Design of experiments for mixed continuous and discrete variables

Thi Thoi Tran, Sébastien da Veiga, Delphine Sinoquet, Marcel Mongeau

Design of experiments for mixed continuous and discrete variables

Thi Thoi Tran[a,c], Sébastien Da Veiga[b], Delphine Sinoquet[a,*], Marcel Mongeau[c]

[a] *IFP Energies Nouvelles, France*
[b] *Safran Tech, France*
[c] *ENAC, Université de Toulouse, France*

Abstract

Design of experiments (DoE) are used in various contexts such as optimization or uncertainty quantification when relying on a time-consuming numerical simulator. It aims to select a limited number of points at which evaluating the simulator provides maximal knowledge on the simulator outputs of interest. One motivating application is the optimal design of turbine blades in an helicopter engine, which takes as inputs mixed continuous and binary variables. This paper proposes two new approaches for space-filling design over the mixed continuous and discrete space. Numerical results for three different types of DoE problems (mixed integers, mixed binaries with cyclic symmetry, and time series) are presented. The obtained results illustrate the good performance of the proposed methods and the wide range of applications they can address.

Keywords: `Design of experiments, space-filling designs, necklace distance, mixed continuous and discrete variables, super-sample kernel herding, reproducing kernel Hilbert spaces (RKHS), global alignment kernels`

[*] delphine.sinoquet@ifpen.fr

1. Introduction

In recent years, the need for efficient design of experiments (DoEs) has emerged as a key research area for analyzing complex physical numerical models. Indeed, such models are often expensive to evaluate, requiring sometimes several hours or even days to run one single simulation. It means that using them to conduct optimization studies or uncertainty quantification and sensitivity analysis investigations is in general too computationally demanding. Standard practice consists in building a surrogate model of the numerical code, and using it as a proxy for all intensive computations required by optimization or uncertainty propagation. However, the final surrogate accuracy heavily depends on the available samples of the computer code inputs/outputs relationship. For this task, it is often crucial to build a design of experiments that provides information about all parts of the experimental region. For example, two appealing concepts are the so-called *space-filling* designs, [1, 2] and Latin Hypercube Sampling (LHS) [3]. Designs of experiments for continuous variables have been extensively studied with several space-filing criteria (minimax and maximin [4], discrepancy [5], maximum projection [6]). However, the case where models involve both continuous and discrete input variables has been much less investigated and tested.

In this paper, we focus on the specific case of models involving a *mixed* design of experiments region, defined as

$$\mathcal{D} = \{z = (x, y) \in \mathbb{R}^m \times \mathbb{I}^n\}, \tag{1}$$

where $x \in \mathbb{R}^m$, and $y \in \mathbb{I}^n$ are the continuous and discrete variables, respectively, and where $\mathbb{I}$ denotes the discrete space (*e.g.*, integer, binary or categorical variables). For handling mixed DoEs, the first proposed approaches consisted mainly of simple extensions of continuous LHS, either by randomly discretizing continuous values [3], or by rounding continuous DoEs to obtain feasible integer candidates [7, 8]. These techniques can recover the integer domain but may destroy the desirable properties of the original DoEs, or even worse, they may generate identical points, which must be avoided. On the other hand, it is possible to generate independently a LHS for every given discrete possible values [2], or use a so-called *sliced LHS* (LHS for the continuous factors which is sliced into groups of smaller LHS designs associated with different discrete levels) [9]. Although popular, these last two approaches usually require a very large number of samples that increases too rapidly with the number of discrete variables. Recent work [10, 11] proposes to sample continuous variables through a single continuous LHS, while the discrete variables are obtained by randomly assigning an equal number of data samples to each discrete value. A different line of work mainly studied in the machine learning research area relies on the approximation of probability measures with an empirical probability distribution supported by a small number of points, also called *quantization* [12]. Our proposal follows this point of view.

More precisely, we shall focus on the framework of kernel-embedding of probability distributions [13], which offers mathematical tools to define distances between probability distributions that can be efficiently computed. It is then straightforward to recast the problem of finding a DoE with good space-filling properties as an optimization problem that aims at finding an empirical probability distribution (the DoE) which is as close as possible to a target probability measure (for example, the uniform measure on the feasible experimental region). For continuous probability measures, this point of view has already been proposed with the name of *kernel herding* in [14], or *support points* in [15]. Note that the target distribution here can either be given explicitly or it can itself be seen as an empirical distribution but involving a large number of points. Here, our goal is to extend this framework to the mixed case.

The paper is organized as follows. First, we present in Section 2 a naive extension of continuous DoEs which will serve as a baseline for numerical comparisons. In Section 3, two novel methods based on kernel embedding are introduced and discussed with respect to their respective advantages and limitations. Extensive numerical experiments are finally conducted in Section 4 where we show that, on a variety of problems, the kernel point of view clearly outperforms standard approaches.

2. Straightforward transformation of a continuous DoE into a mixed DoE

Our goal is to select n_{DoE} representative points from the region D defined by (1). Inspired from previous works [16, 17], we first propose a straightforward algorithm which is based on the projection from continuous space to integers representing the distinct indices of the discrete variables, detailed in Algorithm 1.

Algorithm 1: Projected DoE for discrete variables

```
Input: $n, nDoE$
0. Pre-processing
    • Compute all $n_{levels}$ distinct discrete arrangements of size $n$, and arrange them in a matrix $L$
    • Compute the weight, $w = \frac{1}{n_{levels}}$
1. Main computation
    • Build a continuous DoE of size $n_{DoE}$ in dimension 1:
      $U = \{u_1, \dots, u_{n_{DoE}}\}$
    • Assign discrete values according to the following rule:
      if $u_i \in [(j-1)w, jw], i = 1, \dots, n_{DoE}, j = 1, \dots, n_{levels}$, then assign the $j-$th row of $L$ to $y_i$
Return $y_1, \dots, y_{n_{DoE}}$
```

The choice of the method to generate the continuous DoE is left to the user, but for the sake of having a uniform distribution in the discrete space, it is recommended to use either a low discrepancy method such as Sobol sequences, Halton sequences (see details in [18]), or standard space-filling DoEs such as the minimax or maximin DoEs for example [4, 3].

Remark that this adaptation focuses on picking distinct discrete candidates. If it yields badly-distributed points (this will happen if the chosen continuous DoE is of poor quality), then the algorithm must consider postprocessing duplicated discrete points (to be avoided). Algorithm 1 only provides the DoE for $\mathbb{I}^n$: in order to obtain the full DoE (for $\mathbb{R}^m \times \mathbb{I}^n$) another (continuous) DoE is generated for $\mathbb{R}^m$ and added to the output of Algorithm 1. In the numerical comparisons of Section 4, we shall use a Sobol sequence and a classical LHS for both generating continuous DoEs for Algorithm 1 and above for the adding step.

3. Kernel-embedding of probability distributions for mixed DoEs

This section discusses the main ingredients of our two proposals relying on kernel-embedding of probability distributions extended to mixed continuous and discrete space. The first part presents a literature review on kernel-embeddings and how defining a DoE can be seen as an optimization problem in this framework. Subsections 3.1 and 3.2 detail both proposed algorithms, *Greedy-MDS* and *Adapted-Greedy.*

For two probability distributions P and Q and a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$ with positive-definite kernel k, the maximum mean discrepancy (MMD) distance between P and Q is defined as

$$\mathrm{MMD}(\mathrm{P}, \mathrm{Q}) = \|\mu_{\mathrm{P}} - \mu_{\mathrm{Q}}\|_{\mathcal{H}}, \tag{2}$$

where $\mu_{\mathrm{P}} = \int k(x, .)d\mathrm{P}(x)$ and $\mu_{\mathrm{Q}} = \int k(x, .)d\mathrm{Q}(x)$ are the *kernel embeddings* of P and Q, respectively, and serve as representations of the probability distributions [13]. Interestingly, the RKHS framework makes it possible to write the distance with only expectations of kernel functions:

$$\mathrm{MMD}^2(\mathrm{P}, \mathrm{Q}) = \mathbb{E}_{\xi,\xi' \sim \mathrm{P}} k(\xi, \xi') + \mathbb{E}_{\zeta,\zeta' \sim \mathrm{Q}} k(\zeta, \zeta') - 2\mathbb{E}_{\xi \sim \mathrm{P}, \zeta \sim \mathrm{Q}} k(\xi, \zeta). \tag{3}$$

