

# Surrogate-based categorical neighborhoods for mixed-variable blackbox optimization

C. Audet, Y. Diouane, E. Hallé-Hannan, S. Le Digabel, C. Tribes

G–2026–16

March 2026

---

La collection *Les Cahiers du GERAD* est constituée des travaux de recherche menés par nos membres. La plupart de ces documents de travail a été soumis à des revues avec comité de révision. Lorsqu'un document est accepté et publié, le pdf original est retiré si c'est nécessaire et un lien vers l'article publié est ajouté.

**Citation suggérée :** C. Audet, Y. Diouane, E. Hallé-Hannan, S. Le Digabel, C. Tribes (Mars 2026). Surrogate-based categorical neighborhoods for mixed-variable blackbox optimization, Rapport technique, Les Cahiers du GERAD G– 2026–16, GERAD, HEC Montréal, Canada.

**Avant de citer ce rapport technique**, veuillez visiter notre site Web (<https://www.gerad.ca/fr/papers/G-2026-16>) afin de mettre à jour vos données de référence, s'il a été publié dans une revue scientifique.

---

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2026  
– Bibliothèque et Archives Canada, 2026

The series *Les Cahiers du GERAD* consists of working papers carried out by our members. Most of these pre-prints have been submitted to peer-reviewed journals. When accepted and published, if necessary, the original pdf is removed and a link to the published article is added.

**Suggested citation:** C. Audet, Y. Diouane, E. Hallé-Hannan, S. Le Digabel, C. Tribes (March 2026). Surrogate-based categorical neighborhoods for mixed-variable blackbox optimization, Technical report, Les Cahiers du GERAD G–2026–16, GERAD, HEC Montréal, Canada.

**Before citing this technical report**, please visit our website (<https://www.gerad.ca/en/papers/G-2026-16>) to update your reference data, if it has been published in a scientific journal.

---

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2026  
– Library and Archives Canada, 2026

# Surrogate-based categorical neighborhoods for mixed-variable blackbox optimization

**Charles Audet**  
**Youssef Diouane**  
**Edward Hallé-Hannan**  
**Sébastien Le Digabel**  
**Christophe Tribes**

*Département de mathématiques et de génie industriel, Polytechnique Montréal & GERAD, Montréal, (Qc), Canada, H3T 1J4*

charles.audet@gerad.ca  
youssef.diouane@polymtl.ca  
edward.halle-hannan@polymtl.ca  
sebastien.le-digabel@polymtl.ca  
christophe.tribes@polymtl.ca

**March 2026**  
**Les Cahiers du GERAD**  
**G–2026–16**

Copyright © 2026 Audet, Diouane, Hallé-Hannan, Le Digabel, Tribes

---

Les textes publiés dans la série des rapports de recherche *Les Cahiers du GERAD* n'engagent que la responsabilité de leurs auteurs. Les auteurs conservent leur droit d'auteur et leurs droits moraux sur leurs publications et les utilisateurs s'engagent à reconnaître et respecter les exigences légales associées à ces droits. Ainsi, les utilisateurs:

- Peuvent télécharger et imprimer une copie de toute publication du portail public aux fins d'étude ou de recherche privée;
- Ne peuvent pas distribuer le matériel ou l'utiliser pour une activité à but lucratif ou pour un gain commercial;
- Peuvent distribuer gratuitement l'URL identifiant la publication.

Si vous pensez que ce document enfreint le droit d'auteur, contactez-nous en fournissant des détails. Nous supprimerons immédiatement l'accès au travail et enquêterons sur votre demande.

The authors are exclusively responsible for the content of their research papers published in the series *Les Cahiers du GERAD*. Copyright and moral rights for the publications are retained by the authors and the users must commit themselves to recognize and abide the legal requirements associated with these rights. Thus, users:

- May download and print one copy of any publication from the public portal for the purpose of private study or research;
- May not further distribute the material or use it for any profit-making activity or commercial gain;
- May freely distribute the URL identifying the publication.

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

**Abstract :** In simulation-based engineering, design choices are often obtained following the optimization of complex blackbox models. These models frequently involve mixed-variable domains with quantitative and categorical variables. Unlike quantitative variables, categorical variables lack an inherent structure, which makes them difficult to handle, especially in the presence of constraints. This work proposes a systematic approach to structure and model categorical variables in constrained mixed-variable blackbox optimization. Surrogate models of the objective and constraint functions are used to induce problem-specific categorical distances. From these distances, surrogate-based neighborhoods are constructed using notions of dominance from bi-objective optimization, jointly accounting for information from both the objective and the constraint functions. This study addresses the lack of automatic and constraint-aware categorical neighborhood construction in mixed-variable blackbox optimization. As a proof of concept, these neighborhoods are employed within **CatMADS**, an extension of the MADS algorithm for categorical variables. The surrogate models are Gaussian processes, and the resulting method is called **CatMADS<sub>GP</sub>**. The method is benchmarked on the **Cat-Suite** collection of 60 mixed-variable optimization problems and compared against state-of-the-art solvers. Data profiles indicate that **CatMADS<sub>GP</sub>** achieve superior performance for both unconstrained and constrained problems.

**Keywords :** Blackbox optimization; derivative-free optimization; mixed-variable problems; constrained optimization; categorical variables

---

**Acknowledgements:** We gratefully acknowledge Dr. Paul Saves for his help with the implementation of SMT and the kernel.

This research is funded by a Natural Sciences and Engineering Research Council of Canada (NSERC) PhD Excellence Scholarship (PGS D), a Fonds de Recherche du Québec (FRQNT) PhD Excellence Scholarship and an Institut de l'Énergie Trottier (IET) PhD Excellence Scholarship, as well as by the NSERC discovery grants RGPIN-2020-04448 (Audet), RGPIN-2024-05093 (Diouane) and RGPIN-2024-05086 (Le Digabel).

**Data availability statement:** Scripts and data are publicly available at [https://github.com/bbopt/surrogate\\_based\\_neighborhoods](https://github.com/bbopt/surrogate_based_neighborhoods).

**Conflict of interest statement** The authors state that there are no conflicts of interest.

# 1 Introduction

In most cases, design choices in engineering and machine learning imply the constrained optimization of complex computer simulations that are *expensive-to-evaluate* blackboxes with no analytical expressions. In addition, these blackboxes may involve different types of variables, such as continuous, integer and categorical variables. The latter are qualitative and lack structure. For example, in aerospace engineering, optimal aircraft design involves simulations with choices of materials or propulsion systems [12, 32]. Deep learning models are characterized by hyperparameters, such as the activation function or the choice of optimizer, whose values affect the training and validation pipeline [7, 16, 22]. Standard optimization methods using gradients and relaxations in the presence blackboxes and categorical variables are not suited for this context.

The motivation of this work is to improve the optimization of categorical variables in a constrained blackbox settings where information is limited and constraints may be difficult to handle. This is done by proposing a systematic approach to structure categorical variables with neighborhoods constructed using surrogate models of the objective and constraint functions. These neighborhoods can be directly incorporated into mixed-variable methods using categorical neighborhoods. As a proof-of-concept, they are employed with CatMADS [6], a mixed-variable extension of the Mesh Adaptive Direct Search (MADS) algorithm [4] for constrained blackbox optimization.

## Notation

The notation used follows [6]. Vectors are in bold, and scalars are in normal font. Superscripts are reserved for types of variables. Subscripts without parentheses index variables, *e.g.*,  $x_1^{\text{cat}}$  is the first categorical variable. Subscripts with parentheses are reserved for indexing an iteration  $k$ , *e.g.*  $\mathbf{x}_{(k)}$ , or for listing points, *e.g.*  $\mathbf{x}_{(1)}, \mathbf{x}_{(2)}, \dots, \mathbf{x}_{(s)}$ . Sets are denoted using capital letters with either normal or calligraphic font, *e.g.*  $A$  or  $\mathcal{A}$ , except for the set of known points noted  $\mathbb{X}$ .

### 1.1 Constrained mixed-variable blackbox optimization

This study tackles inequality constrained mixed-variable blackbox optimization problems formulated as

$$\min_{\mathbf{x} \in \Omega} f(\mathbf{x}), \quad (1)$$

where  $f : \mathcal{X} \rightarrow \overline{\mathbb{R}}$  is the objective function with  $\overline{\mathbb{R}} = \mathbb{R} \cup \{+\infty\}$ ,  $\mathcal{X}$  is the domain of the objective and constraint functions,  $\mathbf{x} \in \mathcal{X}$  is a point,  $\Omega := \{\mathbf{x} \in \mathcal{X} : g_j(\mathbf{x}) \leq 0, j \in J\}$  is the feasible set defined by the constraints,  $g_j : \mathcal{X} \rightarrow \overline{\mathbb{R}}$  is the  $j$ -th constraint function of the problem with  $j \in J$  and  $|J| \in \mathbb{N}$  is the number of constraints.

The constraints defining the feasible set  $\Omega$  are supposed to be *relaxable* and *quantifiable*, meaning that when they are evaluated, they can return a proper value without being satisfied [23]. An *unrelaxable* constraint, which must be satisfied to have a proper blackbox execution, is considered out of the domain, such that if  $\mathbf{x} \notin \mathcal{X}$ , then it is assigned  $f(\mathbf{x}) = +\infty$ . Similarly, if a point  $\mathbf{x} \in \mathcal{X}$  hits a *hidden constraint* that crashes or invalidates the output values, then it is simply assigned  $f(\mathbf{x}) = +\infty$ .

The functions involved in the main optimization problem are supposed to be *blackboxes*, which are defined as follows in [8]: “*any process that when provided an input, returns an output, but the inner working of the process is not analytically available*”. For example, the strain experienced by a beam can be estimated using a finite element simulation that outputs strain values for a given geometry, material and loading conditions. The lack of an analytical expression implies that derivatives are unavailable with respect to the continuous variables. The function evaluations are determined by executing a process that is possibly time-consuming.

The problems considered are said to be mixed-variable. Each variable is either *continuous* (cont), *integer* (int) or *categorical* (cat). The variables of the same type are contained in a corresponding

column vector called a *component*. For  $t \in \{\text{cat}, \text{int}, \text{cont}\}$ , the component of type  $t$  is expressed as

$$\mathbf{x}^t := (x_1^t, x_2^t, \dots, x_{n^t}^t) \in \mathcal{X}^t := \mathcal{X}_1^t \times \mathcal{X}_2^t \times \dots \times \mathcal{X}_{n^t}^t, \quad (2)$$

where  $x_i^t \in \mathcal{X}_i^t$  represents the  $i$ -th variable of type  $t$ , and  $\mathcal{X}_i^t$  denotes its bounds. The variable index  $i$  ranges over the set  $I^t := \{1, 2, \dots, n^t\}$ . The number of variables of type  $t$  is noted  $n^t \in \mathbb{N}$ . The set of type  $t$  is noted  $\mathcal{X}^t$  and it is such that  $\mathbf{x}^t \in \mathcal{X}^t$ .

Following the components, a point  $\mathbf{x} \in \mathcal{X}$  is defined by a partition by types, and the domain is expressed as Cartesian products of the sets by types, such that

$$\mathbf{x} := (\mathbf{x}^{\text{cat}}, \mathbf{x}^{\text{int}}, \mathbf{x}^{\text{cont}}) \in \mathcal{X} := \mathcal{X}^{\text{cat}} \times \mathcal{X}^{\text{int}} \times \mathcal{X}^{\text{cont}}. \quad (3)$$

The categorical variables take *categories* representing qualitative values. For  $i \in I^{\text{cat}}$ , the  $i$ -th categorical variable  $x_i^{\text{cat}}$  takes values in the set  $\mathcal{X}_i^{\text{cat}} := \{c_1, c_2, \dots, c_{\ell_i}\}$ , where  $c_j$  is a category with  $j \in \{1, 2, \dots, \ell_i\}$  and  $\ell_i$  is the number of categories of the variable. For example, the color for an object could be taking a category in the set  $\{\text{RED}, \text{BLUE}, \text{GREEN}\}$ . These variables are inherently difficult to treat because their sets are not ordered and usual distance functions are not suitable [7, 19]. The categorical set  $\mathcal{X}^{\text{cat}}$  is assumed to be finite. The total number of categorical components is  $|\mathcal{X}^{\text{cat}}| = \prod_{i=1}^{n^{\text{cat}}} \ell_i$ .

The integer and continuous variables take quantitative values and they belong to ordered sets in which standard distance functions are well-defined. These variables are typically much easier to optimize than the categorical ones. For convenience, the integer and continuous variables may be regrouped in a quantitative component such that  $\mathbf{x}^{\text{qnt}} := (\mathbf{x}^{\text{int}}, \mathbf{x}^{\text{cont}}) \in \mathcal{X}^{\text{qnt}} := \mathcal{X}^{\text{int}} \times \mathcal{X}^{\text{cont}}$ .

