 synthesize the feasible set Ω of the MLP example.

In the literature review, it has been discussed that some optimization approaches tackle categorical variables by solving many subproblems in which a categorical component x^{cat} is fixed. Indeed, in [12, 14] the MADS algorithm was applied to a continuous space where a discrete component, which contained meta, categorical and integer variables, was fixed. This idea can be generalized to the proposed notation system. For example, assume that $x^{\text{m}} = (2, \text{Adam})$ and $x^{\text{cat}} = \text{ReLU}$ are selected and fixed. Then, the objective function f could then be optimized on the parametrized quantitative $\mathcal{X}^{\text{qnt}}(\text{Adam}, 2)$ with both the meta and categorical components fixed. Subproblems are further discussed in the next Section 4 and more particularly in Section 4.1.

4 Solution Strategies

Most blackbox approaches in mixed-variable optimization are built upon two strategies. One solution strategy consists of solving many subproblems in which some selected components are fixed. Another strategy consists of formulating a less costly problem that selects a candidate point to be evaluated by the more costly objective function f. Some methods rely on both strategies.

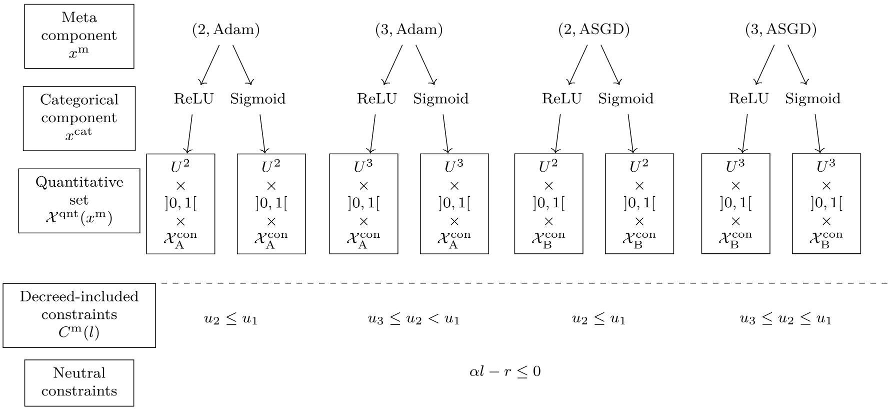

Meta component x^{m}
Categorical component x^{cat}
Quantitative set $\mathcal{X}^{\text{qnt}}(x^{\text{m}})$
(2, Adam)
(3, Adam)
(2, ASGD)
(3, ASGD)
ReLU
Sigmoid
$U^2 \times]0,1[\times \mathcal{X}^{\text{con}}_{\text{A}}$
$U^3 \times]0,1[\times \mathcal{X}^{\text{con}}_{\text{A}}$
$U^2 \times]0,1[\times \mathcal{X}^{\text{con}}_{\text{B}}$
$U^3 \times]0,1[\times \mathcal{X}^{\text{con}}_{\text{B}}$
Decreed-included constraints $C^{\text{m}}(l)$
$u_2 \leq u_1$
$u_3 \leq u_2 < u_1$
$u_2 \leq u_1$
$u_3 \leq u_2 \leq u_1$
Neutral constraints
$\alpha l - r \leq 0$

Fig. 6 Diagram of the domain $\mathcal{X}$ and the constraints for the MLP example

For example, direct search methods [8, 11, 12, 14] divide the main problem into many subproblems, in which the objective function *f* is optimized on a continuous space for a fixed discrete component x^d. Bayesian optimization (BO) formulates an auxiliary problem, with a fixed acquisition function and a probabilistic surrogate, and then selects a candidate point that is subsequently evaluated by the objective function *f*.

The two strategies are respectively defined as the subproblems strategy and the auxiliary problem strategy. These strategies are the basis of the general algorithmic framework, since most algorithms that tackle mixed-variable blackbox optimization conceptually rely on solving many subproblems or on an auxiliary problem.

The purpose of this section is to illustrate that the framework notation may be easily adapted to the main blackbox approaches in mixed-variable optimization. More precisely, direct search and heuristic approaches are discussed through the subproblems strategy in Section 4.1 and the BO approach is discussed through the auxiliary problem strategy in Section 4.2.

4.1 Subproblems

The motivation of dividing a main problem into many subproblems arises from two rationales: 1) there are methods that treat quantitative (integer-continuous) problems, or even categorical-integer-continuous problems (mostly with an auxiliary problem strategy); 2) there are few efficient methods that address mixed-variable optimization problems with both meta and categorical variables.

In the context of this work, subproblems are obtained by fixing values of meta and categorical components. In [8, 11, 12, 14], the component that is fixed is the discrete component, which contains categorical variables. Secondly, note that there's no particular interest fixing the integer or continuous components, since they can be properly optimized in practice.

To further formalize the subproblems, the objective subfunction must be first defined.

Definition 12 (Objective subfunction). An objective subfunction g is the objective function *f* with a single or many fixed components. The objective subfunction is said to be parametrized with respect to the fixed component(s).

From Definition 12, it should be noted that there is a direct correspondence between the fixed component(s) and its subproblem. In other words, a specific subproblem may be referred by its fixed component(s). Again, in Definition 12, the components that are interesting to fix are the meta component x^{m} and the categorical component x^{cat}. In this work, only the quantitative subproblems strategy, in which both the meta and categorical components, is detailed.

4.1.1 Quantitative Subproblems

In the quantitative subproblems strategy, the meta component x^{m} and the categorical component x^{cat} are fixed, in order to generate quantitative subproblems (one per couple $(x^{\mathrm{m}}, x^{\mathrm{cat}})$). Fixing a meta component $x^{\mathrm{m}} \in \mathcal{X}^{\mathrm{m}}$ simplifies the optimization problem, since the included variables, the included constraints and the dimension in the subproblems are determined. In addition, fixing the categorical components also further simplifies the optimization problem. Indeed, with both the meta and categorical components fixed, the subproblems are a quantitative blackbox optimization problem, where the included variables are either integer or continuous variables. In practice, there are efficient methods to tackle these quantitative subproblems.