Kernel-embedding of probability distributions has been studied in many contexts, ranging from two-sample test problems [19], independence tests [20] or generative artificial neural networks [21]. In our context, it has also been used as a method to quantize probability measures, i.e. how well a representative point set approximates a target distribution [14, 15, 5]. More precisely, given a target probability measure Q to be quantized with an empirical distribution $\mathrm{P} = \frac{1}{n_{DoE}} \sum_{i=1}^{n_{DoE}} \delta_{x_i}$ of n_{DOE} points, the problem writes

$$\min_{x_1, x_2, \ldots, x_{n_{DoE}}} \mathrm{MMD}^2 \left(\frac{1}{n_{DoE}} \sum_{i=1}^{n_{DoE}} \delta_{x_i}, \mathrm{Q} \right). \tag{4}$$

When the target Q is the uniform distribution on an hypercube, the squared MMD actually can be written as well-known discrepancy measures, depending on the choice of the kernel k, as elaborated in [22]. However, the choice of the target Q is not restricted, and several options have been discussed in previous work. When Q is explicitly given as a parametric distribution, [15] proposes a minimization-maximization algorithm to solve (4). On the other hand, when Q is a posterior distribution from a Bayesian problem, [23] investigates a specific kernel k giving rise to the so-called *Stein discrepancy*. Finally, Q can also be given itself as an empirical measure $\mathrm{Q} = \frac{1}{N} \sum_{i=1}^{N} \delta_{u_i}$, where N is typically much larger than n_{DoE}. This case is at the core of *kernel herding* [14], which has also been studied in [15]. For the mixed DoE case, we shall only focus on the latter, meaning that in what follows we shall always assume that we have a large sample $(u_i)_{i=1,\ldots,N}$ of points distributed according to the target Q. In this case, problem (4) transforms into

$$\min_{x_1, x_2, \ldots, x_{n_{DoE}}} \mathrm{MMD}^2 \left(\frac{1}{n_{DoE}} \sum_{i=1}^{n_{DoE}} \delta_{x_i}, \frac{1}{N} \sum_{i=1}^{N} \delta_{u_i} \right). \tag{5}$$

It remains, as a last step, to decide how this optimization problem is solved. When the points x_i can be chosen freely, efficient optimization strategies based on minimization-maximization have been proposed [15]. However they cannot be easily generalized to the mixed case, since they would involve developing a specific mixed-integer solver. This is the reason why we concentrate on the case where the points x_i are chosen among the points in the large sample $\mathcal{U} = \{u_i\}$, as in [14], [23], [24]. The optimization problem then however turns into a combinatorial one, with a complexity which must be leveraged with approximations.

In kernel herding [14], a greedy sequential algorithm is proposed, which consists of the following steps:

$$x_1^* = \operatorname*{argmax}_{x \in \mathcal{U}} \frac{1}{N} \sum_{j=1}^{N} k(x, u_j),$$

$$x_{t+1}^* = \operatorname*{argmin}_{x \in \mathcal{U}} \frac{1}{t+1} \sum_{i=1}^{t} k(x, x_t^*) - \frac{1}{N} \sum_{j=1}^{N} k(x, u_j).$$

From a computational perspective, we only precompute $\frac{1}{N} \sum_{j=1}^{N} k(x, u_j)$ once for all $x \in \mathcal{U}$, and it is also recommended to normalize all the points before running the algorithm to ensure its accuracy. A pseudo-code is given in Algorithm 2. Note that this is also the approach taken in [23] (they also extend it to the selection of batches of several points at each iteration by resorting to a mixed-integer quadratic programming solver).

Algorithm 2: Kernel herding: greedy algorithm for continuous variables.

```
Input: Target distribution given as the empirical measure supported on
  the points $\mathcal{U} = (u_j)_{j=1,2,\dots,N}$, and the size of the DoE $n_{DoE}$.
  • Precompute
        $\frac{1}{N} \sum_{j=1}^{N} k(x, u_j), \quad \forall x \in \mathcal{U}.$
  • Initialize
        $x_1^* = \operatorname*{argmax}_{x \in \mathcal{U}} \frac{1}{N} \sum_{j=1}^{N} k(x, u_j).$    (6)
  • For $t = 1, \dots, n_{DoE} - 1$:
        $x_{t+1}^* = \operatorname*{argmin}_{x \in \mathcal{U}} \frac{1}{t+1} \sum_{i=1}^{t} k(x, x_t^*) - \frac{1}{N} \sum_{j=1}^{N} k(x, u_j).$    (7)
Return $x_1^*, x_2^*, \dots, x_{n_{DoE}}^*$
```

Regarding the choice of the kernel, if $(u_j)_{j=1,\dots,N}$ lie in the continuous space

$\mathbb{R}^m$, we have to specify a kernel function

$$\begin{aligned} k :&\mathbb{R}^m \times \mathbb{R}^m \longrightarrow \mathbb{R} \\ &(u, v) \longrightarrow k(u, v) \end{aligned} \tag{8}$$

To define a m dimensional kernel k, one can typically use a tensorized kernel

$$k(u, v) = \prod_{l=1}^{m} \tilde{k}(u^l, v^l),$$

where $\tilde{k} : \mathbb{R} \times \mathbb{R} \longrightarrow \mathbb{R}$ denotes a one-dimensional kernel. Classical examples include the Gaussian and Laplacian kernels

$$\begin{aligned} \tilde{k}_{Gauss}(u, v) &:= e^{-\gamma(u-v)^2}, \\ \tilde{k}_{Lap}(u, v) &:= e^{-\gamma|u-v|} \end{aligned}$$

which depend on a user-defined hyperparameter $\gamma > 0$. In our numerical experiments, we shall rely on the so-called *median* rule of thumb to select γ [20].

3.1. Greedy-MDS: kernel herding and multi-dimensional scaling

In this subsection, we propose a first extension of the previous greedy algorithm to mixed continuous and discrete space by using Multi-Dimensional Scaling (MDS). The main idea is to apply MDS to build a continuous encoding of the discrete variables.

Continuous encoding of discrete variables is a standard way to handle mixed variables in regression or classification problems. Successful strategies rely on a data-driven approach, where the encoding is optimized during the supervised learning task [25, 26]. However in our unsupervised setting, such approaches cannot be envisioned. In order to circumvent this limitation, we take a different road by assuming that a user-defined distance in the discrete space characterizing any prior information is available. Such a distance can include symmetry properties such as in the numerical experiments presented below, or handcrafted correlation given by experts of the phenomenon under study. Once this distance is available, it can be used to compute all the pairwise distances between the points from the target distribution u_i.

MDS can then be applied to this distance matrix. Indeed, MDS is a technique that creates a map displaying the relative positions of a number of objects, given only a table of distances (or dissimilarities) between them. In other words, MDS translates pairwise-distance information among a set of N objects (normally in a high-dimensional space) into a configuration of N corresponding points in a smaller dimensional space, preserving the pairwise distances as much as possible. Suppose that we have a set of N points $\{u_1, u_2, \ldots, u_N\}$ for which we know all the pairwise distances $d_{ij} = d(u_i, u_j)$. The output of MDS is a new set of points $v_1, v_2, \ldots, v_N$ in a Euclidean space such that their Euclidean distance approximates d_{ij}, *i.e.*, $d_{ij} \approx \hat{d}_{ij} = \|v_i - v_j\|$. The dimension of this Euclidean space is chosen by the user, with a maximum equal to $N - 1$. To measure how well the original set of data are represented by the MDS model, it is typical to use the *the goodness-of-fit* value, or *stress* value [27], based on the differences between the actual distances and their predicted values:

$$stress = \sqrt{\frac{\sum_{i,j}(d_{ij} - \hat{d}_{ij})^2}{\sum_{i,j} d_{ij}^2}}. \tag{9}$$

Based on the stress value, [27] asserts how the MDS model performs: if $stress = 0$ the MDS model is said to be *perfect*, else if $stress \backsim 0.025$ the MDS model is said to be *excellent*, else if $stress \backsim 0.05$ the MDS model is said to be *good*, else if $stress \backsim 0.1$ the MDS model is said to be *fair*, and finally otherwise if $stress \backsim 0.2$ or $stress \geq 0.2$ the MDS model is said to be *poor*.