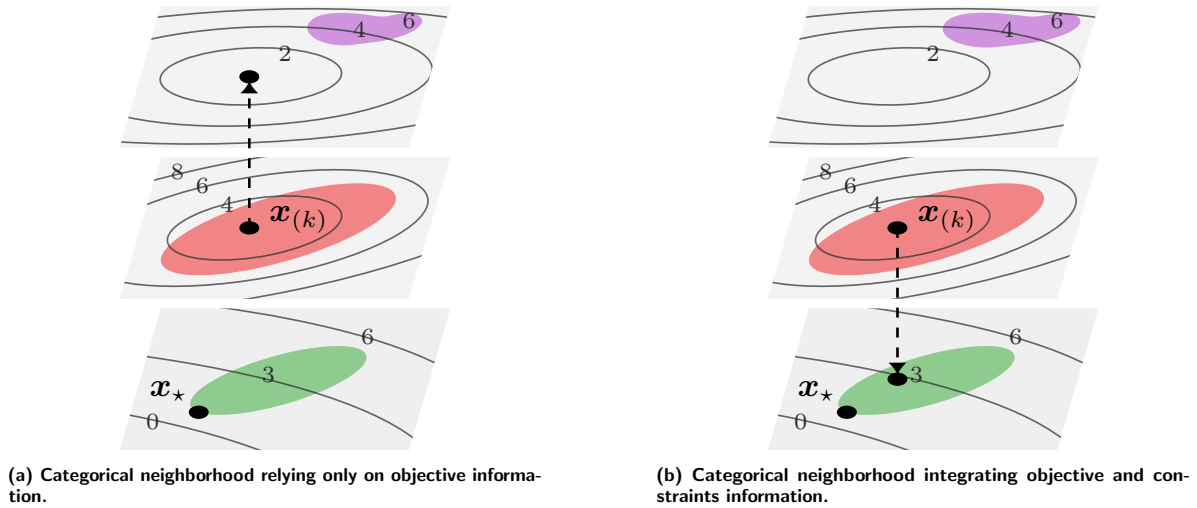
Note that this study does not consider *meta variables*, which influence the inclusion or bounds of other variables [7, 16]. The problems have fixed dimensions and bounds.

## 1.2 Research gap, objectives and contributions

Bayesian optimization (BO) frameworks have been successfully applied to mixed-variable blackbox problems [27, 29, 33, 37]. However, BO does not provide local optimization mechanisms, convergence guarantees, or principled stopping criteria. Several metaheuristics rely on operations based on Gower-type distances that reduce categorical differences to mismatches between categories. Such generic measures are not problem-informed and can lead to a large number of unnecessary evaluations.

There are other mixed-variable methods, such as CatMADS and MV-MADS, that handle categorical variables with neighborhoods [1, 6, 24]. In these methods, neighborhoods are used to generate candidate solutions by modifying the variables of an incumbent. The main advantage of these neighborhoods is that they introduce a local mechanism that leads to theoretical guarantees for discrete variables. The performance of neighborhood-based approaches depends highly on the quality of the neighborhoods. In MV-MADS, neighborhoods are constructed with user-defined rules, which can be difficult to establish in a blackbox context. In practice, these neighborhoods are primarily intended to handle discrete variables, but they may also modify the continuous variables of an incumbent. The theoretical results of MV-MADS assume that the points in the user-defined neighborhoods belong to the feasible set  $\Omega$ . This constitutes an important methodological restriction for handling constraints and does not capitalize on information from relaxable constraints. CatMADS focuses instead on neighborhoods strictly for categorical variables and constructs them using categorical distances defined solely from the objective function: The constraint functions are thus not considered. This can be problematic for problems in which the behavior of the constraint functions changes drastically with respect to the categorical variables.

Figure 1 illustrates a simple example with one categorical variable  $x^{\text{cat}} \in \{\text{PURPLE}, \text{RED}, \text{GREEN}\}$  and two continuous variables. Each plane corresponds to a category, and the colored regions denote feasibility. Each plane also shows some level curves of the objective function.



**Figure 1:** A motivating example where the incumbent  $x_{(k)}$  has objective value  $f(x_{(k)}) = 3$ , while the global minimum  $x_*$  has  $f(x_*) = 1$ . The neighborhood on the left selects PURPLE BASED STRICTLY ON PROXIMITY WITH THE OBJECTIVE. THE NEIGHBORHOOD ON THE RIGHT SELECTS GREEN BASED ON PROXIMITY OF BOTH THE CONSTRAINTS AND THE OBJECTIVE.

In Figure 1, the current solution  $x_{(k)}$  lies in the continuous subspace associated with the categorical component  $x^{\text{cat}} = \text{RED}$ . It corresponds to a local minimum with respect to the continuous variables. However, the global minimum  $x_*$  resides in the continuous subspace associated with  $x^{\text{cat}} = \text{GREEN}$ . In the setting, reaching this global minimum from  $x_{(k)}$  can be difficult when using a neighborhood-based approach. From  $x_{(k)}$  in the subspace of RED, objective-based or user-defined neighborhoods would likely prioritize the category PURPLE. The restriction of the objective function in the RED and PURPLE subspaces appears more similar. Consequently, an objective-based neighborhood may focus on the PURPLE category, although: 1) its associated continuous subspace is highly infeasible and yields poorer feasible values, and 2) the feasible regions in the continuous subspaces of RED and GREEN are much more similar. This can degrade the performance of neighborhood-based methods and the quality of the local minima with respect to the categorical variables. To mitigate this issue, one could rely on global mechanisms such as BO or design better-informed neighborhoods. In the presence of noise or nonsmoothness, BO may fail to properly tackle the continuous variables and by consequence the mixed-variable problem itself. More local and robust mechanisms may therefore be needed to ensure incremental progress.

This motivates the development of local neighborhoods that are better-informed and robust. The research gap addressed is the automatic construction of problem-specific categorical neighborhoods that incorporate information from both the objective and the constraint functions in a blackbox setting. Note that, hereafter, categorical neighborhoods are referred to simply as neighborhoods. The work does not aim to introduce a new optimization method, but rather to provide additional structural information that facilitates handling categorical variables. These better-informed neighborhoods provide a systematic structure for categorical variables, improve interpretability, and are particularly well suited for blackbox constrained optimization. They enhance the quality of local minima with respect to categorical variables and improve the efficiency neighborhood-based methods. The number of evaluations for optimizing categorical variables may decrease, since each categorical neighbor is selected in a more informed manner.

The present work is structured through four objectives, representing the main contributions of the work. The first objective is to construct a problem-specific categorical distance using kernel-based surrogate models, such as Gaussian Processes (GPs), so that categorical components with similar predicted objective values are considered closer. The second objective is to define a categorical distance that captures similarity with respect to the constraint functions by aggregating surrogate models

of the constraints. Building on these two distances, the third objective is to construct categorical neighborhoods balancing similarity between the objective and the constraints. The first three objectives are developed independently of any specific optimization framework and for purely categorical domains. The fourth objective is to generalize the proposed neighborhoods to mixed-variable domains. As a proof of concept for the fourth objective, the neighborhoods are used in **CatMADS**.

### 1.3 Related work

In [6], categorical distance is constructed using a basic interpolation of the objective function. Neighborhoods are characterized with this single distance function and they are constructed without accounting the constraint functions, which may lead to neighborhoods containing elements that are highly dissimilar in terms of feasibility. In [1, 24], categorical neighborhoods are specified manually by the user, which requires prior knowledge about the problem structure, which may be ambiguous in a blackbox context. The primary focus of **CatMADS** and **MV-MADS** lies in the optimization process, while neighborhood design remains secondary. In contrast, the present work emphasizes the systematic construction of categorical neighborhoods, independently of any particular optimization framework.

Several metaheuristic methods have been proposed to address categorical variables in mixed-variable optimization. These methods construct candidate points with operations transforming categorical variables of existing solutions. Such operations have similar roles to categorical neighborhoods. **CatCMA** extends the covariance matrix adaptation evolution strategy (CMA-ES) to problems involving both categorical and continuous variables [18]. In this approach, categorical variables are modeled by probability vectors over their respective categories, while continuous variables are modeled by a multivariate normal distribution. A joint probabilistic distribution over categorical and continuous variables is introduced to guide the optimization in mixed-variable domains. The Non-dominated Sorting Genetic Algorithm-II (NSGA-2) [13], although originally developed for multi-objective optimization, has also been extended to mixed-variable problems involving categorical variables. [10] provides implementations with crossover and mutation operators tailored to each variable type, from which candidate points are generated over a population of solutions. Candidate points are constructed with these operators applied to a population of solutions. For more details, the survey [36] provides an exhaustive survey for mixed-variable metaheuristics.

In Bayesian optimization, categorical variables are structured with similarity measures, called *kernels*. In optimization, kernels are problem-specific similarity measures characterizing surrogate models that guide the optimization. In the literature, various techniques exist for constructing categorical kernels, most notably *one-hot encoding* and matrix-based approaches. One-hot encoding assigns a unique binary variable to each category, and they are assigned their corresponding hyperparameter weighting the importance of each category in the kernel [14]. Matrix-based approaches have shown great results in BO [27, 29, 31, 33]. The elements of these matrices are hyperparameters that model correlations and anti-correlations between different categories [29]. The similarities between categories are directly encoded in matrices. Other techniques for encoding categorical variables include continuous latent spaces. In [37], each categorical variable is assigned a 2D continuous space, in which its categories are mapped in a way that correlated categories are close, and uncorrelated ones are far. The coordinates of the categories can be viewed as hyperparameters representing correlations.

The rest of this document is organized as follows. An illustrative optimization problem is introduced in [Section 2](#) to guide the presentation. The essential concepts of kernels and GPs are presented in [Section 3](#). In [Section 4](#), the categorical distances and neighborhoods are developed. The novel neighborhoods are adapted and used within the **CatMADS** framework in [Section 5](#) to provide proof of concept. Lastly, some numerical experiments benchmarking **CatMADS** using the new neighborhoods with other solvers are presented in [Section 6](#).

## 2 An illustrative mechanical-part design problem

To outline the proposed contributions, a mechanical-part design problem is used as an illustrative example in the following sections. The goal is to minimize the physical strain of a structural piece subject to a budget constraint and an ecological score constraint. A point represents a design, and it is composed of three categorical variables and two continuous variables. The categorical variables correspond to the choice of supplier  $u \in \{A, B\}$ , the material  $a \in \{\text{ALUM}, \text{STEEL}, \text{COMP}^1, \text{WOOD}\}$ , and the shape  $s \in \{\text{SQUARE}, \text{CIRCLE}, \text{ELLIPSE}\}$ . The continuous variables are a length  $l \in [5, 10]$  and a ratio of recycle material  $r \in [0, 1]$ . The domain and a point are defined as

$$\mathbf{x} = (u, a, s, l, r) \in \mathcal{X} = \mathcal{X}^{\text{cat}} \times \mathcal{X}^{\text{cont}} \quad (4)$$

where  $\mathcal{X}^{\text{cat}} = \{A, B\} \times \{\text{ALUM}, \text{STEEL}, \text{COMP}, \text{WOOD}\} \times \{\text{SQUARE}, \text{CIRCLE}, \text{ELLIPSE}\}$ , and  $\mathcal{X}^{\text{cont}} = [5, 10] \times [0, 1]$ . The categorical set  $\mathcal{X}^{\text{cat}}$  contains  $2 \times 4 \times 3 = 24$  categorical components, and the feasible region  $\Omega$  is unknown.

The problem is intentionally constructed as a blackbox in accordance with the scope of this work. The functions are evaluated through numerical iterative routines to a prescribed accuracy. By design, the problem features strong interactions between the categorical variables and constraint feasibility. Each categorical component  $\mathbf{x}^{\text{cat}} = (u, a, s) \in \mathcal{X}^{\text{cat}}$  induces a different feasible region over the continuous subspace  $\mathcal{X}^{\text{cont}} = [5, 10] \times [0, 1]$ . Some categories may appear intuitively similar, *e.g.*, the circle and ellipse categories for the cross-section shape. However, the objective function, and, in particular, the constraint functions, behave very differently across categories. As a result, constructing categorical neighborhoods without incorporating information from both the objective and the constraints can lead to misleading similarity measures and poor optimization performance. The difficulty of the problem is outlined in Figure 2 that describes three Latin Hypercube Sampling (LHS) experiments and a graph.