For the quantitative subproblems strategy, the objective subfunction $g : \mathcal{X}^{\mathrm{qnt}}(x^{\mathrm{m}}) \to \overline{\mathbb{R}}$, parametrized with respect to the meta component $x^{\mathrm{m}} \in \mathcal{X}^{\mathrm{m}}$ and the categorical component $x^{\mathrm{cat}} \in \mathcal{X}^{\mathrm{cat}}(x^{\mathrm{m}})$, is defined as:

$$g(x^{\mathrm{qnt}}; x^{\mathrm{cat}}, x^{\mathrm{m}}) = f(x^{\mathrm{m}}, x^{\mathrm{cat}}, x^{\mathrm{qnt}}), \quad \text{where } x^{\mathrm{m}} \in \mathcal{X}^{\mathrm{m}} \text{ and } x^{\mathrm{cat}} \in \mathcal{X}^{\mathrm{cat}}(x^{\mathrm{m}}) \text{ are fixed.} \tag{31}$$

Thus, for a fixed meta component $x^{\mathrm{m}} \in \mathcal{X}^{\mathrm{m}}$ and a fixed categorical component $x^{\mathrm{cat}} \in \mathcal{X}^{\mathrm{cat}}(x^{\mathrm{m}})$, a quantitative subproblem may be formulated as

$$\begin{array}{llll} (P^{\mathrm{qnt}}) & \displaystyle\min_{x^{\mathrm{qnt}} \in \mathcal{X}^{\mathrm{qnt}}(x^{\mathrm{m}})} & g(x^{\mathrm{qnt}}; x^{\mathrm{m}}, x^{\mathrm{cat}}) & \\ & \text{s.t.} & c^{\mathrm{m}}(x) \le 0, & \forall c^{\mathrm{m}} \in C^{\mathrm{m}}(x^{\mathrm{m}}), \\ & & c_i(x) \le 0, & \forall i \in \{1, 2, \dots, p\}. \end{array} \tag{32}$$

where P^{qnt} stands for quantitative subproblem. Moreover, note that the constraints of the problem are treated directly within the subproblems of the form (P^{qnt}).

4.1.2 Exploration of Subproblems

There is a direct correspondence between the fixed component(s) and their subproblem, hence the exploration of subproblems may be done accordingly to the fixed components. Solving subproblems may be done directly with simple heuristics, such as random searches on the meta and categorical components. However, extra work is required in a direct search framework. Qualitative variables, such as the categorical variables, do not posses intuitive neighborhoods nor directions of exploration. Hence, the meta set $\mathcal{X}^{\mathrm{m}}$, which may contain meta components with meta-categorical variables, and the parametrized categorical set $\mathcal{X}^{\mathrm{cat}}(x^{\mathrm{m}})$ are both endowed with a user-defined neighborhood mapping. To formalize the exploration of subproblems, the following definition based on [8, 11, 12, 14], is proposed.

Definition 13 (User-defined neighborhood mapping). For any $t \in \{\mathrm{m}, \mathrm{cat}\}$, a user-defined neighborhood mapping $\mathcal{N}^t$ assigns a user-defined neighborhood $\mathcal{N}^t(x) \subseteq \mathcal{X}^t$ to a point $x \in \mathcal{X}$, such that each neighbor $y^t \in \mathcal{N}^t(x)$ is a component of type t that is determined by a given rule $r^t : \mathcal{X} \to \mathcal{X}^t$:

$$\begin{aligned} \mathcal{N}^t \ &: \ \mathcal{X} \to \mathcal{P}(\mathcal{X}^t) \\ & \quad x \mapsto \left\{ y^t \in \mathcal{X}^t \ : \ y^t = r^t(x), \ r^t \in \mathcal{R}^t(x) \right\} \subseteq \mathcal{X}^t \end{aligned} \tag{33}$$

where $r^t \in \mathcal{R}^t(x)$ is a rule that assigns a neighbor $y^t = r^t(x) \in \mathcal{X}^t$ to a point $x \in \mathcal{X}$, $\mathcal{R}^t(x)$ is a set of rules defined for the given point $x \in \mathcal{X}$ and $\mathcal{P}(\mathcal{X}^t)$ is the powerset of $\mathcal{X}^t$, which is denoted as the codomain of the mapping $\mathcal{N}^t$ to indicate $\mathcal{N}^t(x)$ can either be:

1. $\mathcal{N}^t(x) = \emptyset$, such that x has no neighbor of type t;
2. $\mathcal{N}^t(x) = \{y^t\}$, such that x has a single neighbor of type t;
3. $\mathcal{N}^t(x) \subseteq \mathcal{X}^t$, such that x^t has multiple neighbors of type t.

The set of rules $\mathcal{R}^t(x)$ embeds the generality of the user-defined neighborhood $\mathcal{N}^t(x)$. Indeed, a rule $r^t \in \mathcal{R}^t(x)$ must only respect the following mapping $r : \mathcal{X} \to \mathcal{X}^t$, which indicates that a component $y^t = r^t(x) \in \mathcal{X}^t$, called a neighbor, is assigned to a point $x \in \mathcal{X}$. In practice, it is from the these rules that user-defined neighborhoods are generated and implemented. Moreover, two issues are specific to the categorical case $t = \text{cat}$: 1) the set $\mathcal{X}^{\text{cat}}$ is the parametrized categorical set: $\mathcal{X}^t = \mathcal{X}^{\text{cat}}(x^{\text{m}})$; 2) the user-defined neighborhood mapping $\mathcal{N}^{\text{cat}}$ takes a point $x \in \mathcal{X}$ as an argument, which allows to take into account the decree property of meta variables for the user-defined neighborhood mapping $\mathcal{N}^{\text{cat}}$ and its constituent parts, such as the rules r^{cat}.

