Master thesis report, SUPAERO

On adapting the Super-Efficient Global Optimization solver to handle mixed-variables, with applications in aircraft design.

Génie Mathématique et Modélisation
Modélisation Mathématique et Numérique, INSA Toulouse

Master II - Recherche Opérationnelle, ENAC,UPS

Paul Saves
Supervisors: J. Morlier & Y. Diouane (ISAE) and N. Bartoli & T. Lefebvre (ONERA)

02.03.2020 - 25.09.2020

Contents

List of Figures

Acronyms

AMIEGO A Mixed Integer Efficient Global Optimization.
ANOVA ANalysis Of VAriance.

B&B Branch and Bounds.
BLUP Best Linear Unbiaised Predictor.
BO Bayesian Optimization.

Cat-EGO Categorical Efficient Global Optimization.
CeRAS Central Reference Aircraft data System.
COBYLA Top Constrained Optimization BY Linear Approximation.
CS Compound Symmetry kernel.
CW Category-Wise kernel.

DACE Design and Analysis of Computer Experiments.
DoE Design of Experiments.

EGO Efficient Global Optimization.
EI Expected Improvement.
ESE Enhanced Stochastic Evolutionary.

FAST Future Aircraft Sizing Tool.
FAST-OAD Future Aircraft Sizing Tool-Overall Aircraft Design.
FoM Figure of Merit.

GA Genetic Algorithm.
GB Gradient-based.
GCS Generalized Compound Symmetry kernel.
GMHL Garrido-Merchán & Hernández-Lobato algorithm.
GP Gaussian Process.
GPS Generalized Pattern Search.
GRF Gaussian Random Field.

He_HS Heteroscedastic Hypersphere kernel.
Ho_HS Homoscedastic Hypersphere kernel.

KPLS Kriging with Partial Least Square.
KPLSK Kriging with Partial Least Square + Kriging.

L-BFGS Limited-memory Broyden Fletcher Goldfarb Shanno algorithm.
LHS Latin Hypercube Sampling.
LOO Leave One Out.
LS Least Squares.

MAB Multi-Armed Bandit.
MAC Mean Average Chord.
MDA Multidisciplinary Design Analysis.
MDO Multidisciplinary Design Optimization.
MIMOS Multiple Infill via a Multi-Objective Strategy.
MLE Maximum likelihood estimation.
MoE Mixture of Experts.
MSE Mean Square Error.
MTOW Maximum Take-Off Weight Brearkdown.
MVP Mixed Variable Programming.

NOMAD Nonlinear Optimization by Mesh Adaptive Direct Search.

OAD Overall Aircraft Design.
OEN One-Exchange Neighborhood.

PCA Principal Component Analysis.
PDF Probability Density Function.
PLS Partial Least Square.
PoF Probability of Feasibility.
PSD Positive SemiDefinite.

RBF Radial Basis Function.
REML Restricted Estimation by Maximum of Likelihood.
RMSE Root Mean Square Error.
RMTS Regularized Minimal-energy Tensor-product Splines.
RSM Response Surface Model.

SBO Surrogate Based Optimization.
SEGO Super Efficient Global Optimizer.
SEGOMOE Super Efficient Global Optimizer with Mixture Of Experts.
SLSQP Sequential Least Squares Programming.
SMAC Sequential model-based opti-mization for general algorithm configuration.
SMT Surrogate Models Toolbox.
SO-MI Surrogate Optimization-Mixed Integer.
SPM Stochastic Process Model.

TLAR Top Level Aircraft Requirements.
TPE Tree-structured Parzen Estimator Approach.

UCB Upper Confident Bound.

Abstract

Recent advances in aircraft design and multidisciplinary formulation have led to new workflows involving many disciplines, ranging up to high levels of fidelity codes. However, there is still a lack of information about the different available optimization methods and their efficiency depending on the problem properties.

Aeronautics design is especially multidisciplinary and, in this context, some integer or categorical variables could arises such as the number of engine, the shape of the tail of the type of the wingtip devices. To tackle this class of problem, some mixed integer multidisciplinary design optimization algorithms have recently been developed. The framework of this internship is limited to low-dimension (less than 50) mixed integer optimization and its application to aeronautic design. The problem will be treated as an expensive black-box that we try to optimize with as few evaluations as possible.

Firstly, we present an overview of the different types of methods that exists to solve these problems. Then, to illustrate this optimization framework, a method based on continuous optimization is presented and was implemented in an open-source library. Finally, a benchmark of functions was proposed to validate the method and a preliminary result was then treated on a reference model of an A320 aircraft design problem.