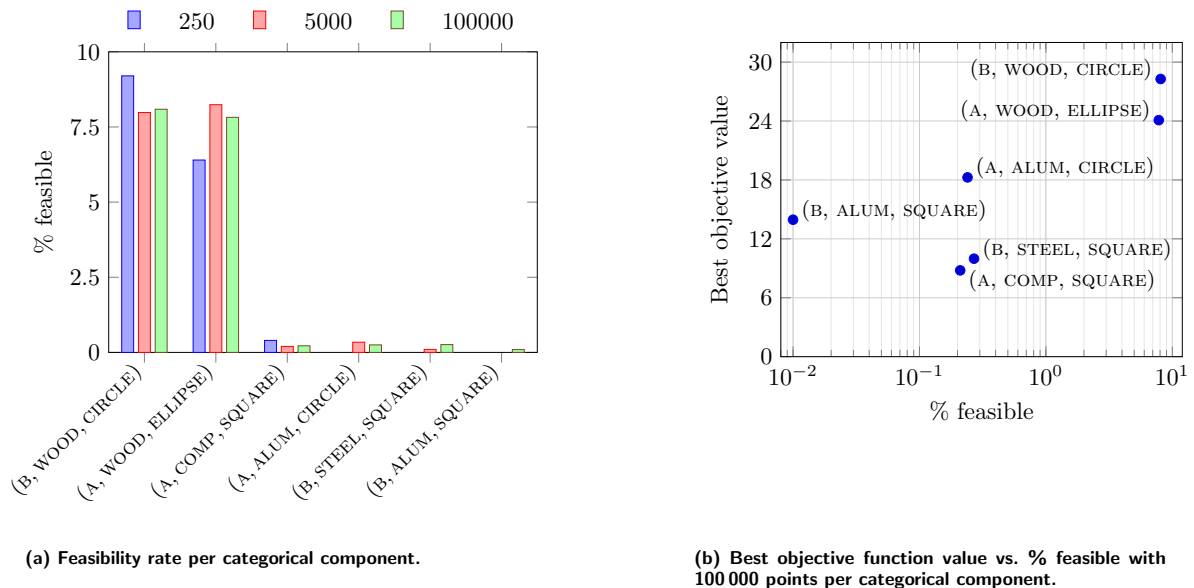


Figure 2: Statistical analysis of feasibility per categorical component in the piece machining problem.

In the histogram of Figure 2a, LHS is performed independently within each categorical component. Each bar represents the percentage of feasible points obtained for a given number of samples. Only the categorical component for which at least one feasible point was sampled is presented. As the number of points per categorical component increases, more components with feasible points are discovered.

<sup>1</sup>COMP for “composite”

The categorical component (A, ALUM, CIRCLE) does not have any feasible point with 250 samples (there is no blue bin), but it does with 5 000 and 100 000 samples. The histogram suggests that only six categorical components may be capable of achieving feasibility.

The graph in [Figure 2b](#) shows the best feasible objective value obtained by performing an LHS with 100 000 points for each categorical component. Overall, best objective values are associated with smaller percentages of feasibility.

### 3 Mixed-variable kernels

This study uses similarity measures derived from available data to induce categorical distances including kernel-based measures. A similarity measure quantifies how close or related two data points are in the decision space. A kernel is a particular type of similarity measure. The data points of Problem (1) lie in the domain  $\mathcal{X}$ , and the kernel is defined on this domain. From such a kernel, a categorical kernel is extracted and used to construct categorical neighborhoods. In practice, kernels can compare mixed-variable points, enabling the construction of interpolation models that rely exclusively on similarity measures, such as mixed-variable Gaussian processes.

Formally, a kernel  $\kappa : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$  is a symmetric similarity measure, such that

1.  $\kappa(\mathbf{x}, \mathbf{y}) = \kappa(\mathbf{y}, \mathbf{x})$  for any  $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ , and
2. for any finite set of points  $\{\mathbf{x}_{(1)}, \mathbf{x}_{(2)}, \dots, \mathbf{x}_{(p)}\}$  with  $p \in \mathbb{N}$ , the symmetric  $p \times p$  matrix  $[K]_{i,j} = \kappa(\mathbf{x}_{(i)}, \mathbf{x}_{(j)})$  for  $i, j \in \{1, 2, \dots, p\}$  is positive semi-definite.

Such kernel  $\kappa$  is said to be positive semi-definite. The kernel is constructed following the approach of the classic textbook [30]. For  $t \in \{\text{cat}, \text{qnt}\}$  and  $i \in I^t$ , each variable  $x_i^t \in \mathcal{X}_i^t$  is assigned a one-dimensional kernel  $\kappa_i^t : \mathcal{X}_i^t \times \mathcal{X}_i^t \rightarrow \mathbb{R}$  of the appropriate variable type. The one-dimensional kernels are combined with multiplications that conserve symmetry and semi-positive definiteness.

Common kernels for quantitative variables include polynomial, Matern or Gaussian kernels, *e.g.*, see [30]. Typically, the quantitative kernel can be built directly as product of one-dimensional Gaussian kernels, *i.e.*,

$$\kappa^{\text{qnt}}(\mathbf{x}^{\text{qnt}}, \mathbf{y}^{\text{qnt}}) := \prod_{i=1}^{n^{\text{qnt}}} \exp\left(-\theta_i^{\text{qnt}} \left(x_i^{\text{qnt}} - y_i^{\text{qnt}}\right)^2\right), \quad (5)$$

where  $\boldsymbol{\theta}^{\text{qnt}} := (\theta_1^{\text{qnt}}, \theta_2^{\text{qnt}}, \dots, \theta_{n^{\text{qnt}}}^{\text{qnt}}) \in \mathbb{R}_+^{n^{\text{qnt}}}$  is the vector of hyperparameters of the quantitative kernel. The quantitative kernel and the following ones are all parameterized by hyperparameters. For readability, hyperparameters are not explicitly noted in the arguments of kernels.

#### 3.1 Categorical kernels

As mentioned in [Section 1.3](#), there are currently two main state-of-the-art approaches for constructing categorical kernels: one-hot encoding and matrix-based methods. Again, the matrix-based approach encodes similarity measures between categories directly into matrices. More precisely, each categorical variable  $x_i^{\text{cat}} \in \mathcal{X}_i^{\text{cat}}$  is assigned a positive semi-definite matrix  $T_i \in \mathbb{R}^{\ell_i^{\text{cat}} \times \ell_i^{\text{cat}}}$  where  $\ell_i^{\text{cat}}$  is the number of categories of  $x_i^{\text{cat}}$ . A matrix  $T_i^{\text{cat}} := L_i L_i^\top$  is built as a symmetric and positive semi-definite matrix where  $L_i$  is a lower-triangular matrix parameterized by a hypersphere decomposition (lengths and angles) [11, 20]. An element of a matrix  $L_i$  contains hyperparameters of the categorical kernel, whereas an element of a matrix  $T_i^{\text{cat}}$  is a correlation measure between two categories of the corresponding categorical variable. For example, in the machining piece problem, the similarity matrix associated



corresponding images  $\mathbf{f} = (f(\mathbf{x}_{(1)}), f(\mathbf{x}_{(2)}), \dots, f(\mathbf{x}_{(p)}))$ . This procedure yields a prediction function  $\hat{f} : \mathcal{X} \rightarrow \mathbb{R}$  and a variance function  $\hat{\sigma}^2 : \mathcal{X} \rightarrow \mathbb{R}_+$  that, in the noiseless setting, can be expressed as in [30]:

$$\begin{aligned}\hat{f}(\mathbf{x}) &:= \boldsymbol{\kappa}(\mathbf{x})^\top K^{-1} \mathbf{f}, \\ \hat{\sigma}^2(\mathbf{x}) &:= \boldsymbol{\kappa}(\mathbf{x}, \mathbf{x}) - \boldsymbol{\kappa}(\mathbf{x})^\top K^{-1} \boldsymbol{\kappa}(\mathbf{x}) \geq 0,\end{aligned}\tag{9}$$

where  $\boldsymbol{\kappa} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$  is the kernel,  $\boldsymbol{\kappa}(\mathbf{x}) := (\boldsymbol{\kappa}(\mathbf{x}, \mathbf{x}_{(1)}), \boldsymbol{\kappa}(\mathbf{x}, \mathbf{x}_{(2)}), \dots, \boldsymbol{\kappa}(\mathbf{x}, \mathbf{x}_{(p)})) \in \mathbb{R}^p$  is the kernel vector comparing an input point  $\mathbf{x} \in \mathcal{X}$  and the points in  $\mathbb{X}$ , and  $K \in \mathbb{R}^{p \times p}$  is the kernel matrix that compares all points in  $\mathbb{X}$  such that  $[K]_{i,j} := \boldsymbol{\kappa}(\mathbf{x}_{(i)}, \mathbf{x}_{(j)})$  for  $i, j \in \{1, 2, \dots, p\}$ .

The kernel characterizes the GP by quantifying how correlated or similar input points are, *i.e.*, how close their output should be. Concretely, it controls the general behavior of the fit, *e.g.*, the smoothness in a continuous setting. In the mixed-variable setting, defining an appropriate mixed-variable kernel is key to constructing a GP surrogate that can model the functions of interest. In fact, the GP that best fits the available data is found by adjusting the hyperparameters of the kernel. They are typically obtained by maximizing the marginal likelihood of the GP [30]:

$$\boldsymbol{\theta}_* \in \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \log \mathbb{P}[\mathbf{f} \mid \mathbb{X}, \boldsymbol{\theta}] \quad \text{where} \quad \log \mathbb{P}[\mathbf{f} \mid \mathbb{X}, \boldsymbol{\theta}] = -\frac{1}{2} \mathbf{f}^\top K^{-1} \mathbf{f} - \frac{1}{2} \log \det(K) - \frac{p}{2} \log(2\pi). \tag{10}$$

The marginal likelihood is continuously differentiable. Hence, it can be optimized with standard continuous solvers. After this step, the GP is properly constructed and the categorical kernel is finely adjusted to the data problem. The categorical kernel is used in the next section.

The GP surrogates of the constraint functions are constructed using the same procedure as for the objective function. Specifically, for  $j \in J$ , each constraint function  $g_j$  is provided a probabilistic surrogate  $\tilde{g}_j$  with a predictive mean function  $\hat{g}_j : \mathcal{X} \rightarrow \mathbb{R}$  and variance function  $\hat{\sigma}_j^2 : \mathcal{X} \rightarrow \mathbb{R}_+$ , as well as its own kernel and hyperparameters.

## 4 Surrogate-based categorical neighborhoods

Kernels and surrogate models are used to construct neighborhoods that structure categorical variables. The proposed neighborhoods are defined using two categorical distances, one associated with the objective function and another one associated with the constraint functions. Proximity between categorical variables simultaneously considers similarities in the objective and constraint functions.

This section focuses exclusively on categorical variables. Quantitative variables are temporarily omitted to simplify the presentation. The surrogate models  $\tilde{f}$  and  $\tilde{g}$ , as well as the kernel, are defined on the categorical set  $\mathcal{X}^{\text{cat}}$ . A neighborhood constructed at an incumbent  $\mathbf{u} \in \mathcal{X}^{\text{cat}}$  is denoted  $\mathcal{N}(\mathbf{u}; m)$  where  $m \in \{1, 2, \dots, |\mathcal{X}^{\text{cat}}|\}$  is the number of neighbors. This notation is formalized at the end of the section.

Section 4.1 presents categorical distances that establish proximity with respect to the objective function. Afterwards, Section 4.2 proposes a categorical distance for the constraint functions. Based on these categorical distances, surrogate-based neighborhoods are developed in Section 4.3.