In the MLP example and using Eq. (20), the meta rules of the form $r^{\text{m}} : \mathcal{X} \to \mathcal{X}^{\text{m}}$, for a given point $y = (y^{\text{m}}, x^{\text{cat}}, x^{\text{qnt}}) \in \mathcal{X}$ with $y^{\text{m}} = (l, o)$, could be

$$r_1^{\text{m}}(y) = (l+1, o), \ r_2^{\text{m}}(y) = (l-1, o),$$
$$r_3^{\text{m}}(y) = (l, \overline{o}), \ r_4^{\text{m}}(y) = (l+1, \overline{o}), \ r_5^{\text{m}}(y) = (l-1, \overline{o}),$$

where $\overline{o}$ represents the other optimizer available. The set of rules would be:

$$\mathcal{R}^{\text{m}}(y) = \begin{cases} \left\{ r_1^{\text{m}}, r_3^{\text{m}}, r_4^{\text{m}} \right\}, & \text{if } l = 0 \\ \left\{ r_2^{\text{m}}, r_3^{\text{m}}, r_5^{\text{m}} \right\}, & \text{if } l = l^{\max} \\ \left\{ r_1^{\text{m}}, r_2^{\text{m}}, r_3^{\text{m}}, r_4^{\text{m}}, r_5^{\text{m}} \right\}, & \text{otherwise,} \end{cases} \tag{34}$$

with corresponding user-defined neighborhood

$$\mathcal{N}^{\text{m}}(y) = \begin{cases} \left\{ (l+1, o), (l, \overline{o}), (l+1, \overline{o}) \right\}, & \text{if } l = 0 \\ \left\{ (l-1, o), (l, \overline{o}), (l-1, \overline{o}) \right\}, & \text{if } l = l^{\max} \\ \left\{ (l+1, o), (l-1, o), (l, \overline{o}), (l+1, \overline{o}), (l-1, \overline{o}) \right\}, & \text{otherwise.} \end{cases} \tag{35}$$

The evaluations of the blackbox objective function f are generally costly, which implies that the user-defined neighborhood mappings have to set a trade-off between being exploratory and computationally expensive. Again, in practice, the user-defined neighborhood mappings $\mathcal{N}^{\text{m}}$ and $\mathcal{N}^{\text{cat}}$ are based on rules provided by a user. Thus, the compromise is set with the discretion of the user. To lower the number of evaluations, some polling strategies may be used in practice. Indeed, instead of exploring all the neighbors at a given iteration, an opportunistic strategy would stop the iteration if a neighbor that offers a better solution is determined and resume from that neighbor.

4.1.3 Direct Search Framework

Direct search methods with strict decrease are iterative algorithms that start with an initial point $x_{(0)}$ and seek a candidate point t whose objective function value $f(t)$ is strictly less than $f(x_{(k)})$, where $x_{(k)}$ is the current incumbent solution at iteration k. More precisely, at every iteration k, a set of trial points T is generated. Opportunistically, if a trial point $t \in T$ improves the objective function value, then it becomes the next incumbent solution $x_{(k+1)} = t$ and the iteration k terminates. Otherwise, the current incumbent solution remains unchanged, such that $x_{(k+1)} = x_{(k)}$ [1, 15]. In practice, stopping the iteration opportunistically reduces the number of evaluations required [1].

Moreover, direct search methods tackle blackbox optimization problems with two main mechanism: a global search strategy (diversification) and a poll that locally searches better solutions (intensification).

By its own, a poll is prone to miss out good point solutions. Indeed, the poll may get caught in a region with local minima or may neglect the exploration of promising regions that are far from the poll. For the meta set $\mathcal{X}^{\mathrm{m}}$ and parametrized categorical set $\mathcal{X}^{\mathrm{cat}}(x^{\mathrm{m}})$ the poll may be emulated with some user-defined neighborhood mappings $\mathcal{N}^{\mathrm{m}}$ and $\mathcal{N}^{\mathrm{cat}}$ respectively. The quality of a poll based on a user-defined neighborhood mapping, such as the meta and categorical polling, depends on the exhaustiveness of the set of rules $\mathcal{R}^{\mathrm{m}}$ and $\mathcal{R}^{\mathrm{cat}}$. Therefore, depending on the quality of implementation by the user and the dimensions of the problem, the poll, based on user-defined neighborhood mappings, is likely to neglect some promising components.

In that regard, a global search may help overcome this problem by evaluating scattered trial points (or components) with a flexible strategy that serves as a diversification mechanism. The global search is generally being done before the poll for opportunistic reasons, given that the global search may find a better or interesting point that deserves to be further explored with the poll. The global search is an optional step that often improves the overall quality of a solution and increases the convergence speed. Many generic and low-cost global search strategies exist, such as the random search, Latin hypercube sampling or a Nelder-Mead search [30], and more sophisticated and costly global search strategies can be implemented to generate promising trial points or unexplored regions, such as the Gaussian Processes (surrogate) paired with an acquisition function (auxiliary problem strategy) that quantifies the uncertainty and the potentiality of a point.

Algorithm 1 presents the main steps of a direct search methodology. The methodology consists of a quantitative subproblems strategy (see Section 4.1.1) paired with an exploration of subproblems that is done with user-defined neighborhood mappings $\mathcal{N}^{\mathrm{m}}$ and $\mathcal{N}^{\mathrm{cat}}$ from Definition 13.