Our proposal, called Greedy-MDS, is then to apply a preliminary MDS step for all the discrete variables $Y = \{y_1, y_2, \ldots, y_N\}$ in the large target sample for which we have all the pairwise distances defined by the user. After this step, we have a continuous encoding $V = \{v_1, v_2, \ldots, v_N\}$ of the discrete variables, which can be stacked with the continuous variables $X = \{x_1, x_2, \ldots, x_N\}$. This gives a new representation of the large target sample with only continuous values. Then, we use kernel-herding from Algorithm 2 with a standard continuous kernel to obtain a small subset of points which represents as well as possible the target distribution. The last step is finally to invert the encoding: this is an easy task, since by bijection we exactly know the correspondence between the original discrete values and their encoding. A pseudo-code is given in Algorithm 3.

```
Algorithm 3: Greedy-MDS algorithm.
  Input: $nDoE, d, Y$
    1. Pre-processing step
        • Compute $d$, the matrix of distances $d_{ij} = d(y_i, y_j)$ for all $y_i, y_j \in Y$.
        • MDS step: apply MDS to $d$, obtain accordingly continuous points,
          $u_1, \dots, u_N$
        • Stacking with continuous variables: for each $u_i, i = 1, \dots, N$, add a
          large number of continuous values, save in matrix $R$
    2. Greedy algorithm Algorithm 2 with inputs $R, nDoE$.
    3. Retrieve the original values by inverting MDS
```

A strong potential limitation of Greedy-MDS lies in the choice of the user-defined distance. Indeed, very poor distance reconstruction performance can occur in practice, meaning that MDS cannot find an encoding which preserves well the pairwise distances even with a maximal dimension of $N-1$. To circumvent this issue, we observed numerically that in practice MDS has a much better reconstruction performance if, instead of relying on a user-defined distance, we employ a user-defined kernel k for discrete variables and build the corresponding kernel-induced pseudo-distance

$$d_k(u, v) = \sqrt{k(u, u) + k(v, v) - 2k(u, v)}.$$

Note however that requiring a user-defined kernel is much more demanding than requiring a distance, and that if a kernel is available the following approach should be preferred.

3.2. Adapted-Greedy: directly using a kernel defined in the mixed space

In this subsection, we propose an adapted greedy algorithm for mixed continuous and discrete variables problems that takes into account directly an *ap-*

propriate mixed kernel inside the greedy algorithm 2, *appropriate* being related here to the distance and type of information that are given by the user. The first step is to build a positive-definite kernel on the discrete variables from a given distance $d(\cdot,\cdot)$. The naive approach consists in generalizing the *Laplace kernel* as follows:

$$k(y,y') = e^{-\gamma d(y,y')},$$

for any values y and y' of the discrete variables. Unfortunately this does not provide a positive-definite kernel in general [28]. A particular case of interest where this result holds concerns *binary* variables with the Hamming distance, $d_H(y,y')$, where the kernel

$$k_H(y,y') = e^{-\gamma d_H(y,y')}, \tag{10}$$

is positive definite for any $\gamma > 0$, see [29]. For the general case, we have to follow a different road and build upon the work of [30] where the so-called *soft string kernel* is introduced. For any distance d between any variables u and v in a space $\mathcal{I}$, it is given by

$$k^{soft}(u,v) := \int\limits_{\omega \in \mathcal{I}} p(\omega) e^{-\gamma(d(u,\omega)+d(v,\omega))} d\omega \tag{11}$$

where $p(\omega) : \mathcal{I} \to \mathbb{R}$ is a probability distribution over $\mathcal{I}$, and it can be proved that it is positive definite [30]. When $\mathcal{I}$ is a discrete space, it boils down to the discrete sum

$$k^{soft}(u,v) := \sum_{\omega \in \mathcal{I}} p(\omega) e^{-\gamma(d(u,\omega)+d(v,\omega))}. \tag{12}$$

Interestingly, for a uniform distribution, this kernel is an approximation of the naive proposal above, by using well-known approximation results for the *Log-SumExp function (LSE)*. More precisely, the LSE function reads

$$\mathrm{LSE}(x_1, x_2, \ldots, x_n) = \log(e^{x_1} + e^{x_2} + \ldots + e^{x_n}).$$

The LSE function can be seen as an approximation of the maximum function $\max(x_1, \ldots, x_n)$ [31]. Consequently, we also have

$$\min(x_1, \ldots, x_n) \approx -\log(e^{-x_1} + e^{-x_2} + \ldots + e^{-x_n}).$$

Now if p is a uniform distribution on $\mathcal{I}$, meaning that for all $\omega \in \mathcal{I},\ p(w) = p_0$, then the soft string kernel simplifies as follows

$$\begin{aligned} k^{soft}(u,v) &= \sum_{\omega_i \in \mathcal{I}} p_0\, e^{-\gamma(d(u,\omega_i)+d(v,\omega_i))}, \\ &= \sum_{\omega_i \in \mathcal{I}} e^{(-\gamma\eta_i + \log(p_0))}, \end{aligned} \tag{13}$$

with $\eta_i = d(u,\omega_i) + d(v,\omega_i)$. Applying $-\log$ to both sides of this equality leads to

$$\begin{aligned} -\log(k^{soft}(u,v)) &= -\log\left(\sum_{\omega_i \in \mathcal{I}} e^{(-\gamma\eta_i + \log(p_0))}\right) \\ &\approx \min_i(\gamma\eta_i - \log(p_0)), \end{aligned}$$

using the approximation of the minimum given above. From the distance triangle inequality, we also have $\eta_i \geq d(u,v)$, which means that $\min_i(\gamma\eta_i - \log(p_0)) = \gamma d(u,v) - \log(p_0)$. This finally gives

$$\begin{aligned} k^{soft}(u,v) &= e^{\log(k^{soft}(u,v))} \\ &\approx e^{-(\gamma d(u,v) - \log(p_0))} \\ &= p_0\, e^{-\gamma d(u,v)}. \end{aligned}$$

This result can help interpret k^{soft} as a positive-definite approximation of the Laplace kernel where we would insert the distance d.

Combining one of the above kernels (10) or (12) for the discrete variables and any kernel k_{cont} for the continuous variables is finally straightforward with tensorization, and leads to a mixed kernel that can directly be used in kernel herding.

Extension beyond mixed variables problems.

Interestingly, the Adapted-Greedy approach presented before is generic, in the following sense: as we discussed building a kernel adapted to discrete variables, the same principle can be applied to any other type of variables (corresponding to appropriate kernel and distance) provided by the user. A prominent example in computer experiments involves time series variables, for which it is usually a challenge to build a space-filling DoE. Some previous works are for example based on the decomposition of functional data on orthogonal bases such as in [32], but here we can directly make use of the literature on kernels for time-series data. We can for example propose a normalized kernel between two time series $u(t)$ and $v(t)$ involving the recently proposed *global alignment kernel* [33]:

$$K^{GAK}(u(t), v(t)) = K(u(t), v(t)) - \frac{1}{2}(K(u(t), u(t)) + K(v(t), v(t))), \qquad (14)$$

where K is the global alignment kernel for same length time series

$$K(u(t), v(t)) = \prod_{i=1}^{|} e^{-\phi_\sigma(u(t_i), v(t_i))}, \qquad (15)$$

with $\phi_\sigma(u(t_i), v(t_i)) = \frac{1}{2\sigma^2}\|u(t_i) - v(t_i)\|^2 + \log(2 - e^{-\frac{\|u(t_i)-v(t_i)\|^2}{2\sigma^2}})$. We shall see in Section 4 how it performs numerically on a time series example.

4. Numerical experiments

The aim of this section is to demonstrate the large range of applicability of the proposed design of experiment methods. The first application is the surrogate modelling of the simulated performance of an electric engine with respect to design parameters of the rotor component. This application deals with continuous, integer and categorical variables. Then, we consider the case of cyclic-symmetry problems, with continuous and binary variables, motivated by an application of optimal design of the turbomachine of an helicopter engine described in [16]. For these specific problems, we address two different practical objectives: (a) surrogate modelling illustrated on various benchmark functions, and (b) the choice of initial points for the shape optimization problem of the turbomachine. Finally, our DoE methods are applied to two examples involving time series.