### 4.1 Categorical distance for the objective function

Kernels are equivalent to scalar products in Hilbert spaces [25]. This property allows categorical kernels to be transformed into categorical distances. Let  $\boldsymbol{\phi} : \mathcal{X}^{\text{cat}} \rightarrow \mathcal{H}$  be a mapping that maps the categorical components of the categorical set  $\mathcal{X}^{\text{cat}}$  into a Hilbert space  $\mathcal{H}$  equipped with a scalar product. Then, the categorical kernel is uniquely defined according to the Moore-Aronszajn Theorem [3] and satisfies  $\boldsymbol{\kappa}^{\text{cat}}(\mathbf{u}, \mathbf{v}) = \boldsymbol{\phi}(\mathbf{u})^\top \boldsymbol{\phi}(\mathbf{v})$ . Note that the categorical kernel  $\boldsymbol{\kappa}^{\text{cat}}$  is still computed as described in

Section 3, *i.e.*, with Gaussian kernels or matrices. The categorical distance  $d_f : \mathcal{X}^{\text{cat}} \times \mathcal{X}^{\text{cat}} \rightarrow \mathbb{R}_+$  is defined with the categorical kernel [15], and it is ensured to be a metric via the implicit mapping  $\phi$ :

$$\begin{aligned} d_f(\mathbf{u}, \mathbf{v}) &:= \kappa^{\text{cat}}(\mathbf{u}, \mathbf{u}) + \kappa^{\text{cat}}(\mathbf{v}, \mathbf{v}) - 2\kappa^{\text{cat}}(\mathbf{u}, \mathbf{v}) \\ &= \phi(\mathbf{u})^\top \phi(\mathbf{u}) + \phi(\mathbf{v})^\top \phi(\mathbf{v}) - 2\phi(\mathbf{u})^\top \phi(\mathbf{v}) \\ &= \|\phi(\mathbf{u}) - \phi(\mathbf{v})\|^2. \end{aligned} \tag{11}$$

Although the mapping  $\phi$  and the Hilbert space  $\mathcal{H}$  are not explicitly known, (11) shows that the categorical distance can be equivalently interpreted in a Hilbert space with  $d_f(\mathbf{u}, \mathbf{v}) = \|\phi(\mathbf{u}) - \phi(\mathbf{v})\|^2$ . Without constraint functions, categorical neighborhoods are characterized only by the categorical distance. Hence, unconstrained neighborhoods can be viewed as being implicitly constructed in a Hilbert space, where two categorical components  $\mathbf{u}, \mathbf{v} \in \mathcal{X}^{\text{cat}}$  that are more similar correspond to a smaller distance  $\|\phi(\mathbf{u}) - \phi(\mathbf{v})\|^2$ . Figure 3 illustrates this Hilbert space as a two-dimensional continuous space, where categorical components are mapped into continuous vectors. Note that this is only an equivalent representation. In practice, the distance is computed as  $d_f(\mathbf{u}, \mathbf{v}) = \kappa^{\text{cat}}(\mathbf{u}, \mathbf{u}) + \kappa^{\text{cat}}(\mathbf{v}, \mathbf{v}) - 2\kappa^{\text{cat}}(\mathbf{u}, \mathbf{v})$ .

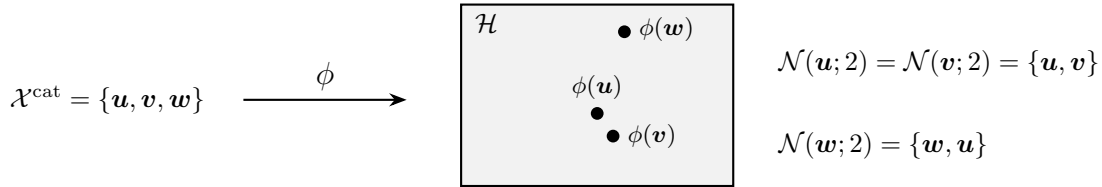


Figure 3: Visualization of a unconstrained neighborhood equivalently constructed in a Hilbert space.

In Figure 3, the categorical components are mapped in a Hilbert space via an implicit mapping  $\phi$ . Schematically, neighborhoods containing two elements can be viewed as follows: the neighborhoods constructed at  $\mathbf{u}$  and  $\mathbf{v}$  coincide and contain the components  $\{\mathbf{u}, \mathbf{v}\}$ , while the neighborhood constructed at  $\mathbf{w}$  contains  $\{\mathbf{w}, \mathbf{u}\}$ .

On the one hand, the construction of the categorical distance  $d_f$  requires non-negligible computational effort, notably because of the maximization of the marginal likelihood of the GP in (10). On the other hand, it is induced by a finely-tuned kernel. The categorical distance in (11) is constructed from the problem data through the kernel, so that correlated categorical components are embedded closer to each other in a Hilbert space, while uncorrelated components are placed farther apart. This behavior is particularly useful for constructing categorical neighborhoods. This distance is finer than binary categorical distances, such as the Hamming distance, which treats all mismatches equally and does not use information from the problem. As an example, let  $\mathcal{X}^{\text{cat}} = \{\text{R}, \text{B}, \text{G}\}$  and  $d_{\text{HM}}$  denote the Hamming distance. Then  $d_{\text{HM}}(\text{R}, \text{G}) = d_{\text{HM}}(\text{R}, \text{B}) = 1$ , implying that it is not possible to determine whether green or blue is closer to red.

If a surrogate model is available, but no kernel is defined, the categorical pseudo-distance can be approximated by a pseudo-distance using prediction values, such that  $d_f(\mathbf{u}, \mathbf{v}) \approx \left| \hat{f}(\mathbf{u}) - \hat{f}(\mathbf{v}) \right|$ . This pseudo-distance trivially satisfies symmetry and triangular inequality properties, but not the identity of indiscernibles property since  $d_f(\mathbf{u}, \mathbf{v}) = 0 \not\Rightarrow \mathbf{u} = \mathbf{v}$  when  $\hat{f}$  is not injective. Hence, categorical components with similar prediction values are considered close. Conceptually, this mimics the behavior of a kernel inducing an interpolation model, such as GPs. Although simple, this pseudo-distance captures proximity with respect to the objective function  $f$ . That said, when a kernel with adjusted hyperparameters is available, the categorical distance of (11) is generally preferable, as it corresponds to measuring similarity in an inner-product Hilbert space.

## 4.2 Categorical pseudo-distance for constraint functions

As discussed previously, the main goal is to construct neighborhoods that incorporate information from both the objective and constraint functions. This section develops a pseudo-distance that establishes proximity with respect to the constraint functions.

When a neighborhood is constructed at a feasible component, it should first maintain feasibility and then promote improvement in the objective function, *i.e.*, similarity with respect to the objective. In this setting, differences in the degree of feasibility between categorical components that are both predicted to be feasible are of limited importance. From an optimization perspective, once feasibility is predicted, the focus should instead be on improving the objective function.

In particular, large distances between a “highly feasible” component and a “marginally feasible” component should be avoided, as both satisfy the constraints and should be compared primarily regarding the objective. For this reason, the proposed pseudo-distance is constructed so that it evaluates to zero when both compared components are predicted to be feasible.

The next functions are used to construct the pseudo-distance. For each  $j \in J$ , a function  $\hat{g}_j^+ : \mathcal{X}^{\text{cat}} \rightarrow \mathbb{R}$  is defined from the predictive model  $\hat{g}_j : \mathcal{X}^{\text{cat}} \rightarrow \mathbb{R}$  and its associated uncertainty model  $\hat{\sigma}_j : \mathcal{X}^{\text{cat}} \rightarrow \mathbb{R}_+$ , both derived from the surrogate model  $\tilde{g}_j$  of the  $j$ -th constraint. Let  $\lambda \geq 0$  be a relaxation parameter used to relax a predicted constraint value  $\hat{g}_j(\mathbf{u})$  using its associated uncertainty  $\hat{\sigma}_j(\mathbf{u}) \geq 0$ . The function  $\hat{g}_j^+$  sets to zero any predicted constraint value  $\hat{g}_j(\mathbf{u})$  that is feasible after relaxation by  $\lambda\hat{\sigma}_j(\mathbf{u})$  and normalizes the remaining values, such that

$$\hat{g}_j^+(\mathbf{u}) := \begin{cases} 0 & \text{if } \hat{g}_j(\mathbf{u}) - \lambda\hat{\sigma}_j(\mathbf{u}) \leq 0, \\ \psi(\hat{g}_j(\mathbf{u})) & \text{otherwise,} \end{cases} \quad (12)$$

where  $\psi$  is a normalization function ensuring that  $\hat{g}_j^+(\mathbf{u}) \in [0, 1]$  holds for any  $\mathbf{u} \in \mathcal{X}^{\text{cat}}$ . For example,  $\psi$  may apply a min-max normalization on the bounds of  $\hat{g}_j(\mathbf{u})$  when  $\hat{g}_j(\mathbf{u}) - \lambda\hat{\sigma}_j(\mathbf{u}) > 0$  is considered. Note that if the surrogate model does not provide an uncertainty measure, then  $\lambda$  can be set to zero, meaning that no uncertainty is taken into account. For readability, (12) may be expressed in a vector form  $\hat{\mathbf{g}}^+ : \mathcal{X}^{\text{cat}} \rightarrow \mathbb{R}_+^{|J|}$ , considering all indices  $j \in J$  compactly.

The pseudo-distance function associated with the constraint functions is constructed from the vector function  $\hat{\mathbf{g}}^+$ , as follows :

$$d_{\mathbf{g}}(\mathbf{u}, \mathbf{v}) := \|\hat{\mathbf{g}}^+(\mathbf{u}) - \hat{\mathbf{g}}^+(\mathbf{v})\|_p, \quad (13)$$

where  $p \geq 1$  and  $d_{\mathbf{g}}(\mathbf{u}, \mathbf{v}) = 0$  are ensured when both arguments are predicted to be feasible after relaxation, *i.e.*, when  $\hat{\mathbf{g}}(\mathbf{u}) - \lambda\hat{\boldsymbol{\sigma}}(\mathbf{u}) \leq \mathbf{0}$  and  $\hat{\mathbf{g}}(\mathbf{v}) - \lambda\hat{\boldsymbol{\sigma}}(\mathbf{v}) \leq \mathbf{0}$ . The normalization function  $\psi$  ensures that the pseudo-distance  $d_{\mathbf{g}}$  is not biased by the relative scales of the constraint function values. The symmetry and triangular inequality properties are trivially satisfied by virtue of the  $p$ -norm. However, the identity of indiscernibles is not respected, *i.e.*,  $d_{\mathbf{g}}(\mathbf{u}, \mathbf{v}) = 0 \not\Rightarrow \mathbf{u} = \mathbf{v}$  since the vector function  $\hat{\mathbf{g}}^+$  is not injective by design. This last unsatisfied property makes  $\hat{\mathbf{g}}^+$  a pseudo-distance function instead of a distance function.

## 4.3 Surrogate-based neighborhoods

The novel categorical neighborhoods are developed in this section. A neighborhood is constructed at a component  $\mathbf{u} \in \mathcal{X}^{\text{cat}}$ . The other components in the neighborhood are selected based on trade-offs between distances concerning the objective and the constraint functions. This selection is formalized using notions of dominance inspired by bi-objective optimization. In the following, two ranking functions are derived from the distance functions introduced in previous sections: a primary and a secondary ranking function, both parameterized by a component  $\mathbf{u} \in \mathcal{X}^{\text{cat}}$  at which the neighborhood is constructed. Conceptually, the first selected components correspond to Pareto-optimal solutions

concerning the two ranking functions, while subsequent components prioritize the primary ranking function over the secondary one.

The primary and secondary ranking functions are defined in two cases depending on the component  $\mathbf{u} \in \mathcal{X}^{\text{cat}}$ . If  $\mathbf{u}$  is feasible, then the primary ranking function is derived from  $d_g$  (constraints), since the priority is to construct neighborhoods with components that maintain feasibility. Otherwise,  $\mathbf{u}$  is infeasible, and the primary ranking function is based on  $d_f$  (objective), allowing greater flexibility with respect to feasibility. This may allow finding promising solutions that are difficult to reach when imposing feasibility at all times.

The primary and secondary ranking functions, noted  $p_u : \mathcal{X}^{\text{cat}} \rightarrow \mathbb{R}_+$  and  $s_u : \mathcal{X}^{\text{cat}} \rightarrow \mathbb{R}$  respectively, are parameterized by  $\mathbf{u} \in \mathcal{X}^{\text{cat}}$  and defined as follows

$$\begin{aligned} p_u(\mathbf{v}) &:= d_g(\mathbf{u}, \mathbf{v}) \quad \text{and} \quad s_u(\mathbf{v}) := d_f(\mathbf{u}, \mathbf{v}) \quad \text{if } \mathbf{u} \in \Omega, \\ p_u(\mathbf{v}) &:= d_f(\mathbf{u}, \mathbf{v}) \quad \text{and} \quad s_u(\mathbf{v}) := d_g(\mathbf{u}, \mathbf{v}) \quad \text{if } \mathbf{u} \notin \Omega. \end{aligned} \tag{14}$$

An illustrative example is presented before introducing the formal ordering rules that determine the components selected in a neighborhood. These rules are induced by the ranking functions. In this example, there are twelve categorical components. Hence, eleven components must be ordered, since the neighborhood stems from  $\mathbf{u} \in \mathcal{X}^{\text{cat}}$ , which is naturally ordered first. Visually, the ordering of the components is performed in the space of images  $(s_u, p_u)$  across three steps, each corresponding to a sub-figure from left to right in Figure 4.