Algorithm 1: Direct search main steps.

```
while stopping criteria not reached do
    1. Global search
      Select $t^{\mathrm{m}} \in \mathcal{X}^{\mathrm{m}}$ with a global meta exploration strategy
      Select $t^{\mathrm{cat}} \in \mathcal{X}^{\mathrm{cat}}(x^{\mathrm{m}})$ with a global categorical exploration strategy
      Let $t$ be obtained by solving the subproblem $(P^{\mathrm{qnt}})$ with $t^{\mathrm{m}}$ and $t^{\mathrm{cat}}$ fixed
    if $f(t) < f\left(x_{(k)}\right)$ then
        $x_{(k+1)} \leftarrow t$
    else

        2. Poll on user-defined neighborhoods
        for $t^{\mathrm{m}} \in \mathcal{N}^{\mathrm{m}}\left(x^{\mathrm{m}}_{(k)}\right)$ do
            for $t^{\mathrm{cat}} \in \mathcal{N}^{\mathrm{cat}}\left(x^{\mathrm{cat}}_{(k)}; t^{\mathrm{m}}\right)$ do
                Let $t$ be obtained by solving the subproblem $(P^{\mathrm{qnt}})$ with $t^{\mathrm{m}}$ and $t^{\mathrm{cat}}$
                 fixed
                if $f(t) < f\left(x_{(k)}\right)$ then
                    $x_{(k+1)} \leftarrow t$
                    break # Opportunistic strategy
                end
            end
        end
end
```

In Algorithm 1, the two main steps to tackle the meta and categorical variables with a direct search approach are compactly presented. For the global search and poll steps, an quantitative subproblem (P^{qnt}), which respects the formulation in Problem (38), is solved. Hence, the constraints of the problem are handled within the subproblems. Moreover, the solving of a subproblem (P^{qnt}) encapsulates many algorithmic details, such as a stopping criteria for a subproblem, as well as a global search and poll on the integer and continuous variables. Note that, a potential solver for the subproblems could be the MADS algorithm [14] which enables to treat simultaneously integer and continuous variables. For more details, see [15]. Then, additionally, constraints can be handled with the progressive barrier technique [31].

4.2 Auxiliary Problem

Auxiliary problems inexpensively allow to select candidate points to be evaluated by the true objective function f. Auxiliary problems are generally built from a surrogate model $\tilde{f}$ of the objective function f, an acquisition function α, as well as surrogates of each neutral constraint $\tilde{c}_j$, $j \in \{1, 2, \ldots, p\}$ and decreed constraint $\tilde{c}^{\mathrm{m}} \in C^{\mathrm{m}}$. The acquisition function α allows to select candidate points in promising regions (intensification) or in unexplored regions (exploration). The acquisition function α is generally applied to a surrogate model $\tilde{f}$ that quantifies the uncertainty of a point of its domain, and provides a prediction of the true objective function f. This is the case in

BO where $\tilde{f}$ is a GP probabilistic surrogate model. Other surrogate models can be considered, such as random forests, however the most common remains the GPs. In this section, only GP surrogate models are adapted to the notation framework, since they are the basis of BO, an important blackbox approach to tackle mixed-variable problems. Before discussing BO, the encoding of variables is discussed.

4.2.1 Encoding of Variables and Auxiliary Domain

BO methodologies (from Section 1.3) often tackle categorical variables by encoding them as quantitative variables. For instance, the categorical variables may be encoded by the emerging latent variables or simply with the popular one-hot encoding binary vectors relaxed into a continuous vector [21].

Definition 14 (Encoder). For any $t \in \{\text{cat}, \text{nom}, \text{ord}\}$ and iteration $k \in \mathbb{N}$, the encoder $\phi^t_{(k)}$, parametrized with respect to the meta component $x^{\text{m}} \in \mathcal{X}^{\text{m}}$, is a mapping that assigns an encoded component l^t to a component x^t, such that

$$\begin{array}{rccl} \phi^t_{(k)} & : \mathcal{X}^t(x^{\text{m}}) & \to & \mathcal{L}^t(x^{\text{m}}) \\ & x^t & \mapsto & l^t = \phi^t_{(k)}(x^t;x^{\text{m}}). \end{array} \tag{36}$$

An encoder $\phi^t_{(k)}$ may be updated at every iteration $k \in \mathbb{N}$, such as the latent variables discussed in Section 1.3. In order to take into account the decree properties of the meta variables, an encoder is parametrized with respect to the meta component $x^{\text{m}} \in \mathcal{X}^{\text{m}}$. In general, a meta variable may be a meta-categorical variable. However, in this work, the meta variables are not encoded for two reasons. First, the decreeing property of encoded meta variables may be ambiguous and difficult to conserve through sophisticated mappings, such as the latent variables. Secondly, there are categorical kernels that allow to avoid encoding categorical variables, hence in a BO framework, meta-categorical variables may be treated with these kernels.

One of the main purpose of encoding categorical variables (or equivalently categorical component) is to formulate an auxiliary problem in which these encoded variables possess mathematical properties, making them easier to manipulate. However, by encoding the categorical variables, the domain of the surrogate model may differ from the domain of the objective function $\mathcal{X}$. Hence, the auxiliary domain $\mathcal{X}_{\text{aux}}$ is defined as follows.

Definition 15 (Auxiliary domain). The auxiliary domain at an iteration $k \in \mathbb{N}$ is defined by:

$$\mathcal{X}_{\text{aux}} = \left\{ \begin{array}{ll} (x^{\text{m}}, l^{\text{cat}}, x^{\text{qnt}}) & : x^{\text{m}} \in \mathcal{X}^{\text{m}}, \\ & \quad l^{\text{cat}} \in \mathcal{L}^{\text{cat}}(x^{\text{m}}), \\ & \quad x^{\text{qnt}} \in \mathcal{X}^{\text{qnt}}(x^{\text{m}}) \end{array} \right\} \tag{37}$$

where $l^{\text{cat}} = \phi^{\text{cat}}_{(k)}(x^{\text{cat}};x^{\text{m}})$ and $\mathcal{L}^{\text{cat}}(x^{\text{m}})$ is the encoded parametrized categorical set.