Chapter 1

Introduction

1.1 Description of the host laboratory

This internship was realized at the Institut supérieur de l'aéronautique et de l'espace (ISAE-SUPAERO) and partly at the Office National d'Etudes et de Recherches Aérospatiales (ONERA) in Toulouse, France. I was working for the research federation between ENAC, ISAE-SUPAERO and ONERA in the team XOAD working on Multidisciplinary Design Optimization. The supervisors of this internship were Joseph Morlier (ISAE/DMSM), Youssef Diouane (ISAE/DISC), Nathalie Bartoli (ONERA/DTIS) and Thierry Lefebvre (ONERA/DTIS).
ISAE-SUPAERO is an engineering school, it is one of the world leader in higher education in the field of aerospace engineering. ONERA is the French national aerospace research centre. It is a public establishment with industrial and commercial operations, and carries out application-oriented research to support enhanced innovation and competitiveness in the aerospace and defense sectors. Both institute are managed by the french ministry of armed force.

At ISAE-SUPAERO, I was mostly posted in the Department of Complex Systems Engineering (DISC), I was also partly involved with the Department of Mechanics, Structures and Materials (DMSM). The DISC departement of ISAE-SUPAERO develops knowledge in mathematics and computer science for the aerospace industry. In education as in research, it is interested in models, methods and tools to control the behavior and the performances of complex systems. This complexity may come from the multiphysics or multiscale nature of the systems, their dynamic behavior or their connected and distributed structure.

At ONERA, I was posted in *Département de Traitement de l'Information et Systèmes* (DTIS), Unité Méthodes Multidisciplinaires, Concepts Intégrés (M2CI). M2CI takes advantage of the applications-methods duality in the field of aerospace vehicle design for:

- On the one hand, developing methods inspired by the needs of applications and validated on representative cases of these needs.
- On the other hand, proposing and assessing innovative integrated concepts for aircraft or launch vehicles, the definition of which is improved by the use of appropriate design methods.

The developed methods (optimization algorithms, Multidisciplinay Design Optimization (MDO) formulations, surrogate models, uncertainties management) aim at increasing the number of disciplines covered, improving multi-fidelity exploration of the design space, and quantifying uncertainties in a multidisciplinary framework. From the application viewpoint, the goal is to promote innovative concept ideas (e.g., hybrid-electric aircraft, reusable winged launch vehicles), from initial exploration to expert assessment for the benefit of the sector's manufacturers, including collaborative approaches at national or European level.

1.2 Context and objectives of the internship

In the context of the European AGILE 4.0 project (https://www.agile4.eu/, 2019-2022), ISAE-SUPAERO and ONERA proposed an internship related with the application of numerical optimization methods to solve aerospace engineering problems. The AGILE 4.0 project is the AGILE (https://www.agile-project.eu/, 2015-2018) follow-up project that intended to develop the next generation of aircraft multidisciplinary design and optimization processes, which target significant reductions in aircraft development costs and time to market, leading to cheaper and greener aircraft solutions.

The internship is proposed in collaboration with ONERA (the French aerospace lab). AGILE 4.0 significantly extends AGILE's scope by adding manufacturing, maintenance, and certification aspects and extending the aircraft product optimization to the entire life cycle and addressing the extensive aeronautical supply chain. The extension to production and certification domains challenges the optimization strategies currently in use in the aeronautical world. Therefore one of the ambitions of AGILE 4.0 is to enhance the AGILE optimization strategies with the capabilities to address a larger number of heterogeneous disciplines, considering multiple optimization criteria and handling discrete variables such as a choice of architecture, or a choice of materials.

In the context of this collaborative project, ONERA and ISAE-SUPAERO proposed to extend an in-house Bayesian optimization (BO) solver to handle mixed continuous/integer/categorical design variables. The high level target of AGILE is to obtain a significant reduction in development costs of aircraft through the implementation of a more competitive supply chain at the early stages of design. This will be reached by practicing research on the project's four technical objectives.

Notwithstanding the availability of powerful software systems to integrate complex computational design processes, today there is a lack of quantified knowledge on how optimization workflows involving many disciplines, ranging up to high levels of fidelity codes, should be set up in the most effective and efficient way. To this purpose, the first objective of AGILE is the structured development of advanced multidisciplinary optimization techniques and their integration, reducing the convergence time in aircraft optimization.

Today's advanced analysis codes and software tools are mostly discipline-specific and well understood by disciplinary experts. However, the operation of the system of tools as a whole and the interpretation of the results are additional challenges in the collaboration between the disciplinary specialists, and the aircraft generalists. Therefore, the second objective of AGILE is the structured development of processes and techniques for efficient multisite collaboration in the overall design teams.