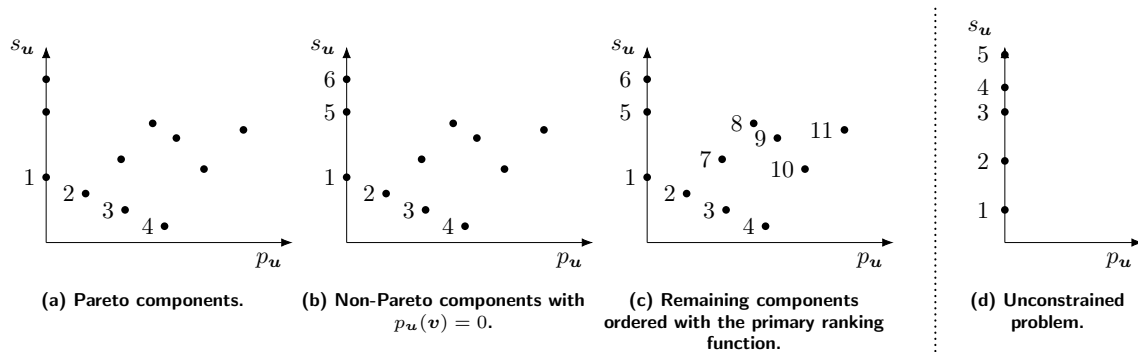


Figure 4: Ordering components with ranking functions. (a)-(c) three steps for constrained problems. (d) single step for unconstrained problems. The component  $\mathbf{u}$  is located at the origin and is not displayed.

The first four components are the Pareto solutions, *i.e.*, the nondominated solutions, in the space of images. The Pareto components are ordered among themselves solely with the primary ranking function  $p_u$ . The components 5 and 6 are not Pareto, but they have a primary ranking function value of zero with the component  $\mathbf{u}$  at which the neighborhood is constructed. This situation may most notably occur when the primary ranking function is based on the pseudo-distance  $d_g$ : recall that when both components are predicted to be feasible, this pseudo-distance returns zero. Among themselves, components 5 and 6 are ordered using the secondary ranking function. Finally, the remaining components 7 to 11 are directly ordered with the primary ranking function.

Note that when the problem is unconstrained, the component  $\mathbf{u} \in \Omega$  and, consequently, the primary ranking function is based on  $d_g$ . However, in this case, the pseudo-distance function  $d_g$  is always set to zero by convention, since its components are always feasible. Therefore, only the distance function of the objective  $d_f$  is considered. An unconstrained example is illustrated with six components (including  $\mathbf{u}$  that is not displayed) in Figure 4d, where the ordering is done directly based on their height.

The relation establishing a partial order on the categorical set  $\mathcal{X}^{\text{cat}}$ , given a component  $\mathbf{u} \in \mathcal{X}^{\text{cat}}$  and ranking functions, is now formally introduced.

**Definition 1** (Ordering relation). Given a component  $\mathbf{u} \in \mathcal{X}^{\text{cat}}$ , and ranking functions  $p_{\mathbf{u}} : \mathcal{X}^{\text{cat}} \rightarrow \mathbb{R}_+$  and  $s_{\mathbf{u}} : \mathcal{X}^{\text{cat}} \rightarrow \mathbb{R}_+$ , the ordering relation  $\preceq$  defined on the categorical set  $\mathcal{X}^{\text{cat}}$  is such that for any pair  $\mathbf{v}, \mathbf{w} \in \mathcal{X}^{\text{cat}}$ ,  $\mathbf{v}$  is ordered before or tied to  $\mathbf{w}$ , noted  $\mathbf{v} \preceq \mathbf{w}$ , when any of the following conditions hold:

1.  $p_{\mathbf{u}}(\mathbf{v}) = p_{\mathbf{u}}(\mathbf{w}) = 0$  and  $s_{\mathbf{u}}(\mathbf{v}) \leq s_{\mathbf{u}}(\mathbf{w})$ ,
2.  $p_{\mathbf{u}}(\mathbf{v}) > 0, p_{\mathbf{u}}(\mathbf{w}) > 0$  and  $p_{\mathbf{u}}(\mathbf{v}) \leq p_{\mathbf{u}}(\mathbf{w})$ ,
3.  $\mathbf{v}$  is not dominated by any component of  $\mathcal{X}^{\text{cat}} \setminus \{\mathbf{u}\}$  in the space of images  $(p_{\mathbf{u}}, s_{\mathbf{u}})$ , but  $\mathbf{w}$  is dominated by at least one such component,
4.  $p_{\mathbf{u}}(\mathbf{v}) \leq p_{\mathbf{u}}(\mathbf{w})$ .

The ordering relation in [Definition 1](#) does not establish a total order, because there can be ties, notably when  $\mathbf{v}$  and  $\mathbf{w}$  are such that  $p_{\mathbf{u}}(\mathbf{v}) = p_{\mathbf{u}}(\mathbf{w})$  and  $s_{\mathbf{u}}(\mathbf{v}) = s_{\mathbf{u}}(\mathbf{w})$ . Nevertheless, it provides a systematic way to identify promising categorical components with information from both the objective and constraint functions. If the problem is unconstrained, then only the first condition is used since all components are feasible, as illustrated in [Figure 4d](#).

Now that the ordering relation is formally defined, the neighborhoods based on surrogate models are finally introduced.

**Definition 2** (Surrogate-based neighborhood). For a given component  $\mathbf{u} \in \mathcal{X}^{\text{cat}}$  and integer  $m \in \{1, 2, \dots, |\mathcal{X}^{\text{cat}}|\}$ , the *surrogate-based neighborhood*  $\mathcal{N}(\mathbf{u}; m) \subseteq \mathcal{X}^{\text{cat}}$  is the set containing  $m$  categorical components of lowest ordering with respect to an ordering relation  $\preceq$  defined on  $\mathcal{X}^{\text{cat}}$ . The relation  $\mathbf{v} \preceq \mathbf{w}$  is satisfied whenever  $\mathbf{v} \in \mathcal{N}(\mathbf{u}; m)$  and  $\mathbf{w} \notin \mathcal{N}(\mathbf{u}; m)$ .

[Definition 2](#) generalizes distance-induced neighborhoods in [\[6\]](#) by using relations instead of a single distance for selecting components. In fact, when a problem is unconstrained, a surrogate-based neighborhood is equivalent to a distance-induced neighborhood using  $d_f$ . That said, the implementation in [\[6\]](#) uses a basic mixed-variable interpolation to construct the categorical distance. The kernel-induced categorical distance  $d_f$  introduced in [Section 4.1](#) provides a more structured similarity measure, as it is induced by an inner product in a Hilbert space through well-defined categorical kernels. The improvements does not only concerns constrained problems.

#### 4.4 Illustrative case study of the mechanical-part problem

The mechanical-part design problem from [Section 2](#) is used here to develop a case study with categorical neighborhoods. A single Latin Hypercube Sampling (LHS) of 96 points is performed on the mixed-variable domain described in [Section 2](#), in which three points are generated for each of the 32 categorical components. Only one point sample is feasible, that is,  $\mathbf{x}_* = (\text{B}, \text{WOOD}, \text{CIRCLE}, 5.5, 1)$  with objective function value  $f(\mathbf{x}_*) = 28.55$  is known. In study, the goal is to find another triplet of categorical variables maintaining feasibility and improving the objective value without modifying the continuous variables. The continuous variables are fixed, and the current categorical component is noted  $\mathbf{u} = (\text{B}, \text{WOOD}, \text{CIRCLE}) \in \mathcal{X}^{\text{cat}}$ . The neighborhood resulting of the Gower distance has exactly six categorical components, at distance one from  $\mathbf{u}$ . [Table 1](#) compares three neighborhood-generation strategies producing exactly six points. Each point has same continuous variables of  $\mathbf{x}_*$  and is different than  $\mathbf{u}$ . The first and simplest strategy uses the Gower distance. The second strategy uses an objective-based categorical distance [\[6\]](#), induced by a mixed-variable Inverse Distance Weighting (IDW) regression model. The third strategy employs the surrogate-based neighborhoods from [Definition 2](#), using GPs as surrogate models.

The results highlight clear differences between the three strategies. The Gower distance and the objective-based neighborhood fail to identify any feasible point, whereas the surrogate-based neighborhood proposed identifies four feasible points. Recall that [Figure 2b](#) shows the results of a Latin Hypercube Sampling with 100 000 samples per categorical component, suggesting that only six

**Table 1: Neighbors of  $u = (B, \text{WOOD}, \text{CIRCLE})$  proposed by three strategies.**

Strategy	Neighbor	Supplier	Material	Shape	$f$	$g_1$	$g_2$	Feasible
Gower distance	1	B	ALUM	CIRCLE	18.4928	-0.1300	0.0150	No
	2	B	STEEL	CIRCLE	14.5128	-0.0225	0.0625	No
	3	B	COMP	CIRCLE	9.5329	-0.0800	0.0575	No
	4	B	WOOD	SQUARE	24.0428	0.0050	0.0650	No
	5	B	WOOD	ELLIPSE	24.5628	-0.1075	0.0125	No
	6	A	WOOD	CIRCLE	28.3468	0.3450	0.5600	No
Objective-based [6]	1	A	WOOD	CIRCLE	28.3468	0.3450	0.5600	No
	2	B	WOOD	SQUARE	24.0428	0.0050	0.0650	No
	3	B	ALUM	CIRCLE	18.4928	-0.1300	0.0150	No
	4	B	WOOD	ELLIPSE	24.5628	-0.1075	0.0125	No
	5	B	STEEL	CIRCLE	14.5128	-0.0225	0.0625	No
	6	B	COMP	CIRCLE	9.5329	-0.0800	0.0575	No
Surrogate-based	<b>1</b>	<b>a</b>	<b>alum</b>	<b>circle</b>	<b>18.2868</b>	<b>-0.0225</b>	<b>-0.0451</b>	<b>Yes</b>
	2	B	ALUM	CIRCLE	18.4928	-0.1300	0.0150	No
	3	A	WOOD	CIRCLE	28.3468	0.3450	0.5600	No
	<b>4</b>	<b>a</b>	<b>wood</b>	<b>ellipse</b>	<b>24.3568</b>	<b>-0.6900</b>	<b>-0.4715</b>	<b>Yes</b>
	<b>5</b>	<b>b</b>	<b>steel</b>	<b>square</b>	<b>10.0028</b>	<b>-0.0225</b>	<b>-0.0451</b>	<b>Yes</b>
	<b>6</b>	<b>a</b>	<b>comp</b>	<b>square</b>	<b>8.8169</b>	<b>-0.0225</b>	<b>-0.0451</b>	<b>Yes</b>

categorical components are likely to admit feasible continuous regions. The proposed method identifies four of these six components within the six evaluations, while the incumbent component accounts for another one. The only remaining feasible component is (B, ALUM, SQUARE), which appears to be extremely unlikely to yield feasible points, with only 0.01% feasibility observed in a LHS of 100 000 points in Figure 2b.

Only the surrogate-based strategy improves the objective value while maintaining feasibility. The best improvement is obtained with the categorical component (A, COMP, SQUARE), yielding an objective value of 8.8169. This component also corresponds to that of the best solution identified in Section 2 using an LHS with 100,000 points per component.

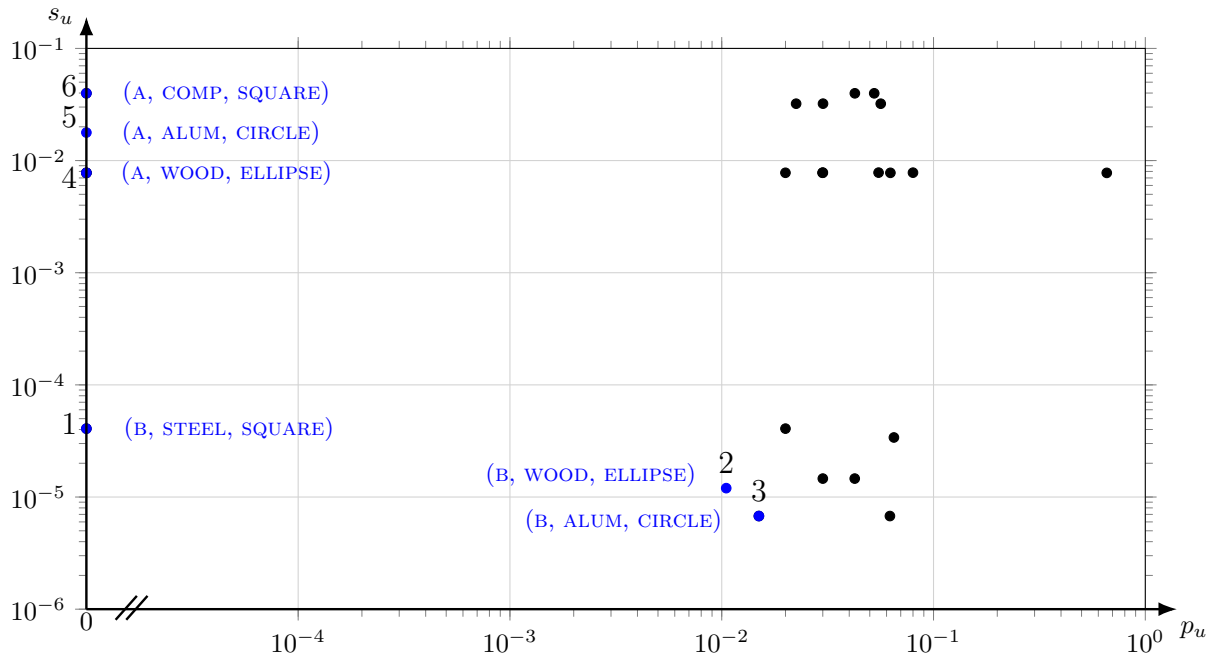
The ordering and selection of categorical components in the surrogate-based neighborhood are presented in Figure 5. The figure plots the primary and secondary ranking functions  $p_u$  and  $s_u$  introduced in Section 4.3, with logarithmic scales for readability. The selected components are shown in blue, together with their categorical component and a number indicating their ranking. On the horizontal axis  $p_u$ , an axis break is introduced to display components with distance zero on a logarithmic scale. In the figure, the feasible neighbors all have a distance of zero with respect to  $p_u$ : they lie on the vertical axis. The objective-based approach identifies (B, STEEL, SQUARE), which corresponds to the component ranked fifth by the surrogate-based approach.