Definition 15 allows to set $\mathcal{L}^{\text{cat}}(x^{\text{m}}) = \mathcal{X}^{\text{cat}}(x^{\text{m}})$, so that no encoding is done: $l^{\text{cat}} = x^{\text{cat}}$. In addition, since some categorical kernels do not require encoding, it follows that the auxiliary domain $\mathcal{X}_{\text{aux}}$ is compatible with encoded categorical variables or with the original categorical variables.

From Definition 15, the auxiliary maximization problem may be formulated as:

$$\begin{array}{lll} (P^{\text{aux}}) & \max\limits_{x \in \mathcal{X}_{\text{aux}}} & \alpha\left(x;\tilde{f}\right) \\ & \text{s.t.} & \tilde{c}_i(x) \leq 0, \qquad \forall i \in \{1, 2, \dots, p\} \\ & & \tilde{c}^{\text{m}}(x) \leq 0, \qquad \forall \tilde{c}^{\text{m}} \in \tilde{C}^{\text{m}}(x^{\text{m}}), \\ & & x^{\text{cat}} = \phi^{\text{cat}}_{(k)}(x^{\text{cat}};x^{\text{m}}) \text{ for some } x^{\text{cat}} \in \mathcal{X}^{\text{cat}}(x^{\text{m}}), \end{array} \tag{38}$$

where (P^{aux}) stands for auxiliary problem, $\alpha : \mathcal{X}_{\text{aux}} \to \mathbb{R}$ is an acquisition function applied to a surrogate model $\tilde{f}$, $\tilde{c}_i$ $\forall i \in \{1, 2, \dots, p\}$ are surrogate constraints for the neutral constraints, $\tilde{c}^{\text{m}} \in \tilde{C}^{\text{m}}$ is a surrogate constraint for a decreed constraint. The last constraint imposes the existence of some $x^{\text{cat}} \in \mathcal{X}^{\text{cat}}(x^{\text{m}})$ such that $x^{\text{cat}} = \phi^{\text{cat}}_{(k)}(x^{\text{cat}};x^{\text{m}})$ is a pre-image constraint that recovers a categorical component $x^{\text{cat}} \in \mathcal{X}^{\text{cat}}(x^{\text{m}})$ from the encoded parametrized categorical set $\mathcal{L}^{\text{cat}}(x^{\text{m}})$. The pre-image constraint also ensures that the optimal auxiliary problem solution resides in the domain $\mathcal{X}$. For more details on pre-images problem, refer to [26].

4.2.2 Bayesian Optimization

In this section, the BO approach is formulated as an auxiliary problem (P^{aux}), without detailing the algorithmic steps or the construction of the GP (see [19] or [32] for more details on this subject). For the purpose of this work, it is sufficient to formulate the BO approach as an auxiliary problem (P^{aux}) and to develop the kernel from the notation framework, since the kernel almost entirely characterizes the probabilistic surrogate (GP). A kernel $k : \mathcal{X}_{\text{aux}} \times \mathcal{X}_{\text{aux}} \to \mathbb{R}$ is a positive semi-definite covariance function. Conceptually, the kernel establishes the mathematical properties of the GP, such as the degree smoothness.

In its simplest noise free form, a probabilistic BO distribution is built from a GP, which allows to compute for any given point $x \in \mathcal{X}_{\text{aux}}$, a prediction $\hat{f}(x)$ and an uncertainty measure $\hat{\sigma}^2(x)$, such that

$$\begin{cases} \hat{f}(x) & = \kappa^{\intercal}(x) K^{-1} f(\mathbb{X}) \\ \hat{\sigma}(x)^2 & = k(x,x) - \kappa^{\intercal}(x) K^{-1} \kappa(x) \end{cases} \tag{39}$$

where $\mathbb{X}$ is a set of sample points, $f(\mathbb{X})$ is the vector of objective function values of the sample points, $\kappa(x)$ is a vector in which an element is the computed kernel $k(x, y)$ with $(x, y) \in \mathcal{X}_{\text{aux}} \times \mathbb{X}$, K is a matrix in containing all pairs $(y, z) \in \mathbb{X} \times \mathbb{X}$, such that an element of K is $k(y, z)$. In Eq. (39), everything is computed from the kernel k. In other words, Eq. (39) displays that the GP is entirely characterized by the kernel: it is assumed that the GP is noise free and that the mean function is zero, which is a common practice [19]. The surrogate probabilistic model $\tilde{f}$ satisfies

$$\tilde{f}(x) \sim \mathcal{N}\big(\hat{f}(x), \hat{\sigma}(x)^2\big). \tag{40}$$

where $\mathcal{N}$ is the normal distribution. Moreover, a common acquisition function α applied on GP surrogates is the *EI* from [20]:

$$EI\big(x;\tilde{f}\big) = \mathbb{E}\big[\max(f_\star - \tilde{f}(x), 0)\big] = \big(f_\star - \hat{f}(x)\big)\Phi\left(\frac{f_\star - \hat{f}(x)}{\hat{\sigma}(x)}\right) + \hat{\sigma}(x)\phi\left(\frac{f_\star - \hat{f}(x)}{\hat{\sigma}(x)}\right) \tag{41}$$

where $f_\star = f(x_k)$ is current best known objective function value at iteration $k > 1$, $\hat{\sigma}(x)$ is the standard deviation of the GP, Φ and ϕ are respectively the cumulative distribution and the density function of a standard normal distribution (centered at zero with variance of one). In Eq. (41), the intensification and exploration trade-off of the *EI* (acquisition function) is displayed by the two terms: the first term favors promising low surrogate values (intensification) and the second term favors highly uncertain points (exploration). In the auxiliary problem (P^{aux}), the acquisition function could be $\alpha(x;\tilde{f}) = EI(x;\tilde{f})$.