Mastering complex systems highly depends on the exploitation of knowledge. Besides the interaction of experts, the smart handling of data, information and knowledge using information technologies offers high potential. Thus; the third objective of AGILE is the structured development of knowledge enabled information technologies to support interdisciplinary design campaigns. The fourth objective of AGILE is to develop and publish an Open MDO Test Suite, enabling the access to the project technologies by other research activities, and providing a reference database for future aircraft configurations research.

ONERA and ISAE developed solvers for efficient global optimization based on surrogate models.
The objective of this internship is threefold:

- First, we aim at implementing new mixed integer surrogate models. These models can be used for optimization but also for Multidisciplinary Design Analysis, sensitivity analysis...
- Second, the generalization of Efficient Global Optimization (EGO) for expensive mixed integer black-box optimization.
- Third, this internship can be seen as a preparatory my PhD thesis under the supervision of the same tutors.

Concretely, this internship allows me:

- To understand the state of the art and the background.
- To do a review of the state-of-the-art methods and include them in my comparison.
- To implement mixed integer metamodels.
- To implement mixed integer Bayesian optimization techniques.
- To test these models and run optimization problems.
- To test a practical aerospace test case.

The rest of this report is organized as follows. First, I will investigate Bayesian optimization existing methods as well as the background of the current study. My aim is to give a detailed review of the state-of-the-art methods in order to include them in my numerical comparison. Then, implementation details regarding the mixed integer metamodels for Bayesian optimization will be given. Last, to the validation of the implemented approach will be exposed using both academic test cases as well as an aerospace application.

Chapter 2

State of the art

2.1 The Aircraft Optimization Framework

Aeronautic design is a complex system and, as such, is highly multidisciplinary [1]. Therefore, to find an acceptable aircraft, many disciplines should be balanced together as Aerodynamics, Structure, Geometry, Performance, Weight, Dynamic Stability,... This part is called Multidisciplinary Design Analysis (MDA). Once we have a good balanced aircraft design after MDA, it can be optimize by modyfing inputs. This optimization process is called Multidisciplinary design optimization (MDO) [2]. Therefore, when we modify the Design variables inputs to find a better aircraft according to an objective, we should balanced it another time to have a new balanced aircraft that we expect to be better according to an objective [3].

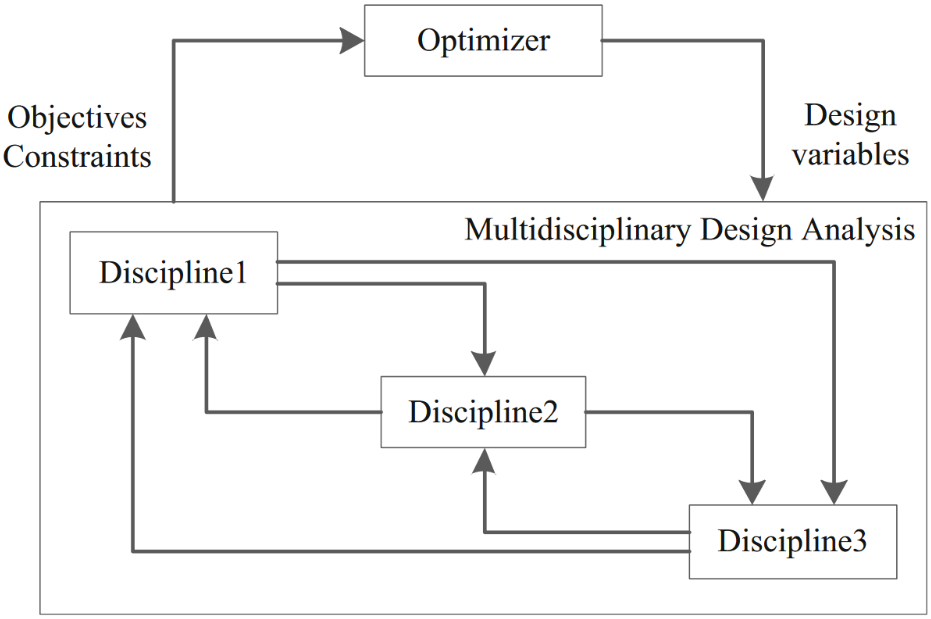


Figure 2.1: Multidisciplinary Design Analysis and Optimization diagram [4]

Figure 2.1 shows the MDO/MDA process. After, in this report, the MDA process will be considered by the MDO process as an expensive black-box to optimize [5].
Let $f(x)$ be the value of objective function evaluated through the MDA black-box and $\forall i \in \{1 \dots q\}$ $g_i(x)$

be the values of the q constraints functions. Assuming the variables x are continuous, the optimization problem is:

$$\begin{aligned} \underset{x \in \mathbb{R}^k}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & \forall i \; g_i(x) \leq 0 \end{aligned} \tag{2.1}$$

To begin with, we will talked about Efficient Global Optimization (EGO) to solve the problem (2.1) without constraints (excepted when it is explicitly mentioned that we are talking about the constrained case). The goal of EGO is to solve Equation (2.1) with the fewest possible evaluation points $f(x)$ of an expensive black-box [6].