## 5 Adapting surrogate-based neighborhoods for mixed-variable direct search

The categorical neighborhoods developed in Section 4 are now used for optimization in this section, which also reintroduces the integer and continuous variables discarded previously.

### 5.1 Adapting the surrogate-neighborhoods for mixed-variables domains

The pseudo-distance and distance functions from Section 4 require minor adaptations now that the surrogate models are defined on the domain  $\mathcal{X}$  rather than on the categorical set  $\mathcal{X}^{\text{cat}}$ . They are now computed between points in the domain, while the quantitative variables are held fixed so that comparisons only concern the categorical variables. More precisely, for a given and fixed quantitative



**Figure 5: Ordering of categorical components with the surrogate-based neighborhood in the mechanical-part design problem. The incumbent  $u = (B, WOOD, CIRCLE)$  is at the origin and it is not displayed.**

component  $\mathbf{x}^{\text{qnt}} \in \mathcal{X}$ , the distances introduced in Section 4 are computed on the set

$$(\mathcal{X} \times \mathcal{X})_{\mathbf{x}^{\text{qnt}}} := \{(\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{X} : \mathbf{x}^{\text{qnt}} = \mathbf{y}^{\text{qnt}}\}. \quad (15)$$

The distance and pseudo-distance functions are said to be parameterized by the quantitative component  $\mathbf{x}^{\text{qnt}}$ , since it influences their behavior without appearing as an explicit argument.

Consequently, the primary and secondary ranking functions  $p_{\mathbf{x}} : \mathcal{X} \rightarrow \mathbb{R}_+$  and  $s_{\mathbf{x}} : \mathcal{X} \rightarrow \mathbb{R}$  are now defined on the domain  $\mathcal{X}$  and parameterized at a point  $\mathbf{x} \in \mathcal{X}$ , which can be either feasible or infeasible. In this context, an argument of the ranking functions shares the same quantitative component as the incumbent solution, that is  $\mathbf{x}^{\text{qnt}}$ . The ordering relation  $\preceq$  thus compares points of the domain sharing the same quantitative component, making the comparisons effectively on categorical components.

Finally, surrogate-based neighborhoods must be generalized to the mixed-variable setting. They are redefined by considering the ordering relation on the full domain and by choosing that the neighborhoods themselves contain categorical components only. This choice ensures consistency with both the CatMADS framework and the notion of local minima discussed in the next section. The following definition adapts Definition 2 in the more general mixed-variable case.

**Definition 3** (Mixed-variable surrogate-based neighborhood). For a given point  $\mathbf{x} \in \mathcal{X}$  and integer  $m \in \{1, 2, \dots, |\mathcal{X}^{\text{cat}}|\}$ , the *surrogate-based neighborhood*  $\mathcal{N}(\mathbf{x}; m) \subseteq \mathcal{X}^{\text{cat}}$  is the set containing  $m$  categorical components of the lowest ordering with respect to an ordering relation  $\preceq$  defined on  $\mathcal{X}$ . The relation  $\mathbf{y} \preceq \mathbf{z}$  is satisfied whenever

$$\begin{aligned} \mathbf{y} &= (\mathbf{y}^{\text{cat}}, \mathbf{x}^{\text{qnt}}) \in \mathcal{X} & \text{where} & \quad \mathbf{y}^{\text{cat}} \in \mathcal{X}^{\text{cat}} \cap \mathcal{N}(\mathbf{x}; m) \\ \text{and } \mathbf{z} &= (\mathbf{z}^{\text{cat}}, \mathbf{x}^{\text{qnt}}) \in \mathcal{X} & \text{where} & \quad \mathbf{z}^{\text{cat}} \in \mathcal{X}^{\text{cat}} \setminus \mathcal{N}(\mathbf{x}; m). \end{aligned}$$

Note that [Definition 3](#) reduces to [Definition 2](#) when restricted to categorical components. Moreover, a consequence of this definition is that for any  $\mathbf{x} \in \mathcal{X}$  and  $m \in \{1, 2, \dots, |\mathcal{X}^{\text{cat}}|\}$ ,

$$\mathcal{N}(\mathbf{x}; m) = \{\mathbf{y}_{(1)}^{\text{cat}}, \mathbf{y}_{(2)}^{\text{cat}}, \dots, \mathbf{y}_{(m)}^{\text{cat}}\} \subseteq \mathcal{X}^{\text{cat}}$$

in which  $\mathbf{y}_{(1)}^{\text{cat}} = \mathbf{x}^{\text{cat}}$  and where  $(\mathbf{y}_{(1)}^{\text{cat}}, \mathbf{x}^{\text{qnt}}) \preceq (\mathbf{y}_{(2)}^{\text{cat}}, \mathbf{x}^{\text{qnt}}) \preceq \dots \preceq (\mathbf{y}_{(m)}^{\text{cat}}, \mathbf{x}^{\text{qnt}})$ . In the next sections, the surrogate-based neighborhoods from [Definition 3](#) are used to handle the categorical variables.

## 5.2 Background on CatMADS

CatMADS is a direct search framework that iteratively seeks points yielding a strict decrease of the objective function and/or the constraints [\[6\]](#). It generalizes MADS by tackling categorical variables with neighborhoods induced by data from the problem. The quantitative variables are essentially treated as they are in MADS. For simplicity, CatMADS is presented in [Algorithm 1](#) without considering constraints.

---

### Algorithm 1: The CatMADS framework for unconstrained problems (adapted from [\[6\]](#)).

---

0. **Initialization.** Set  $k = 0$ , perform an Design of Experiment (DoE) and define an initial mesh
  1. **Opportunistic search** (optional).
  2. **Opportunistic poll.** Perform polling around the incumbent solution
  3. **Opportunistic extended poll** (optional). If Steps 1 & 2 are unsuccessful, perform quantitative polls around points in  $P_{(k)}^{\text{cat}}$  with objective function values sufficiently close to  $f(\mathbf{x}_{(k)})$
  4. **Update.** Set  $k \leftarrow k + 1$ , update mesh and check stopping criterion  
If the iteration is successful, increase mesh size and **GO TO** Step 1.  
Else, decrease mesh size, and if the mesh is at its minimum size, then **STOP**
- 

Before any optimization is done, a DoE samples points in the domain using a portion of the budget of evaluations. The DoE is used to gather information on the problem and construct categorical neighborhoods, as well as for determining an initial solution.

The main steps of the CatMADS algorithm are Steps 1, 2, and 3. If a solution with strictly lower objective function value is found during these steps, it becomes the incumbent solution and the iteration is deemed *successful*. Otherwise, the iteration is *unsuccessful*. The *search* is an optional step allowing flexible evaluation of points at the intersection of the mesh and the domain. The *poll* is the core mechanism of CatMADS. At iteration  $k$ , trial points are generated around the incumbent  $\mathbf{x}_{(k)}$  through two polls, a quantitative one and a categorical one. The quantitative poll fixes the categorical component and performs a standard MADS poll on the quantitative variables. The categorical poll fixes the quantitative component and explores neighboring categorical components selected through a surrogate-based ordering relation. Two consecutive quantitative polls on  $\mathbb{R}^2$  are illustrated in [Figure 6](#). [Figure 6c](#) illustrates both poll types.

Arrows emerging from an incumbent solution represent a positive basis. Quantitative components produced are on the *mesh* in gray and within the black square, called the *frame*. The frame delimits the region in which a quantitative poll can produce quantitative components.

In the example in [Figure 6c](#),  $x_{(k)}^{\text{cat}} = \text{RED}$  and the number of components in the neighborhood is three. The neighborhood contains the RED (incumbent), YELLOW, and PURPLE categories. Each colored plane represents the quantitative domain associated with one categorical component. Vertical arrows correspond to the categorical poll, while the quantitative poll is performed within the RED plane.

If both the search and poll steps fail to improve the objective, the optional step, called the *extended poll*, may be launched at Step 3. This step revisits points in the categorical poll  $P_{(k)}^{\text{cat}}$  that almost improved the incumbent value  $f(\mathbf{x}_{(k)})$  by calibrating their quantitative variables with additional evaluations.

On an unsuccessful iteration, the mesh and frame are further discretized. CatMADS terminates when the mesh and frame have reached their minimum allowable discretization.

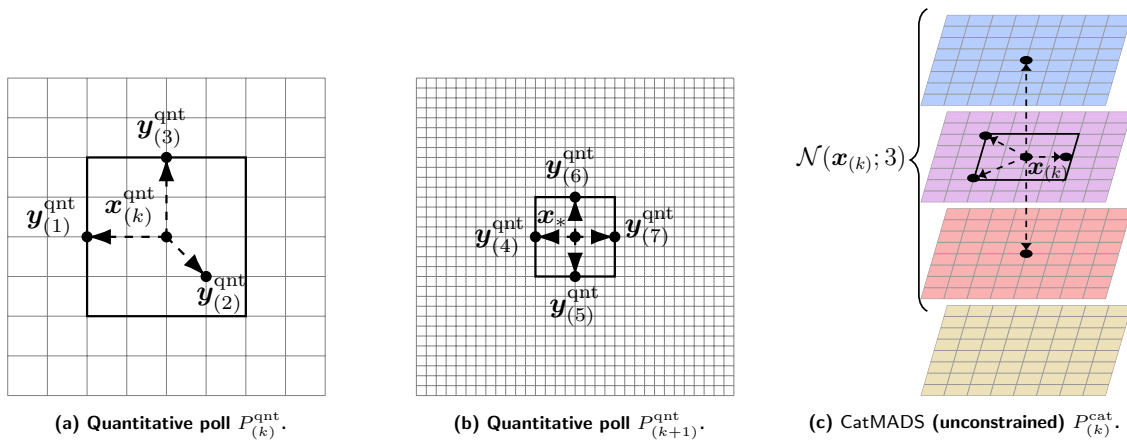


Figure 6: (a)–(b) MADS (quantitative) polls at iteration  $k$  and  $k + 1$  where  $x_{(k)} = x_{(k+1)}$ , and (c) a CatMADS poll.

In CatMADS, the constraints are handled with the progressive barrier strategy (PB) [5]. The PB works with two incumbent solutions, each with their own independent poll, as described above. The *feasible poll* is performed around the *feasible incumbent solution*  $x_{\text{FEA}}$ , and this solution is considered feasible when it strictly improves the objective function. The *infeasible poll* is performed at the *infeasible incumbent solution*  $x_{\text{INF}}$ , and this solution is forced to become more feasible through the iterations. For more details on the update with the PB, see [8, Chapter 12].

### 5.3 Improvements of CatMADS with surrogate-based neighborhoods

In [6], categorical neighborhoods are constructed using information from the objective function only. Moreover, the implementation in [6] relies on a simple mixed-variable interpolation. The surrogate-based neighborhoods introduced here do not affect the theoretical convergence results, as the proposed method remains an instance of the CatMADS framework. However, the resulting categorical neighborhoods incorporate information from both the objective and the constraint functions and rely on an objective distance induced by kernel-based similarities. As a result, the categorical polls should be guided more effectively toward feasible regions than in the original implementation. Again, this has no impact on the theoretical results, but it is expected to improve empirical performance and accelerate convergence. The improved categorical poll with the surrogate-based neighborhoods is illustrated in Figure 7.