In a similar manner to surrogate model $\tilde{f}$ evaluated at a point $x \in \mathcal{X}_{\text{aux}}$ in Eq. (40), the surrogate constraints in the auxiliary problem (P^{aux}), may be developed into GP probabilistic surrogates. Thus, a given surrogate constraint $\tilde{c}_i$ would have its own prediction function $\hat{c}_i$ (similarly to $\hat{f}(x)$ in Eq. (39)), which could be directly used in the auxiliary problem (P^{aux}), *i.e.*, $\tilde{c}_i(x) = \hat{c}_i(x)$. Acquisition functions may also be applied to probabilistic surrogate constraints, which is not covered in this work.

At this stage, the BO framework is formulated in a general manner, which does not explicit the mixed-nature of the optimization problems at stake. To adapt the BO framework on a mixed-variable context, the kernel k, must be further detailed with the support of the notation framework. Many possible kernels can be built with operations of multiplication and additions that respects the RKHS formalism [22, 23]. An example of a specific kernel is detailed next to illustrate the compatibility of the framework with the mixed-variable optimization BO literature.

The kernel k is built piece-by-piece with the partition of a point $x = (x^{\text{m}}$, $x = (x^{\text{m}}, x^{\text{nom}}, x^{\text{ord}}, x^{\text{int}}, x^{\text{con}})$. The parametrized continuous kernel $k^{\text{con}} : \mathcal{X}^{\text{con}}(x^{\text{m}}) \times \mathcal{X}^{\text{con}}(x^{\text{m}}) \rightarrow \mathbb{R}$ is formulated as multiplication of one-dimensional squared-exponential kernels:

$$k^{\text{con}}(x^{\text{con}}, y^{\text{con}}; x^{\text{m}}) = \exp\left(-\sum_{i=1}^{n^{\text{con}}(x^{\text{m}})} \lambda_i^{\text{con}} \left[x_i^{\text{con}} - y_i^{\text{con}}\right]^2\right). \tag{42}$$

where the λ_i^{con} are weight coefficients (hyperparameters of the surrogate model) that can be adjusted by various methods, such as the MLE.

The parametrized integer kernel $k^{\text{int}} : \mathcal{X}^{\text{int}}(x^{\text{m}}) \times \mathcal{X}^{\text{int}}(x^{\text{m}}) \rightarrow \mathbb{R}$ is similar to k^{con}, but applies a transformation T that rounds the relaxed integer variables to the nearest integer [21]:

$$k^{\text{int}}(x^{\text{int}},y^{\text{int}};x^{\text{m}})=\exp\left(-\sum_{i=1}^{n^{\text{int}}(x^{\text{m}})}\lambda_i^{\text{int}}\left[T\left(x_i^{\text{int}}\right)-T\left(y_i^{\text{int}}\right)\right]^2\right) \tag{43}$$

where $x_i^{\text{int}}, y_i^{\text{int}}\ \forall i \in I^{\text{int}}(x^{\text{m}})$ are relaxed integer variables and λ_i^{int} are hyperparameters of the surrogate model. The transformation T conserves the order of an integer variable and ensures that the one-dimensional kernels in (43) are piecewise functions [21].

The parametrized quantitative kernel $k^{\text{qnt}} : \mathcal{X}^{\text{qnt}}(x^{\text{m}}) \times \mathcal{X}^{\text{qnt}}(x^{\text{m}}) \to \mathbb{R}$ is formulated as multiplication of k^{int} and k^{con}:

$$k^{\text{qnt}}(x^{\text{qnt}},y^{\text{qnt}};x^{\text{m}})=k^{\text{int}}(x^{\text{int}},y^{\text{qnt}};x^{\text{m}})\cdot k^{\text{con}}(x^{\text{qnt}},y^{\text{qnt}};x^{\text{m}}). \tag{44}$$

The parametrized categorical kernel k^{cat} may be formulated with an encoding on the categorical variables [25] ($k^{\text{cat}} : \mathcal{L}^{\text{cat}}(x^{\text{m}}) \times \mathcal{L}^{\text{cat}}(x^{\text{m}}) \to \mathbb{R}$), or without any encoding ($k^{\text{cat}} : \mathcal{X}^{\text{cat}}(x^{\text{m}}) \times \mathcal{X}^{\text{cat}}(x^{\text{m}}) \to \mathbb{R}$). With an encoding, the parametrized categorical kernel k^{cat} is similar to k^{con}:

$$k^{\text{cat}}(l^{\text{cat}},u^{\text{cat}};x^{\text{m}})=\exp\left(-\sum_{i\in\mathcal{L}_{\text{aux}}^{\text{cat}}(x^{\text{m}})}\lambda_i^{\text{cat}}\left[l_i^{\text{cat}}-y_i^{\text{cat}}\right]^2\right), \tag{45}$$

where λ_i^{cat} are hyperparameters of the surrogate model and $\mathcal{L}_{\text{aux}}^{\text{cat}}(x^{\text{m}})$ is the set of indices of the encoded (included) categorical variables. Without any encoding, the parametrized categorical kernel k^{cat} is formulated as tensor products of matrices (one matrix per categorical variable) [22]:

$$k^{\text{cat}}(x^{\text{cat}},y^{\text{cat}};x^{\text{m}})=\left(\otimes_{i=1}^{n^{\text{nom}}(x^{\text{m}})}T_i^{\text{nom}}\left(x_i^{\text{nom}},y_i^{\text{nom}}\right)\right)\otimes\left(\otimes_{i=j}^{n^{\text{ord}}(x^{\text{m}})}T_j^{\text{ord}}\left(x_j^{\text{ord}},y_j^{\text{ord}}\right)\right), \tag{46}$$

where, for $t \in \{\text{nom}, \text{ord}\}$ and a categorical variable $x_i^t \in \{1, 2, \dots, c_i\}$, $T_i^t \in \mathbb{R}^{c_i \times c_i}$ is a positive semi-definite matrix in which an element is the correlation between two categories of the variable x_i^t. Hence, for two given variables with specific categories $x_i^t = c_1$ and $y_i^t = c_2$, $T_i^t\left(x_i^t, y_i^t\right)$ is a correlation measure between the categories c_1 and c_2. In Eq. (46), the matrices for the nominal and ordinal variables T_i^{nom} and T_j^{ord} are distinguished, since there exist more sophisticated matrices for the ordinal variables [22].