4.1. Design of experiments for a surrogate model with continuous, integer and categorical variables

The first application relates to the optimal design of an electric engine with respect to the rotor component. The purpose of this study is to choose a set of values of the design variables that will be used to perform simulations of the engine operation in order to build surrogate models of the maximal power and of the maximal torque of the engine. The simulations are performed with the FEMM simulator[1] designed for electromagnetic problems. The design variables are composed of one continuous variables (the rotor length), two integer variables (the number of wires and the number of coils with respectively four and eleven possible values) and one categorical variable that characterizes the type of rotor geometry (with 16 possible types of rotor). For this problem, we use a tensorized kernel composed of a Gaussian kernel for continuous variables and the kernel (10) based on the Hamming distance applied to mixed binary variables resulting from the encoding of the integer and the categorical variables. Here we compare designs of experiments built with Adapted-Greedy approach to the projected LHS (Algorithm 1) with a number of points ranging from 20 to 100. Surrogate models based on Gaussian processes and adapted kernel for mixed continuous and categorical variables ([34] , [9]) are built from simulations performed on these designs. The accuracy of the obtained surrogate models is assessed by

$$1 - Q^2(F, \hat{F}) = \frac{\frac{1}{N_v}\sum_{i=1}^{N_v}(F_i - \hat{F}_i)^2}{\frac{1}{N}\sum_{i=1}^{N}(F_i - \bar{F})^2}, \tag{16}$$

where Q^2, the predictive accuracy coefficient, is computed on a new set of validation points of cardinality N_v, $N = N_v + n_{DOE}$ is the total number of simulations including the points of the DoE, F is the vector of simulated values, $\hat{F}$ is the vector of the values predicted by the surrogate model, and $\bar{F}$ is the mean of F. In our experiments, we choose $N_v = 300$ validation points obtained

[1] https://www.femm.info/wiki/HomePage

by a projected LHS method. Figure 1 compares the predictivity of each model built from the designs of experiments for the two simulator outputs of interest: the maximal torque and the maximal power of the electric machine. We achieve a better predictivity of the surrogate models for the Adapted-Greedy method, that is smaller values of $1-Q^2$ than projected LHS for both responses of interest and for all the design sizes.

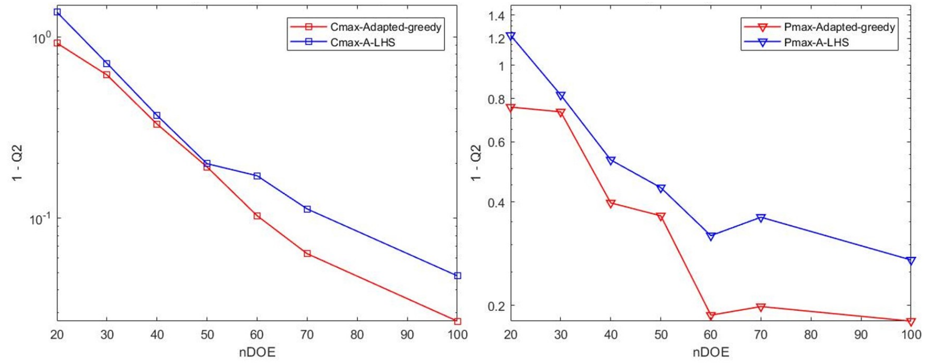

Figure 1: Accuracy of the surrogate models of the maximal torque (left) and the maximal power (right) built with designs of experiments obtained by the projected LHS method (in blue) and by the Adapted-Greedy algorithm (in red) with respect to the size of the designs.

4.2. Design of experiments for cyclic-symmetry problems involving continuous and binary variables

The study of cyclic-symmetry problems is motivated by an application of optimal design for turbine blades in an helicopter engine [35], which involves mixed continuous and binary input variables.

We first apply our novel DoE methods on a set of benchmark functions (all with cyclic symmetry) with the aim of estimating the expectation of the actual functions. The second experiment presented in this section focuses on the choice of the initial points of an optimization method.

For both experiments, the value of the function of interest, f, is not affected by cyclic permutations of the vector of binary variables y. More precisely, for any given value of the continuous-variable vector $x \in \mathbb{R}^m$ and for any $y' \in$

$\{0,1\}^n$ such that there exists an index $i \in \{1,2,\dots,n\}$ such that $Rot^i(y') = y$, (where $Rot^i(y)$ is the rotation of y by i positions), one has $f(x,y') = f(x,y)$. For the Greedy-MDS and the Adapted-Greedy methods, we use a tensorized kernel with, for the binary variables, the soft kernel (12) with the *necklace distance* [16]:

$$d_{neck}(y,y') = \min_{i=1,2,\dots,n} d_H(y, Rot^i(y')), \tag{17}$$

where d_H denotes the Hamming distance. A *necklace* of binary variables $y \in \{0,1\}^n$ is defined as the equivalence class of binary vectors considering all rotations as equivalent vectors (analogy with a necklace of n beads of two colors represented by 0 and 1 values): $\mathcal{N}_y = \{Rot^i(y), i = 1,2,\dots,n\}$.

4.2.1. Results for expectation estimation

The benchmark consists of 6 different functions chosen from [36, 37, 38, 14], and listed in Table 1. These functions were set up originally for continuous optimization purposes. We transform them into mixed binary functions with cyclic symmetry following the methodology of [16]. The problem dimensions are listed in Table 1 with a number m of continuous variables ranging from 1 to 10, and a number n of binary variables from 4 to 7 (corresponding to 6 to 20 *distinct* necklaces, *i.e.*, 6 to 20 equivalent classes of binary vector).

Test problem	Dimension ($[m,n]$)	# distinct binary vectors (necklaces)	source instance	Reference
sin($\|x\|$)-nl	2×7	20	sin($\|x\|$)	[14]
Wong2-nl	10×4	6	Wong2	[36]
Branin-nl	1×7	20	Branin	[38]
Hartman3-nl	3×6	14	Hartman3	[38]
Perm6-nl	5×5	10	Perm6	[37]
Perm8-nl	7×5	20	Perm8	[37]

Table 1: Benchmark functions for expectation estimation.

DoEs are built with the methods introduced in Section 3: two projected designs (1) based on LHS and Sobol sequence, and our two methods based on kernel embeddings: Greedy-MDS and Adapted-Greedy. These results are compared with random sampling and standard LHS with rounded values for the binary variables. The empirical means of the benchmark functions for each necklace are computed from the points of the DoE and compared with the empirical means computed with a large set of points obtained by Monte-Carlo sampling (10^4 evaluations). For each DoE, the accuracy of the estimations is rated by the mean squared errors of the function expectations at the different necklaces:

$$\sqrt{\frac{1}{n_{neck}} \sum_{h=1}^{n_{neck}} (F_h^{ref} - F_h^{DoE})^2}, \tag{18}$$

where n_{neck} is the number of necklaces, F_h^{ref} and F_h^{DoE} are respectively the empirical expectations of the function for the Monte-Carlo sampling points and the points of the DoE for the necklace h.

The results of 20 repeated runs of the 6 methods with a design size $n_{DOE} = k\, n_{neck}$, where k is ranging from 1 to 27, are shown in Figure 2. Figure 3 provides the distributions of estimation errors for each necklace obtained by one run of the 6 methods on the Branin-nl test case with a design size of 100 points ($k = 5$). We observe on both figures that the approximation errors are much smaller for Greedy-MDS and Adapted-Greedy methods than for the four other methods. We also note that the two projected methods based on LHS and Sobol sequences provide slightly better global results on the benchmark when compared with standard methods using rounded values (Figure 2).