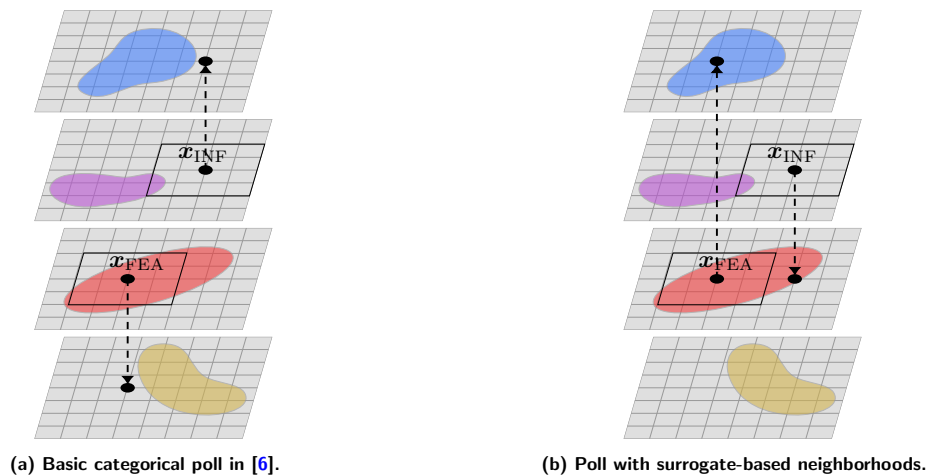


Figure 7: The CatMADS framework in the presence of constraints.

In [Figure 7a](#), the infeasible regions are in gray and the feasible regions are colored. The feasible categorical poll starts from the red category and selects the yellow category, which is the most similar with respect to the objective function. The generated trial point is infeasible.

In contrast, in [Figure 7b](#), the generated trial point lies in the blue category.

A similar behavior is observed for the infeasible categorical poll. In the basic implementation, the generated trial point comes from the category that is closest with respect to the objective function, which again leads to an infeasible trial point in [Figure 7a](#). In comparison, the surrogate-based implementation selects a category that is less similar in terms of the objective but considers the similarities of both the objective and the constraint functions. This results in a feasible trial point in the red category in [Figure 7b](#).

## 6 Computational experiments

This section describes the benchmarking of an instance of CatMADS using the novel neighborhoods, as described in [Section 5](#). This is a proof of concept demonstrating the utility of surrogate-based neighborhoods in mixed-variable blackbox optimization. Since the surrogate models are GPs with one-hot encoding of the categorical variables, the new resulting method is referred to as CatMADS<sub>GP</sub>.

The implementation of CatMADS<sub>GP</sub> is identical to that of CatMADS, except for the construction of the neighborhoods and a BO search step strategically reusing the surrogates. Moreover, in CatMADS, IDW interpolation is constructed following the DoE, whereas CatMADS<sub>GP</sub> also updates the GPs as long as a BO search step is performed. To mitigate computational costs, the BO search step and update of GPs are stopped whenever the number of evaluations exceeds 500 or 33% of the budget of evaluations. The GPs are implemented with the SMT Python library [\[34\]](#). For a given problem, the number of neighbors is given by  $m = \max(3, |\mathcal{X}^{\text{cat}}|^{1/2})$  [\[6\]](#). This value scales with the total number of categories  $|\mathcal{X}^{\text{cat}}|$ , and ensures there are at least two different categorical components other than the incumbent one. For the detailed implementation, see [\[6, Section 4.1\]](#).

The benchmarking includes state of the art solvers used in different mixed-variable blackbox optimization applications [\[21, 28, 35\]](#):

- [Pymoo](#), a Python library containing a collection of optimization algorithms. Specifically, the mixed-variable implementation of a genetic algorithm is used [\[10\]](#);
- [Optuna](#), a mixed-variable solver extending the metaheuristic covariance matrix adaptation evolution strategy (CMA-ES) to categorical variables [\[2\]](#). The solver is also available as a Python implementation;
- [CatMADS](#), a prototype implementation built on the NOMAD software. It serves as the reference implementation of the original CatMADS framework, from which CatMADS<sub>GP</sub> is derived.

### 6.1 Optimization results for the mechanical-part problem

The mechanical-part design problem from [Section 2](#) is now treated as a mixed-variable blackbox optimization problem. To emulate a realistic blackbox setting in which function evaluations are expensive, a small budget of only 200 evaluations is allowed. All solvers are initialized with the same DoE comprising 40 evaluations, corresponding to 20% of the budget, without any feasible point. As shown in [Figure 2](#), an important difficulty of this problem is the identification of a feasible solution using a limited number of evaluations.

Convergence graphs are presented in [Figure 8](#). The plot on the left shows the progress of the constraint aggregation function, and the one on the right shows the progress of the best feasible objective function. Feasibility is declared when  $h(\mathbf{x}_{\text{FEA}}) \leq 10^{-8}$ , as represented by the horizontal dashed line. On

the left plot, the curves are interrupted as soon as a feasible solution is generated. The plots on the right start at this value.

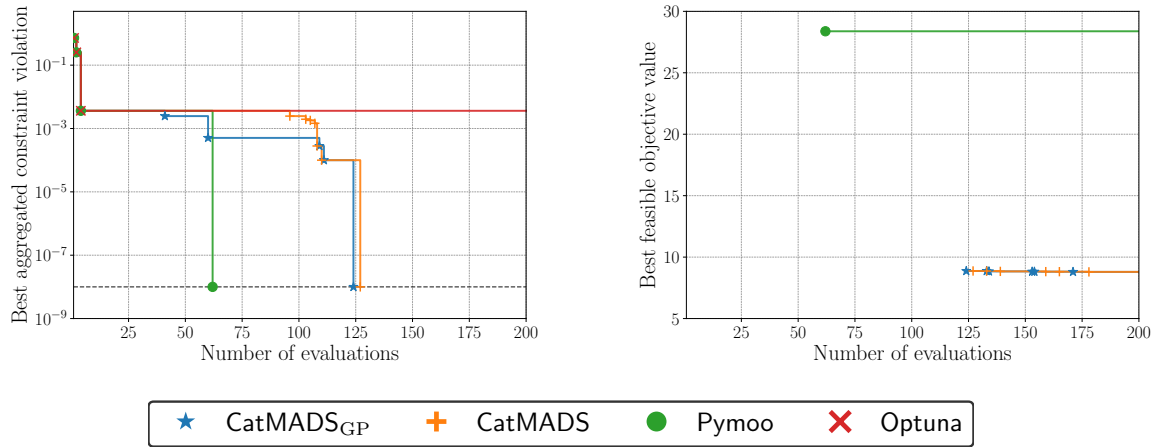


Figure 8: Convergence graphs for the mechanical-part design problem with a budget of 200.

Pymoo is the first method to reach feasibility, as shown on the left of Figure 8. However, the corresponding objective function value is high and remains unchanged throughout the budget. This indicates that Pymoo is stuck in a suboptimal categorical component. In contrast, CatMADSGP and CatMADS reach feasibility later, but with lower objective values. Both methods identify significantly better solutions, suggesting that they reach more favorable categorical components. Finally, Optuna fails to produce a feasible point.

The mechanical-part problem was run with multiple seeds. The seed affects both the DoE and the stochastic components of the solvers, such as sampling procedures or the generation of directions. The results lead to similar conclusions as those observed in Figure 8. Pymoo reaches feasibility in most runs, and when it does, it is usually the first one. However, in approximately 75% of the runs, the resulting objective function value remains high, as observed in Figure 8. CatMADSGP and CatMADS consistently achieve the best objective values, and CatMADSGP reaches feasibility faster than CatMADS. Optuna rarely finds a feasible point.

## 6.2 Data profiles with existing solvers

In this section, CatMADSGP is tested on the 30 unconstrained and 30 constrained mixed-variable problems of the Cat-Suite collection [17]. As in CatMADS [6], the evaluation budget per problem is set to  $250n$ , where  $n$  is the number of variables, and the initial DoE consists of 20% of this budget.

Each problem is instantiated with three different seeds, resulting in a total of 180 instances. An instance, denoted  $p \in \mathcal{P}$ , corresponds to a problem with a specific seed. The set of all instances is denoted  $\mathcal{P}$ . The convergence test for an instance  $p \in \mathcal{P}$  depends on an initial objective value  $f_0 \in \mathbb{R}$ . This value is defined as

- the least objective function value in the common DoE, for unconstrained problems [9];
- the smallest objective function value among the first feasible solutions found by the solvers, for constrained problems [9].

This definition enables the comparison of solvers even when the DoE fails to produce a feasible solution.

The set of solvers in the comparison is  $\mathcal{S} = \{\text{CatMADSGP}, \text{CatMADS}, \text{Pymoo}, \text{Optuna}\}$ . A solver  $s \in \mathcal{S}$   $\tau$ -solves an instance  $p \in \mathcal{P}$  if it produces a feasible incumbent solution  $\mathbf{x}_{\text{FEA}} \in \Omega$  such that the

reduction  $f_0 - f(\mathbf{x}_{\text{FEA}})$  is within  $\tau$  of the smallest reduction  $f_0 - f_*$  obtained by any solver on this instance [26]. More precisely, the  $\tau$ -convergence test is defined as

$$f_0 - f(\mathbf{x}_{\text{FEA}}) \geq (1 - \tau)(f_0 - f_*), \tag{16}$$

where  $\tau \in [0, 1]$  is a given tolerance.

A data profile [26] represents the fraction of instances  $\tau$ -solved by a solver  $s \in \mathcal{S}$  as a function of the computational budget. It is defined as

$$\text{data}_s(\kappa) := \frac{1}{|\mathcal{P}|} \left| \left\{ p \in \mathcal{P} : \frac{k_{p,s}}{n_p + 1} \leq \kappa \right\} \right| \in [0, 1], \tag{17}$$

where  $k_{p,s} \geq 0$  is the number of evaluations required by solver  $s$  to  $\tau$ -solve instance  $p$ , and  $n_p$  is the number of variables in  $p$ . In the following plots,  $\kappa$  represents budgets measured in multiples of  $(n_p + 1)$  evaluations. The data profiles comparing CatMADS<sub>GP</sub> with the solvers from the literature on the unconstrained and constrained instances are presented in Figures 9a and 9b.

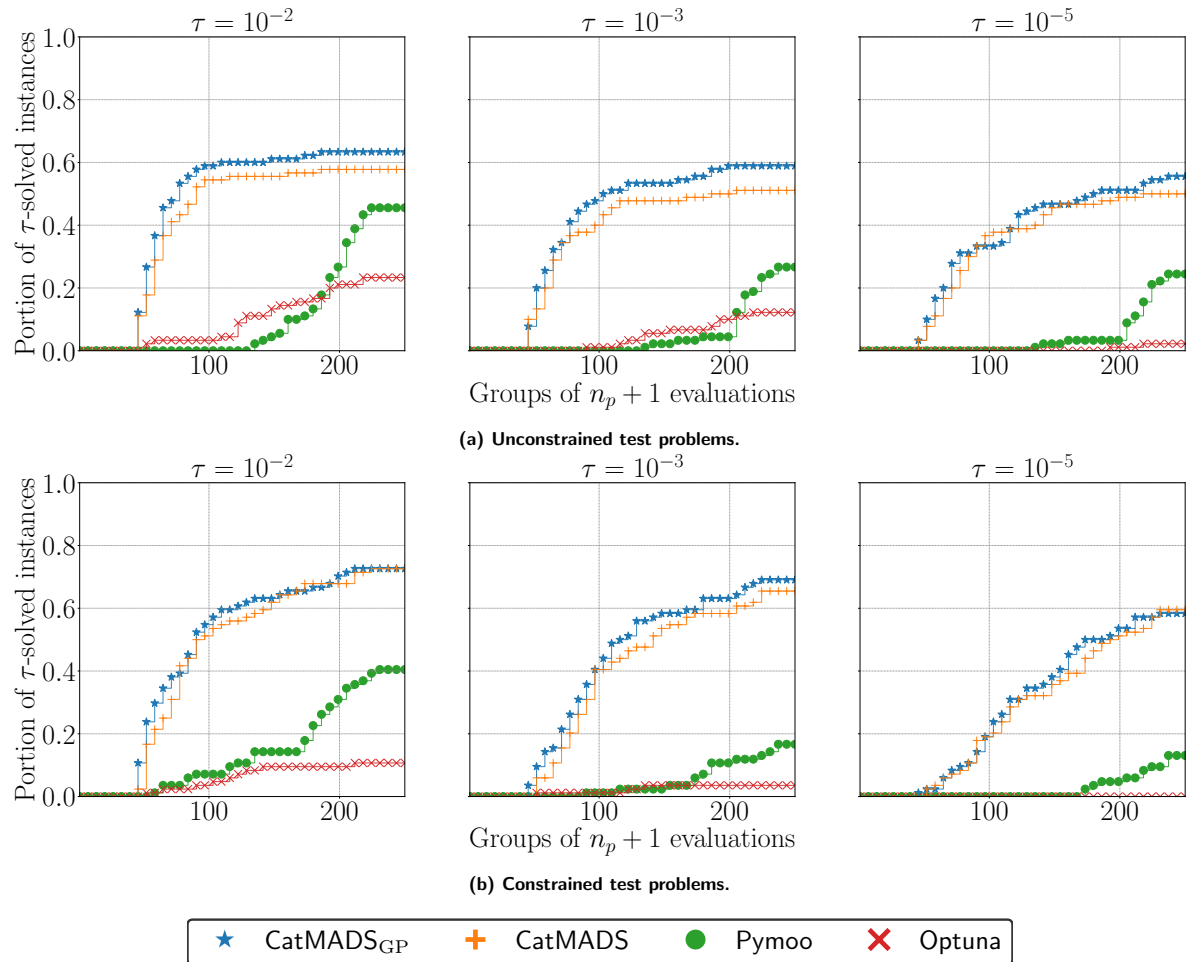


Figure 9: Comparison results on the test problems.