Finally, a mixed kernel $k : \mathcal{X}_{\text{aux}} \times \mathcal{X}_{\text{aux}} \to \mathbb{R}$, based on [23], is formulated as:

$$k(x,y)=\begin{cases}\displaystyle\prod_{i=1}^{n^{\text{m}}} k_i^{\text{m}}(x_i^{\text{m}},y_i^{\text{m}}), & \text{if } x^{\text{m}} \neq y^{\text{m}},\\ \displaystyle\prod_{i=1}^{n^{\text{m}}}\left(k_i^{\text{m}}(x_i^{\text{m}},y_i^{\text{m}})+\left[k^{\text{cat}}(l^{\text{cat}},u^{\text{cat}};x^{\text{m}})k^{\text{qnt}}(x^{\text{qnt}},y^{\text{qnt}};x^{\text{m}})\right]\right), & \text{otherwise,}\end{cases} \tag{47}$$

where $k_i^{\text{m}} : S_i^{\text{m}} \times S_i^{\text{m}} \to \mathbb{R}$ is a one-dimensional kernel for a meta variable $x_i^{\text{m}} \in S_i^{\text{m}}$, k^{cat} is the parametrized categorical kernel that may take the form in Eqs. (45) or (46),

k^{qnt} is the parametrized quantitative kernel in Eq. (44). In Eq. (47), the meta kernel $k^{\mathrm{m}} : \mathcal{X}^{\mathrm{m}} \times \mathcal{X}^{\mathrm{m}} \to \mathbb{R}$ is implicitly decomposed into one-dimensional kernels (one per meta variable), which is again, common practice in the literature. Moreover, in Eq. (47), the kernel computations for the categorical and quantitative variables are only being done if the two points in share the same meta component: for $t \in \{\mathrm{cat}, \mathrm{qnt}\}$, the kernel computation $k(x^t, y^t; x^{\mathrm{m}})$ in Eq. (47) is only done if $x^{\mathrm{m}} = y^{\mathrm{m}}$, which implies that x^t and y^t must both reside in the same parametrized set $\mathcal{X}^{\mathrm{t}}(x^{\mathrm{m}})$.

5 Conclusion

To the best of the authors' knowledge, no previous work has explicitly and formally defined the domain $\mathcal{X}$ of the objective function nor the feasible set Ω for the class of problems of interest: this is mainly what justified the need of a formal mathematical framework to model this class of problem. This work proposes a thorough notation framework for mixed-variable optimization problems. The framework formally and properly models mixed-variable problems with a careful emphasis on meta and categorical variables.

Many definitions are developed to shed light on the intrinsic difficulties resulting from the presence of the new meta variables. The roles of variables, which encompasses decreed, neutral and meta variables, and the decree property establish a novel and systematic approach to model such optimization problems.

The general constrained optimization problem (1) is explicitly and formally formulated, for the class of problems, throughout the new definitions of a point $x = (x^{\mathrm{m}}, x^{\mathrm{cat}}, x^{\mathrm{qnt}})$, the domain $\mathcal{X}$ of the objective function, and the feasible set Ω. Moreover, for $t \in \{\mathrm{cat}, \mathrm{nom}, \mathrm{ord}, \mathrm{qnt}, \mathrm{int}, \mathrm{con}\}$, a parametrized set $\mathcal{X}^t(x^{\mathrm{m}})$ elucidates that some variables of type t may be included or excluded of the problem depending on the meta variables. The parametrized categorical set $\mathcal{X}^{\mathrm{cat}}(x^{\mathrm{m}})$ and the parametrized quantitative set $\mathcal{X}^{\mathrm{qnt}}(x^{\mathrm{m}})$ are building blocks of the domain $\mathcal{X}$ that has two equivalent formulations. Both formulations provide a different perspective on mixed-variable problems. Furthermore, the constraints are split into neutral and decreed constraints, which allow to formulate the feasible set Ω.

In Section 4, a direct search approach and Bayesian optimization are presented using the proposed framework. More precisely, in Section 4.1, the subproblems strategy is introduced. The objective subfunction g and the quantitative subproblems strategy are explicitly formulated, which leads to the formal definition of an user-defined neighborhood mapping $\mathcal{N}^t : \mathcal{X} \to \mathcal{X}^t$ (exploration of subproblems) for direct search methods. Subsequently, the encoder $\phi^t_{(k)}$, the auxiliary domain and the auxiliary maximization problem are all introduced and defined in Section 4.2. From these new definitions, Bayesian optimization is formalized within the framework, notably from a mixed kernel $k : \mathcal{X}^{\mathrm{aux}} \times \mathcal{X}^{\mathrm{aux}} \to \mathbb{R}$ that is constructed with the framework.

Thereby, the notation framework is shown to be compatible with the two main approaches of the literature on mixed-variable optimization with meta and categorical variables in a blackbox optimization context. The mathematical framework has been carefully developed to be compatible with both approaches.

In general, direct search methods are exclusively developed by blackbox optimization researchers, whereas Bayesian optimization methods exclusively by machine learning researchers. Thus, one of the intentions of this work is to bridge blackbox optimization and machine learning specifically for mixed-variable blackbox optimization problems.

Computational experiments will be carried out in future studies with the mathematical framework of this work as a foundation.

Funding This study was funded by the Natural Sciences and Engineering Research Council of Canada (NSERC Alliance grant 544900-19 in collaboration with Huawei-Canada), an IVADO (Institut de valorisation des données) MSc Excellence Scholarship and a Hydro-Québec MSc Excellence Scholarship.

Data Availability Statement Not relevant for the present theoretical study.