4.2.2. Results for initial design of optimization

The aim of the turbine application presented in [35, 16] is to design the turbine blades of an helicopter engine in order to minimize the vibrations of the compressor. This optimization problem involves $m = 1$ continuous parameter controlling the frequency amplitude, and a vector of $n = 12$ binary variables describing the repartition of two reference blade geometries on the turbine disk. In [16], the authors propose an adapted optimization method based on

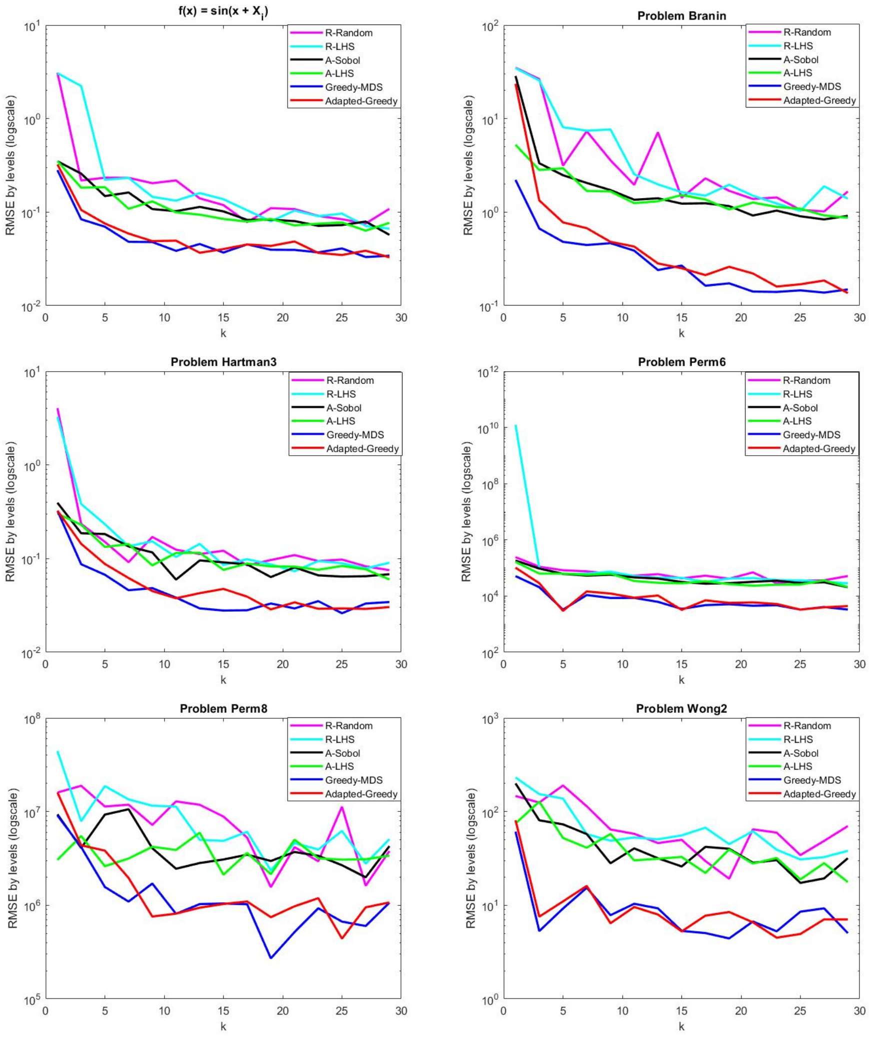

Figure 2: Median RMSE (in logscale) of the estimation of benchmark function expectation obtained with 5 repetitions of the 6 methods: a random sampling and a standard LHS with rounded values for binary variables, 2 projected methods based on Sobol sequence and projected LHS, the Greedy-MDS and Adapted-Greedy methods, for $n_{DoE} = kn_{neck}$.

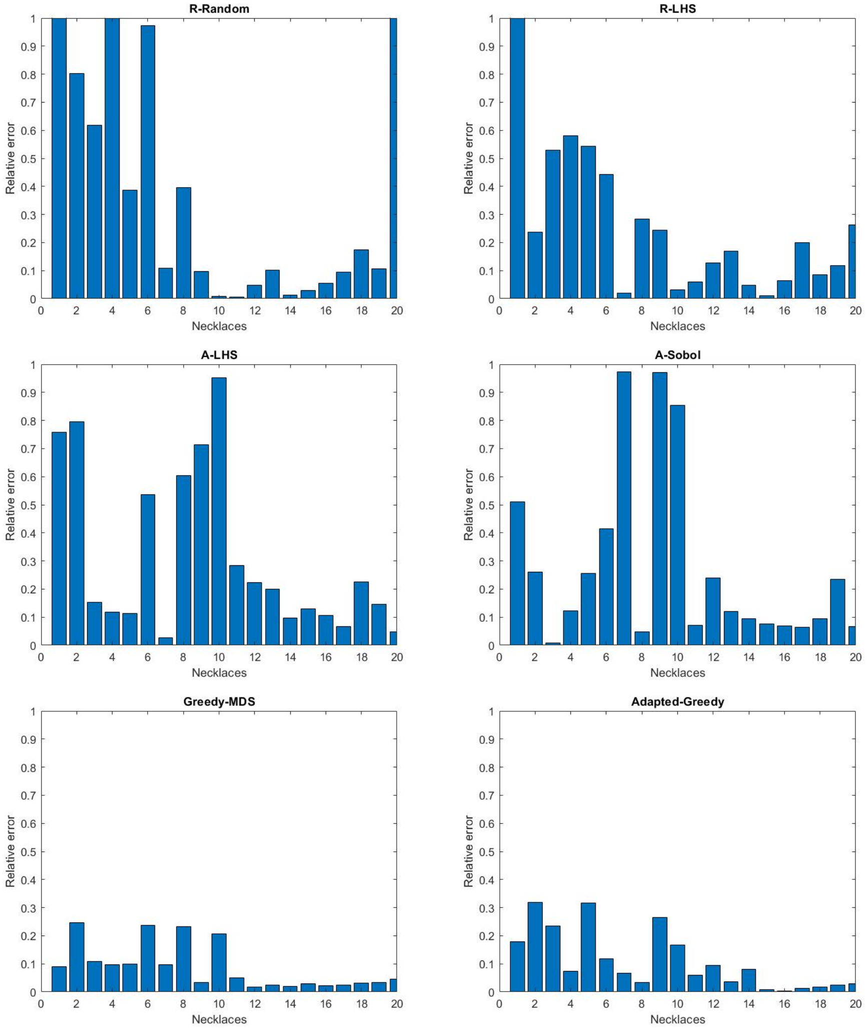

Figure 3: Distributions of estimation errors for each of the 20 necklaces of the Branin-nl function obtained with 6 methods: random sampling and standard LHS with rounded values for the binary variables, 2 projected methods based on Sobol sequence and projected LHS, the Greedy-MDS and Adapted-Greedy methods, for $n_{DoE} = 100$.

a derivative-free trust-region method that uses the necklace distance (17) to take into account the cyclic symmetry of the problem. In [16], the initial set of points are chosen by a LHS procedure with rounded values for binary variables. We compare here these optimization results with the results obtained with the same optimization method coupled with two types of initial DoE: the projected LHS and Adapted-Greedy methods. As for the previous benchmark functions, the chosen kernel for the latter is a tensorized kernel of a Gaussian kernel for continuous variables and the softmax kernel with necklace distance. The size of the initial DoE is $n_{DoE} = n+m+1 = 14$, and 100 repetitions of each DoE methods and associated optimizations are run. The results are reported as data and performance profiles in Figures 4 and 5, respectively. As explained in [39], data and performances profiles with respect to the number of function evaluations are standard tools to compare the performances of derivative-free optimization methods, counting the ratio of *successful* runs for each solver with respect to a chosen criterion, the number of simulations in our study. We consider that an optimization run is successful if the best solution $\bar{z}$ obtained within a given number of function evaluations satisfies

$$f(z_0) - f(\bar{z}) \geq (1-\tau)(f(z_0) - f^*), \qquad (19)$$

where, in the sequel, f^* denote the best function value found by any solver (or the global-minimum value, if known), z_0 is the best point of the initial points for each run, and τ is the desired accuracy, a user-defined tolerance value (in our tests, $\tau = 10^{-5}$). If a run does not provide a solution that satisfies (19), we consider that it *fails*. The data profile displays the ratio of successful runs of each solver over the total number of runs with respect to the number of simulations (normalized by the number of optimization variables $n + m + 1 = 14$), whereas the performance profile displays this ratio with respect to a performance ratio that is, in our case, the ratio of the number of evaluations needed by a given method to satisfy the condition (19), and the smallest number of evaluations for all the compared methods.

Figures 4 and 5 show that the optimization method coupled with DoEs generated by projected LHS is the most robust method with 90% of successful runs, while the optimization coupled with Adapted-Greedy method succeeds to solve 88% of problems. The optimization method coupled with LHS with rounded values for binary variables is the least robust method with only 82% of successful runs. The optimization method coupled with Adapted-Greedy DoE appears to be less efficient for small budgets of simulations. For this kind of derivative-free optimization methods coupled with expensive simulators, the size of the initial set is generally small in order to limit the number of expensive simulations. In the presented application, the size of the initial set (14) is small compared to the number of necklaces (equal to 352). This can explain the small differences between the results of the optimization method coupled with the three DoE methods: the exploration of the input space is done essentially along the optimization iterations and not during the initial phase.