The data profiles for unconstrained problems in Figures 9a and 9b show that CatMADS<sub>GP</sub> performs best across all tolerances, followed by CatMADS. The performances of the three other methods significantly deteriorate as the tolerance  $\tau$  gets smaller. For instance, at  $\tau = 10^{-2}$  and  $250(n_p + 1)$  evaluations, CatMADS<sub>GP</sub>  $\tau$ -solves approximately 65% of the problems, compared to 60% for CatMADS, 45% for Pymoo and 25% for Optuna.

For the constrained case, the profiles in Figure 9b show that both CatMADS variants clearly outperform Pymoo and Optuna over all tolerances. To better analyze the relative performance of the two variants, Figure 10a shows the performance profiles constructed using only the two CatMADS variants. CatMADS<sub>GP</sub> dominates CatMADS for the three tolerances on the constrained problems. The most significant improvement is at  $\tau = 10^{-3}$ .

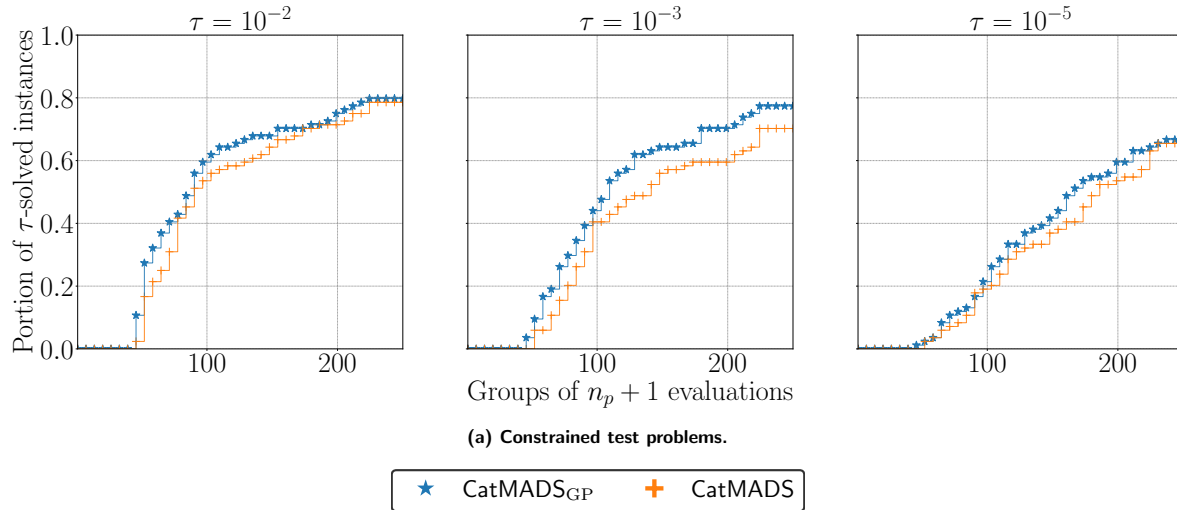


Figure 10: Comparison results on the constrained test problems between the two CatMADS variants.

## 7 Discussion

This work introduces a novel approach to structure categorical variables through neighborhoods constructed from surrogate models. These neighborhoods incorporate information from both the objective and the constraint functions, making them suitable for constrained mixed-variable optimization. To balance similarity with respect to the objective and constraint functions, the construction relies on dominance relations.

As a proof of concept, the surrogate-based neighborhoods are first illustrated on the mechanical-part design problem. A distinctive feature of this problem is the difficulty of identifying feasible points among the categorical components. The proposed neighborhoods navigate this challenge more effectively than alternative neighborhood strategies.

The surrogate-based neighborhoods are then integrated into the CatMADS algorithm, resulting in the new method CatMADS<sub>GP</sub>, that is tested against several state-of-the-art solvers. Data profiles indicate that CatMADS<sub>GP</sub> consistently achieves better performance on the Cat-Suite collection. These results suggest that surrogate-based neighborhoods provide a promising mechanism for constrained mixed-variable blackbox optimization.

Further improvements could extend this study. The current implementation of CatMADS<sub>GP</sub> uses a fixed number of neighbors throughout the iterations, and more sophisticated strategies could dynamically adjust the number of neighbors. Future extensions would define and analyze neighborhoods involving meta variables [7], *i.e.*, variables whose values dictate the number of variables and/or constraints present in the optimization problem.

## References

- [1] M.A. Abramson, C. Audet, J.W. Chrissis, and J.G. Walston. Mesh Adaptive Direct Search Algorithms for Mixed Variable Optimization. *Optimization Letters*, 3(1):35–47, 2009.

- [2] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. Optuna: A Next-generation Hyperparameter Optimization Framework. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Association for Computing Machinery, 2019.
- [3] N. Aronszajn. Theory of Reproducing Kernels. Transactions of the American mathematical society, 68(3):337–404, 1950.
- [4] C. Audet and J.E. Dennis, Jr. Mesh Adaptive Direct Search Algorithms for Constrained Optimization. SIAM Journal on Optimization, 17(1):188–217, 2006.
- [5] C. Audet and J.E. Dennis, Jr. A Progressive Barrier for Derivative-Free Nonlinear Programming. SIAM Journal on Optimization, 20(1):445–472, 2009.
- [6] C. Audet, Y. Diouane, E. Hallé-Hannan, S. Le Digabel, and C. Tribes. CatMADS: Mesh Adaptive Direct Search for constrained blackbox optimization with categorical variables. Technical Report G-2025-42, Les cahiers du GERAD, 2025.
- [7] C. Audet, E. Hallé-Hannan, and S. Le Digabel. A General Mathematical Framework for Constrained Mixed-variable Blackbox Optimization Problems with Meta and Categorical Variables. Operations Research Forum, 4(12), 2023.
- [8] C. Audet and W. Hare. Derivative-Free and Blackbox Optimization. Springer Series in Operations Research and Financial Engineering. Springer, Cham, Switzerland, 2017.
- [9] C. Audet, W. Hare, and C. Tribes. Benchmarking constrained, multi-objective and surrogate-assisted derivative-free optimization methods. Technical Report G-2025-36, Les cahiers du GERAD, 2025.
- [10] J. Blank and K. Deb. Pymoo: Multi-Objective Optimization in Python. IEEE Access, 8:89497–89509, 2020.
- [11] D. Brigo, F. Mercurio, and F. Rapisarda. Parameterizing correlations: a geometric interpretation. IMA Journal of Management Mathematics, 18(1):55–73, 2007.
- [12] J.H. Bussemaker, P.D. Ciampa, T. De Smedt, B. Nagel, and G. La Rocca. System Architecture Optimization: An Open Source Multidisciplinary Aircraft Jet Engine Architecting Problem. In AIAA AVIATION 2021 Forum. American Institute of Aeronautics and Astronautics, 2021.
- [13] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 6(2):182–197, 2002.
- [14] E.C. Garrido-Merchán and D. Hernández-Lobato. Dealing with categorical and integer-valued variables in Bayesian Optimization with Gaussian processes. Neurocomputing, 380:20–35, 2020.
- [15] A. Gretton, K.M. Borgwardt, M.J. Rasch, B. Schölkopf, and A. Smola. A Kernel Two-Sample Test. The Journal of Machine Learning Research, 13(25):723–773, 2012.
- [16] E. Hallé-Hannan, C. Audet, Y. Diouane, S. Le Digabel, and P. Saves. A distance for mixed-variable and hierarchical domains with meta variables. Neurocomputing, 653:131208, 2025.
- [17] E. Hallé-Hannan, C. Audet, Y. Diouane, S. Le Digabel, and C. Tribes. Cat-Suite: A collection of optimization problems with categorical and quantitative variables for benchmarking. Technical Report G-2025-39, Les cahiers du GERAD, 2025.
- [18] R. Hamano, S. Saito, M. Nomura, K. Uchida, and S. Shirakawa. CatCMA: Stochastic Optimization for Mixed-Category Problems. Technical Report 2405.09962, Arxiv, 2024.
- [19] T. Hastie, R. Tibshirani, and J. Friedman. The Elements of Statistical Learning. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.
- [20] P. Jäckel and R. Rebonato. The Most General Methodology to Create a Valid Correlation Matrix for Risk Management and Option Pricing Purposes. Journal of Risk, 2(2):17–27, 1999.
- [21] Y. Jina and P.V. Kumar. Bayesian optimisation for efficient material discovery: a mini review. Nanoscale, 15(26):10975–10984, 2023.
- [22] D. Lakhmiri. HyperNOMAD. <https://github.com/bbopt/HyperNOMAD>, 2019.
- [23] S. Le Digabel and S.M. Wild. A taxonomy of constraints in black-box simulation-based optimization. Optimization and Engineering, 25(2):1125–1143, 2024.
- [24] S. Lucidi, V. Piccialli, and M. Sciandrone. An Algorithm Model for Mixed Variable Programming. SIAM Journal on Optimization, 15(4):1057–1084, 2005.
- [25] J. Mercer. Functions of positive and negative type, and their connection the theory of integral equations. Philosophical transactions of the royal society of London A, 209(441-458):415–446, 1909.
- [26] J.J. Moré and S.M. Wild. Benchmarking Derivative-Free Optimization Algorithms. SIAM Journal on Optimization, 20(1):172–191, 2009.

- [27] J. Pelamatti, L. Brevault, M. Balesdent, E.-G. Talbi, and Y. Guerin. Efficient global optimization of constrained mixed variable problems. *Journal of Global Optimization*, 73(3):583–613, 2019.
- [28] R.P. Prager and H. Trautmann. Exploratory Landscape Analysis for Mixed-Variable Problems. *IEEE Transactions on Evolutionary Computation*, 2024.
- [29] P.Z.G. Qian, H. Wu, and C.F.J. Wu. Gaussian process models for computer experiments with qualitative and quantitative factors. *Technometrics*, 50(3):383–396, 2008.
- [30] C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.
- [31] O. Roustant, E. Padonou, Y. Deville, A. Clément, G. Perrin, J. Giorla, and H. Wynn. Group kernels for Gaussian process metamodels with categorical inputs. *Uncertainty Quantification*, 8(2):775–806, 2020.
- [32] P. Saves, N. Bartoli, Y. Diouane, T. Lefebvre, J. Morlier, C. David, E. Nguyen Van, and S. Defoort. Constrained Bayesian Optimization Over Mixed Categorical Variables, with Application to Aircraft Design. In *AeroBest*, 2021.
- [33] P. Saves, Y. Diouane, N. Bartoli, T. Lefebvre, and J. Morlier. A mixed-categorical correlation kernel for Gaussian process. *Neurocomputing*, 550:126472, 2023.
- [34] P. Saves, R. Lafage, N. Bartoli, Y. Diouane, J.H. Bussemaker, T. Lefebvre, J.T. Hwang, J. Morlier, and J.R.R.A. Martins. SMT 2.0: A Surrogate Modeling Toolbox with a focus on Hierarchical and Mixed Variables Gaussian Processes. *Advances in Engineering Software*, 188:103571, 2024.
- [35] P. Sridevi, Z. Arefin, and S.I. Ahamed. An integrated machine learning and hyperparameter optimization framework for noninvasive creatinine estimation using photoplethysmography signals. *Healthcare Analytics*, 7:100395, 2025.
- [36] E.-G. Talbi. Metaheuristics for variable-size mixed optimization problems: A unified taxonomy and survey. *Swarm and Evolutionary Computation*, 89:101642, 2024.
- [37] Y. Zhang, S. Tao, W. Chen, and D.W. Apley. A Latent Variable Approach to Gaussian Process Modeling with Qualitative and Quantitative Factors. *Technometrics*, 62(3):291–302, 2020.