In a second part, we will talked about the extension of EGO to mixed integer problem where the variables in the problem could also be integers or categorical variables.

2.2 Efficient Global Optimization

In Efficient Global Optimization (EGO) of Black-Box Functions [6], Jones et al. were among the first to adapt a whole model of bayesian kriging to optimize expensive to evaluate black boxes, globally and without differentiation (based on preliminary results of Močkus [7]). Stochastic global optimization methods are used to adjust response surfaces models (RSM) from a set of initial point picked up from a specific pattern (Design of Experiments) [8, 9, 10].

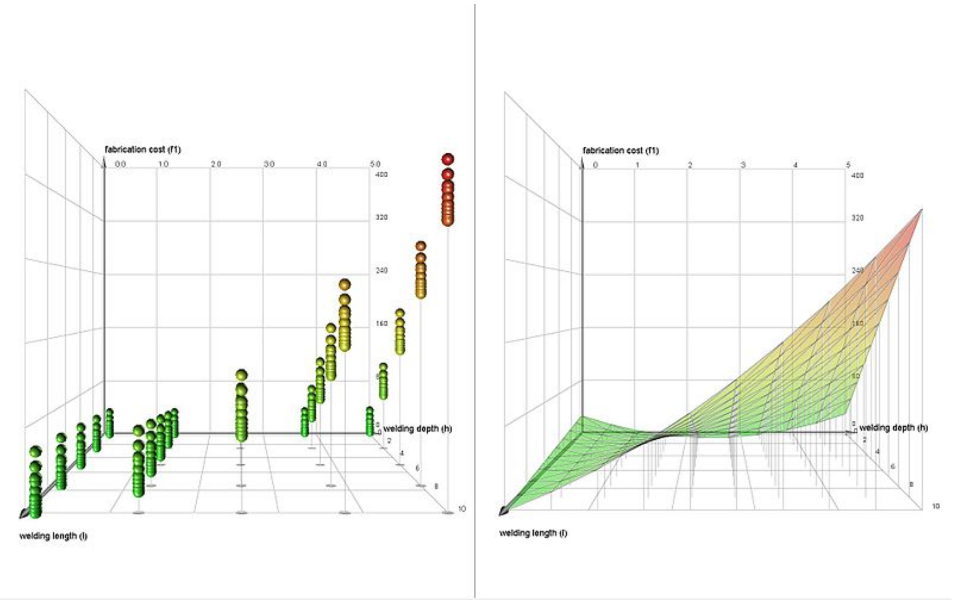


Figure 2.2: A response surface model (right) built from a given Design of Experiments (left).

Figure 2.2 represents a model explaining the values of chosen points. This is a 2D example of a model with second-degree polynomial for a augmented full factorial Design of Experiments (DoE) of 250 points.

Consequently, the model is exact on the known point but approximative everywhere else. Thereafter, one will seek to construct a model interpolating our DoE based on stochastic processes. This will

allow us to obtain a mean response hypersurface with a pointwise estimation of the variance. The calibration of stochastic processes to a Design of Experiments is called kriging.
We will consider that our unknown black-box functional "f(.)" is a realization of an underlying distribution of zero mean (or $\mu, \hat{\mu}, \hat{\mu}(x)$) and of covariance function $k(x, x')$: $f(.) \sim GP(0, k(., .))$.