4.3. Design of experiments for times series

To illustrate the ability of the proposed DoEs to be relevant for other types of variables than the mixed discrete case, we propose in this section to apply the Greedy-MDS and the Adapted-Greedy approaches coupled with the global alignment kernel (see (14) in Section 3) to two time series examples, and we compare the results with those obtained with some state-of-the-art approaches.

	Adapted Greedy		Greedy MDS		LHS + FPCA		Quant. +FPCA	
nDoE	t(s)	errors	t(s)	errors	t(s)	errors	t(s)	errors
20	0.04	$\mathbf{8.10^{-8}}$	**0.02**	2.10^{-7}	0.08	2.10^{-7}	0.36	$1.3.10^{-7}$
40	0.07	9.10^{-8}	**0.03**	2.10^{-7}	0.09	3.10^{-8}	0.94	$\mathbf{7^{-9}}$
60	0.11	$\mathbf{2.10^{-8}}$	**0.030**	4.10^{-8}	0.11	6.10^{-8}	2.04	$\mathbf{2^{-8}}$

Table 2: Computational times and relative errors of the expectation estimation obtained for the function f_1 with the functional data of Brownian motion using 4 methods: the Adapted-Greedy and Greedy-MDS methods coupled with the global alignment kernel; the LHS method and the quantization method coupled with the FPCA dimension reduction method.

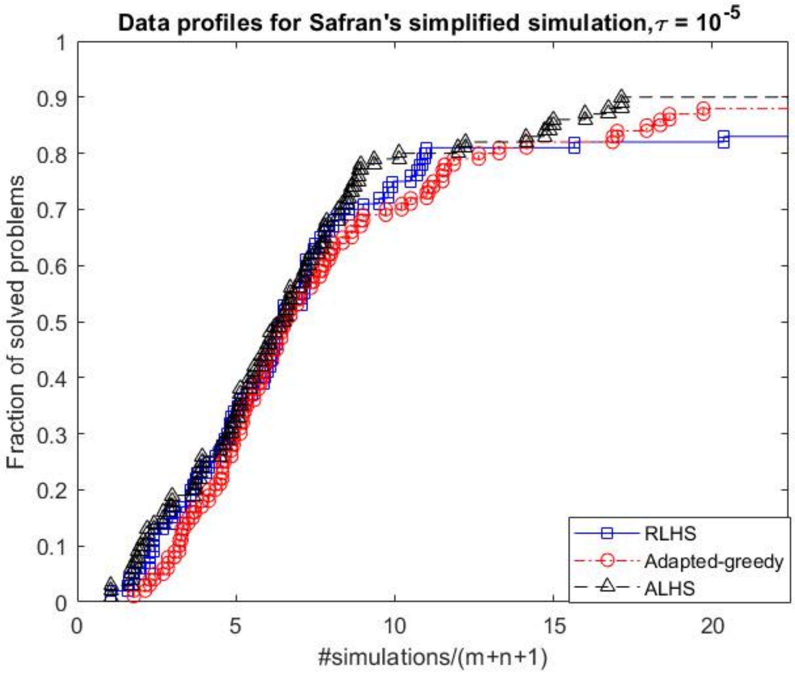

Figure 4: Turbine blade design: data profiles of the optimization runs with 100 initial DoE obtained with 3 methods: LHS with rounded values for binary variables, projected LHS, and Adapted-Greedy methods.

We consider two analytical functions with a functional input random variable $V(t)$ that is known through a sample of 200 realizations. Two cases are studied: V is either a standard Brownian motion, denoted by BM, (Figure 6) or a max-stable process, denoted by MS (Figure 7). The aim of these experiments is to estimate the expectation of the function with respect to the functional variable $V(t)$ with a limited number of samples of V. To achieve this goal, [40, 41] propose a greedy functional data-driven quantization approach coupled with a functional principal component analysis (FPCA) to reduce the dimension of V to a finite (small) dimension. The impact of the number of components in FPCA on the explained variance of the functional data for two types of uncertainties (Brownian and max-stable processes) is discussed in [42]. For the presented numerical experiments, we choose 8 components that explain respectively 97.6%

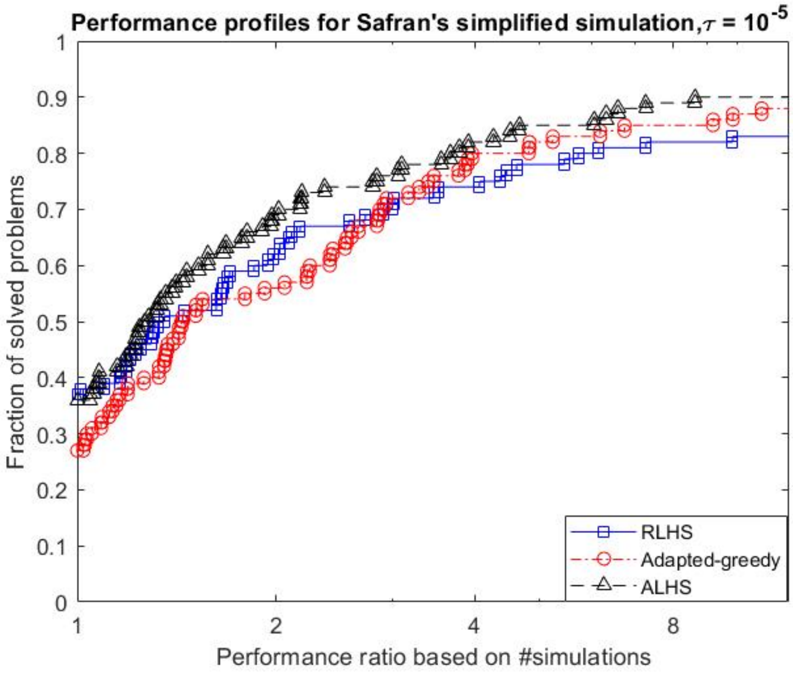

Figure 5: Safran's application: of the optimization runs with 100 initial DoE obtained with 3 methods: LHS with rounded values for binary variables, projected LHS, and Adapted-Greedy methods.

and 70% of the data variance. We propose here to compare this quantization method with our two DoE methods applied directly to the functional domain with the global alignment kernel, K^{GAK}, introduced in Section 3 for time series. The Greedy-MDS method relies on the associated *GAK kernel distance*, d_{GAK}, defined as

$$d^2_{GAK}(u(t), v(t)) = K^{GAK}(u(t), u(t)) + K^{GAK}(v(t), v(t)) - 2K^{GAK}(u(t), v(t)). \tag{20}$$

A standard LHS method applied to the reduced space obtained by FPCA is also evaluated. The two studied functions are defined as

$$f_1 : (x, V) \to x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7 + \int_0^T e^{V_t} dt, \tag{21}$$

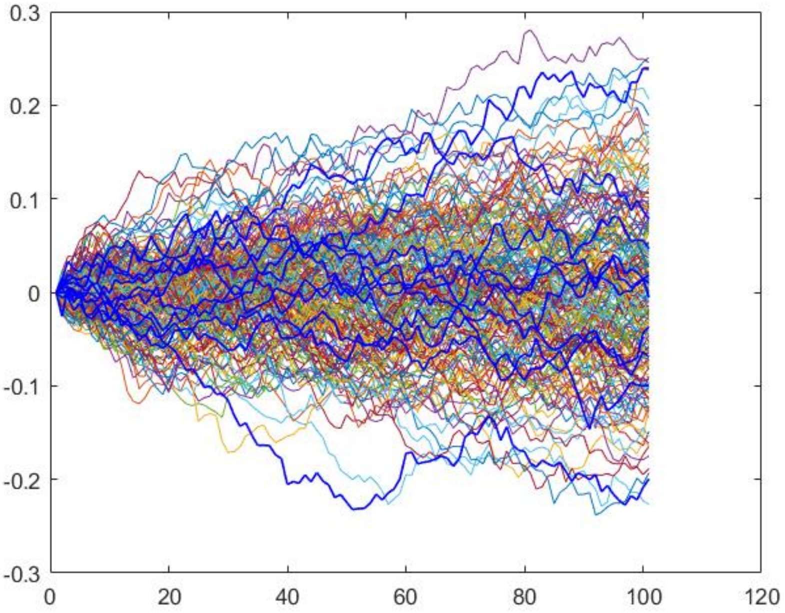

Figure 6: 200 realizations of a Brownian motion: the thick blue curves are the 10 curves selected by the first iterations of the Adapted-Greedy algorithm.

where we fix $x_1 = 50$ and $x_2 = -80$, and

$$f_2 : (x, V) \to 2x_1^2 + x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7 + \int_0^T 2\sin(V_t)dt, \quad (22)$$

where $x_1 = 2.95$ and $x_2 = 3.97$.