Declarations

Conflict of Interest Statement On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Audet C, Hare W (2017) Derivative-Free and Blackbox Optimization. Springer Series in Operations Research and Financial Engineering. Springer, Cham, Switzerland
2. Alarie S, Audet C, Gheribi AE, Kokkolaras M, Le Digabel S (2021) Two decades of blackbox optimization applications. EURO Journal on Computational Optimization 9:100011
3. Choi TD, Eslinger OJ, Kelley CT, David JW, Etheridge M (2000) Optimization of automotive valve train components with implicit filtering. Optim Eng 1(1):9–27
4. Xu J, Audet C, DiLiberti CE, Hauck WW, Montague TH, Parr AF, Potvin D, Schuirmann DJ (2016) Optimal adaptive sequential designs for crossover bioequivalence studies. Pharm Stat 15(1):15–27
5. Audet C, Orban D (2006) Finding optimal algorithmic parameters using derivative-free optimization. SIAM J Optim 17(3):642–664
6. Marsden AL, Wang M, Dennis JE Jr, Moin P (2007) Trailing-edge noise reduction using derivative-free optimization and large-eddy simulation. J Fluid Mech 572:13–36
7. Abramson MA (2004) Mixed Variable Optimization of a Load-Bearing Thermal Insulation System Using a Filter Pattern Search Algorithm. Optim Eng 5(2):157–177
8. Kokkolaras M, Audet C, Dennis JE Jr (2001) Mixed variable optimization of the Number and composition of heat intercepts in a thermal insulation system. Optim Eng 2(1):5–29
9. Lucidi S, Piccialli V, Sciandrone M (2005) An Algorithm Model for Mixed Variable Programming. SIAM J Optim 15(4):1057–1084
10. Lakhmiri D, Le Digabel S, Tribes C (2021) HyperNOMAD: Hyperparameter Optimization of Deep Neural Networks Using Mesh Adaptive Direct Search. ACM Trans Math Softw 47(3)
11. Audet C, Dennis JE Jr (2001) Pattern Search Algorithms for Mixed Variable Programming. SIAM J Optim 11(3):573–594
12. Abramson MA, Audet C, Dennis JE Jr (2007) Filter pattern search algorithms for mixed variable constrained optimization problems. Pacific J Optim 3(3):477–500
13. Abramson MA, Audet C, Chrissis JW, Walston JG (2009) Mesh Adaptive Direct Search Algorithms for Mixed Variable Optimization. Optim Lett 3(1):35–47
14. Audet C, Dennis JE Jr (2006) Mesh Adaptive Direct Search Algorithms for Constrained Optimization. SIAM J Optim 17(1):188–217

15. Audet C, Le Digabel S, Tribes C (2019) The Mesh Adaptive Direct Search Algorithm for Granular and Discrete Variables. SIAM J Optim 29(2):1164–1189
16. Lucidi S, Piccialli V (2004) A Derivative-Based Algorithm for a Particular Class of Mixed Variable Optimization Problems. Optim Methods Softw 17(3–4):317–387
17. Nannicini G (2021) On the implementation of a global optimization method for mixed-variable problems. Open Journal of Mathematical Optimization 2:1–25
18. Regis RG, Shoemaker CA (2007) A stochastic radial basis function method for the global optimization of expensive functions. INFORMS J Comput 19:497–509
19. Rasmussen CE, Williams CKI (2006) Gaussian Processes for Machine Learning. The MIT Press
20. Jones DR, Schonlau M, Welch WJ (1998) Efficient Global Optimization of Expensive Black Box Functions. J Glob Optim 13(4):455–492
21. Garrido-Merchán EC, Hernández-Lobato D (2020) Dealing with categorical and integer-valued variables in Bayesian Optimization with Gaussian processes. Neurocomputing 380:20–35
22. Roustant O, Padonou E, Deville Y, Clément A, Perrin G, Giorla J, Wynn H (2020) Group kernels for Gaussian process metamodels with categorical inputs. Uncertainty Quantification 8(2):775–806
23. Pelamatti J, Brevault L, Balesdent M, Talbi E-G, Guerin Y (2021) Bayesian optimization of variable-size design space problems. Optim Eng 22:387–447
24. MunozZuniga M, Sinoquet D (2020) Global optimization for mixed categorical-continuous variables based on Gaussian process models with a randomized categorical space exploration step. INFOR: Information Systems and Operational Research 58(2):310–341
25. Zhang Y, Tao S, Chen W, Apley DW (2020) A Latent Variable Approach to Gaussian Process Modeling with Qualitative and Quantitative Factors. Technometrics 62(3):291–302
26. Cuesta-Ramirez J, Le Riche R, Roustant O, Perrin G, Durantin C, Gliere A (2022) A comparison of mixed-variables Bayesian optimization approaches. Advanced Modeling and Simulation in Engineering Sciences 9(1):6
27. Booker AJ, Dennis JE Jr, Frank PD, Serafini DB, Torczon V, Trosset MW (1999) A Rigorous Framework for Optimization of Expensive Functions by Surrogates. Struct Multidiscip Optim 17(1):1–13
28. Goodfellow I, Bengio Y, Courville A (2016) Deep Learning. MIT Press
29. Hastie T, Tibshirani R, Friedman J (2001) The Elements of Statistical Learning. Springer Series in Statistics. Springer New York Inc., New York, NY, USA
30. Audet C, Tribes C (2018) Mesh-based Nelder-Mead algorithm for inequality constrained optimization. Comput Optim Appl 71(2):331–352
31. Audet C, Dennis JE Jr (2009) A Progressive Barrier for Derivative-Free Nonlinear Programming. SIAM J Optim 20(1):445–472
32. Shahriari B, Swersky K, Wang Z, Adams RP, De Freitas N (2015) Taking the human out of the loop: A review of Bayesian optimization. Proc IEEE 104(1):148–175

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