We apply the 4 DoE methods to the functions f_1 and f_2 and the two functional data, Brownian motion and max-stable process, with various sizes of DoE $n_{DoE} = 20, 40$ and 60 chosen among the 200 available realizations of V. The performances of the methods are measured in terms of accuracy and computational time in Tables 2, 3, 4, and 5. The errors are computed as the relative errors of the expectation estimations based on the obtained DoE, and the estimation based on the 200 realizations. The best values among the 4 methods are indicated in bold. The results of LHS and the quantization methods are the means of the results obtained for 50 repeated runs.

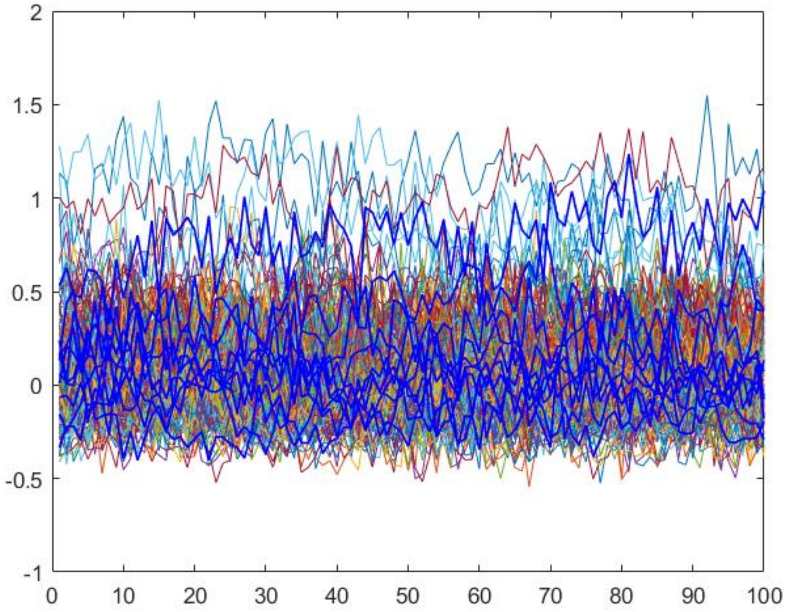

Figure 7: 200 realizations of a Max-stable process. The thick blue curves are the 10 curves selected by the first iterations of the Adapted-Greedy algorithm.

The quantization method which is dedicated to the expectation estimation often yields the best results. Our two methods provide results that are close to it in terms of accuracy, whereas the results of the LHS method applied to the reduced space are in general not as accurate. In terms of computational time, our two methods are more efficient than the quantization method implementation of [41]. Moreover, our methods directly consider the functional variable and do not require to apply a dimension reduction method.

5. Conclusions

The aim of the paper is to propose design of experiment methods adapted to mixed discrete and continuous variable problems. A first proposal is the

	Adapted Greedy		Greedy MDS		LHS + FPCA		Quant. +FPCA	
nDoE	t(s)	errors	t(s)	errors	t(s)	errors	t(s)	errors
20	0.04	$\mathbf{2.10^{-7}}$	**0.03**	2.10^{-6}	0.07	2.10^{-6}	0.4	5.10^{-7}
40	0.07	2.10^{-6}	**0.03**	$\mathbf{2.10^{-8}}$	0.1	8.10^{-7}	0.9	3^{-7}
60	0.11	5.10^{-7}	**0.04**	4.10^{-7}	0.11	4.10^{-7}	1.9	$\mathbf{2.10^{-7}}$

Table 3: Computational times and relative errors of the expectation estimation obtained for the function f_1 with the functional data of Max-stable process. 4 methods are evaluated: the Adapted-Greedy and Greedy-MDS methods coupled with the global alignment kernel, the LHS method and the quantization method coupled with the FPCA dimension reduction method.

straightforward extension of continuous DoE techniques such as low discrepancy methods or standard space filling approaches coupled with the projection from continuous space to integers representing the distinct indices of discrete variables. A second proposed approach relies on kernel-embedding of probability distributions extended to mixed continuous and discrete space, which leads to two new methods: Greedy-MDS and Adapted-Greedy methods. These methods are generic and can address types of other objects than mixed discrete and continuous variables, provided that a suitable kernel is available. A variety of useful kernels are presented: in particular, the softmax kernel that can be built from any distance and ensures the positive-definite property, and the global alignment kernel that is suited to time series.

We illustrated the performances of the proposed DoE methods in various contexts: building a training data set for a surrogate model handling continuous, integer and categorical variables, and for the estimation of the expectation of a function with cyclic symmetry for a vector of binary variables, choosing initial points of a mixed binary and continuous shape optimization problem for an helicopter engine, and finally DoE for time series for expectation estimation of a function that depends on a functional variable.

Future studies can be carried on with other types of objects and associated

	Adapted Greedy		Greedy MDS		LHS + FPCA		Quant. +FPCA	
nDoE	t(s)	errors	t(s)	errors	t(s)	errors	t(s)	errors
20	0.04	3.10^{-7}	**0.02**	6.10^{-7}	0.05	6.10^{-7}	0.3	$\mathbf{2.10^{-7}}$
40	0.07	3.10^{-7}	**0.03**	8.10^{-7}	0.07	2.10^{-7}	0.89	$\mathbf{6.10^{-8}}$
60	0.11	3.10^{-8}	**0.03**	$\mathbf{1.10^{-8}}$	0.09	1.10^{-7}	1.9	3.10^{-8}

Table 4: Computational times and relative errors of the expectation estimation obtained for the function f_2 with the functional data of Brownian motion. 4 methods are evaluated: the Adapted-Greedy and Greedy-MDS methods coupled with the global alignment kernel, the LHS method and the quantization method coupled with the FPCA dimension reduction method.

kernels (such as images or graphs) and also with different operational objectives. For instance, the exploration step in global optimization methods such as surrogate optimization or direct search methods could exploit the proposed DoEs for mixed continuous and discrete problems or involving other types of variables provided that some prior information is available and can be associated to a kernel or a distance (as we did for cyclic symmetry).

	Adapted Greedy		Greedy MDS		LHS + FPCA		Quant. +FPCA	
nDoE	t(s)	errors	t(s)	errors	t(s)	errors	t(s)	errors
20	0.04	$\mathbf{5.10^{-7}}$	**0.03**	1.10^{-6}	0.07	3.10^{-6}	0.35	2.10^{-6}
40	0.07	2.10^{-6}	**0.03**	2.10^{-6}	0.09	1.10^{-6}	0.9	$\mathbf{9.10^{-7}}$
60	0.11	$\mathbf{4.10^{-7}}$	**0.04**	2.10^{-6}	0.11	8.10^{-7}	1.9	6.10^{-7}

Table 5: Computational times and relative errors of the expectation estimation obtained for the function f_2 with the functional data of Max-stable process. 4 methods are evaluated: the Adapted-Greedy and Greedy-MDS methods coupled with the global alignment kernel, the LHS method and the quantization method coupled with the FPCA dimension reduction method.

References

[1] K. K. Vu, C. D'Ambrosio, Y. Hamadi, L. Liberti, Surrogate-based methods for black-box optimization, International Transactions in Operational Research 24 (2017) 393–427. doi:https://onlinelibrary.wiley.com/doi/pdf/10.1111/itor.12292.

[2] T. J. Santner, B. J. Williams, W. I. Notz, The design and analysis of computer experiments, Springer, New York, NY, 2003.
URL https://doi.org/10.1007/978-1-4757-3799-8

[3] M. McKay, R. Beckman, W. Conover, A comparison of three methods for selecting values of input variables in the analysis of output from a computer code, Technometrics 21 (1979) 239–245.

[4] L. Pronzato, Minimax and maximin space-filling designs: some properties and methods for construction, Journal de la Société Française de Statistique 158 (1) (2017) 7–36.

[5] O. Teymur, J. Gorham, M. Riabiz, C. J. Oates, Optimal quantisation of probability measures using maximum mean discrepancy, in: AISTATS, Vol. 130, 2020, pp. 1027–1035. arXiv:2010.07064.

[6] V. R. Joseph, E. Gul, S. Ba, Maximum projection designs for computer experiments, Biometrika 102 (2) (2015) 371–380. arXiv:https://academic.oup.com/biomet/article-pdf/102/2/371/9642501/asv002.pdf, doi:10.1093/biomet/asv002.
URL https://doi.org/10.1093/biomet/asv002

[7] A. Costa, G. Nannicini, RBFOpt: An open-source library for black-box optimization with costly function evaluations, Mathematical Programming Computation 10 (4) (2018) 597–629.
URL https://doi.org/10.1007/s12532-018-0144-7

[8] J. Müller, C. A. Shoemaker, R. Piché, SO-MI: A surrogate model algorithm for computationally expensive nonlinear mixed-integer black-box global optimization problems, Computers & Operations Research 40 (5) (2013) 1383 – 1400. doi:https://doi.org/10.1016/j.cor.2012.08.022.
URL http://www.sciencedirect.com/science/article/pii/S0305054812001967

[9] P. Z. G. Qian, Sliced latin hypercube designs, Journal of the American Statistical Association 107 (497) (2012) 393–399.
URL http://www.jstor.org/stable/23239678

[10] J. Pelamatti, L. Brevault, M. Balesdent, E.-G. Talbi, Y. Guerin, Efficient global optimization of constrained mixed variable problems, Journal of Global Optimization 73 (2019) 583–613. arXiv:1806.03975.
URL https://doi.org/10.1007/s10898-018-0715-1

[11] J. Pelamatti, L. Brevault, M. Balesdent, E.-G. Talbi, Y. Guerin, Bayesian optimization of variable-size design space problems, Optimization & Engineering 22 (2020) 387–477. arXiv:2003.03300.
URL https://doi.org/10.1007/s11081-020-09520-z

[12] S. Graf, H. Luschgy, Foundations of quantization for probability distributions, Springer, 2007.

[13] A. Smola, A. Gretton, L. Song, B. Schölkopf, A Hilbert space embedding for distributions, in: Algorithmic Learning Theory, Vol. 4754, Springer, 2007, pp. 13–31.

[14] Y. Chen, M. Welling, A. Smola, Super-samples from kernel herding. Proceedings of the twenty-sixth conference on uncertainty in artificial intelligence, 2010, pp. 109–116.

[15] S. Mak, V. R. Joseph, Support points, The Annals of Statistics 46 (6A) (2018) 2562–2592.

[16] T.-T. Tran, D. Sinoquet, S. Da Veiga, M. Mongeau, An Adapted Derivative-Free Optimization Method for an Optimal Design Application with Mixed Binary and Continuous Variables, in: 6th International Conference on Computer Science, Applied Mathematics and Applications, ICCSAMA 2019, Hanoï, Vietnam, 2020. doi:10.1007/978-3-030-38364-0.
URL https://hal.archives-ouvertes.fr/hal-02494138

[17] T. T. Tran, D. Sinoquet, S. D. Veiga, M. Mongeau, Derivative-free mixed binary necklace optimization for cyclic-symmetry optimal design problems, Optimization & Engineering 28.
URL https://doi.org/10.1007/s11081-021-09685-1

[18] J. Franco, Planification d'expériences numériques en phase exploratoire pour la simulation des phénomènes complexes, Ph.D. thesis (2008).

[19] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, A. Smola, A kernel two-sample test, The Journal of Machine Learning Research 13 (1) (2012) 723–773.

[20] A. Gretton, K. Fukumizu, C. H. Teo, L. Song, B. Schölkopf, A. J. Smola, A kernel statistical test of independence, in: Advances in neural information processing systems, 2008, pp. 585–592.

[21] Y. Li, K. Swersky, R. Zemel, Generative moment matching networks, in: International Conference on Machine Learning, Proceedings of Machine Learning Research, 2015, pp. 1718–1727.

[22] F. Hickernell, A generalized discrepancy and quadrature error bound, Mathematics of Computation 67 (221) (1998) 299–322.

[23] L. F. South, M. Riabiz, O. Teymur, C. Oates, Post-processing of MCMC (2021).

[24] J. Yang, Q. Liu, V. Rao, J. Neville, Goodness-of-fit testing for discrete distributions via Stein discrepancy, Vol. 80 of Proceedings of Machine Learning Research, PMLR, Stockholmsmässan, Stockholm, Sweden, 2018, pp. 5561–5570.
URL `http://proceedings.mlr.press/v80/yang18c.html`

[25] C. Guo, F. Berkhahn, Entity embeddings of categorical variables, arXiv preprint arXiv:1604.06737.

[26] Y. Zhang, S. Tao, W. Chen, D. W. Apley, A latent variable approach to Gaussian process modeling with qualitative and quantitative factors, Technometrics 62 (3) (2020) 291–302.

[27] J. Kruskal, Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis, Psychometrika 29 (1964) 1–27.
URL `https://doi.org/10.1007/BF02289565`

[28] C. Van Den Berg, J. P. R. Christensen, P. Ressel, Harmonic analysis on semigroups: Theory of positive definite and related functions, Vol. 100, Springer Science & Business Media, 2012.

[29] F. Hutter, L. Xu, H. H. Hoos, K. Leyton-Brown, Algorithm runtime prediction: Methods & evaluation, Artificial Intelligence 206 (2014) 79 – 111. `doi:https://doi.org/10.1016/j.artint.2013.10.003`.
URL `http://www.sciencedirect.com/science/article/pii/S0004370213001082`

[30] L. Wu, I. E.-H. Yen, S. Huo, L. Zhao, K. Xu, L. Ma, S. Ji, C. Aggarwal, Efficient global string kernel with random features: Beyond counting substructures (2019). arXiv:1911.11121.

[31] M. Lange, D. Zühlke, O. Holz, T. Villmann, S.-G. Mittweida, Applications of l_p-norms and their smooth approximations for gradient based learning vector quantization., in: ESANN, Citeseer, 2014, pp. 271–276.

[32] S. Nanty, C. Helbert, A. Marrel, N. Pérot, C. Prieur, Sampling, metamodeling, and sensitivity analysis of numerical simulators with functional stochastic inputs, SIAM/ASA Journal on Uncertainty Quantification 4 (1) (2016) 636–659.

[33] M. Cuturi, J. Vert, O. Birkenes, T. Matsui, A kernel for time series based on global alignments, in: 2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07, Vol. 2, 2007, pp. II–413–II–416. doi:10.1109/ICASSP.2007.366260.

[34] Z. M. Munoz, D. Sinoquet, 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 (2020) 310–341.

[35] M. Moustapha, Conception robuste en vibration et aéroélasticité des roues aubagées de turbomachines, Ph.D. thesis, Université Paris-Est Marne la Vallée (2009).
URL https://tel.archives-ouvertes.fr/tel-00529002v2/document

[36] L. Luksan, J. Vlček, Test problems for nonsmooth unconstrained and linearly constrained optimization, Institute of Computer Science, Academy of Sciences of the Czech Republic. Technical report VT798-00.

[37] A. Neumaier, Neumaier's collection of test problems for global optimization (Retrieved in May 2014) 1–15.
URL http://www.mat.univie.ac.at/~neum/glopt/my_problems.html

[38] L. Dixon, G. Szego, The global optimization problem: An introduction, In: Dixon, L., Szego, G. (eds.) Towards Global Optimization, North Holland (1975) 1–15.

[39] J. Moré, S. M. Wild, Benchmarking derivative-free optimization algorithms, SIAM Journal on Optimization 20 (2009) 172–191. doi:https://doi.org/10.1137/080724083.

[40] G. Pagès, Introduction to optimal vector quantization and its applications for numerics, ESAIM: proceedings and surveys 48 (2015) 29–79.
URL https://hal.archives-ouvertes.fr/hal-01034196

[41] M. R. El Amri, C. Helbert, O. Lepreux, M. M. Zuniga, C. Prieur, D. Sinoquet, Data-driven stochastic inversion via functional quantization, Statistics and Computing 30 (3) (2020) 525–541. doi:10.1007/s11222-019-09888-8.
URL https://hal-ifp.archives-ouvertes.fr/hal-02291766

[42] M. R. El Amri, C. Helbert, M. Munoz Zuniga, C. Prieur, D. Sinoquet, Set inversion under functional uncertainties with joint meta-models, preprint (2021).
URL https://hal-ifp.archives-ouvertes.fr/hal-02986558