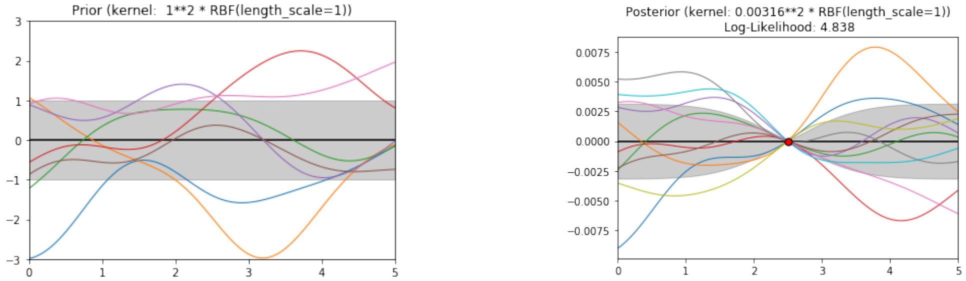


Figure 2.3: 1D Kriging with Gaussian Random Field

A Gaussian Random Field (GRF) of zero mean and standard deviation 1 (the model a priori) is represented on the left of **Figure 2.3**. On the right, it was added an intial DoE consisting of one central point, to show how the information is incorporated into the model. The point being known, it corresponds to a zero standard deviation and to a small one on its neighbourhood. Because a priori, the process is $\mathbf{C}^{\infty}$, it was chosen to use a square exponential covariance kernel $k(x, x')$ in this example.
At every given iteration, a new surrogate model (our metamodel) of the black box is calculated with a known DoE. Then, one wants to estimate the best new point to evaluate by taking into account all the information a priori to converge as fast as possible to the real optimum of the black-box.
This new point is the point which should probably gives the best improvement a priori. This point value is evaluated and then incorporate into a new metamodel. The hyperparameters which characterize and define the model and especially the form of the covariance kernel are thus updated at each iteration until convergence. The bayesian optimization made from these Gaussian processes is called Efficient Global Optimization (EGO).

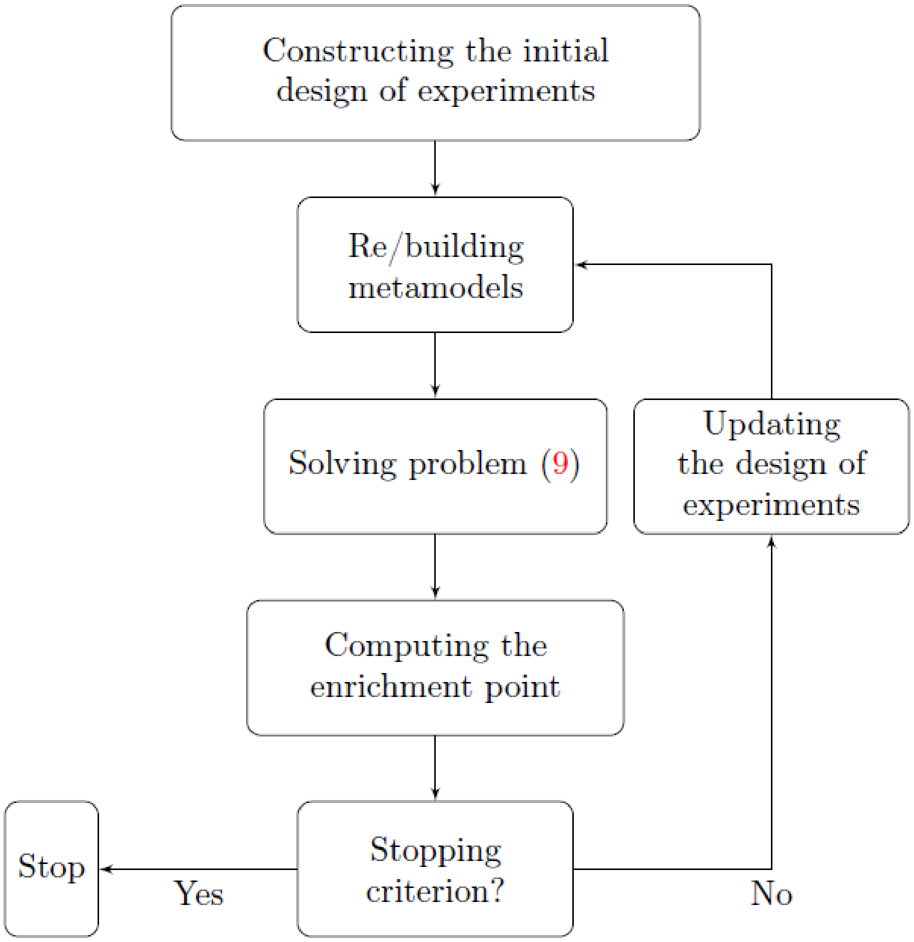

Constructing the initial design of experiments
Re/building metamodels
Solving problem (9)
Updating the design of experiments
Computing the enrichment point
Stopping criterion?
Stop
Yes
No

Figure 2.4: Diagram showing the architecture of the EGO algorithm

Figure 2.4 comes from [11] and describes the general functioning of the EGO. In the following sections, we will explore each part of this general process of EGO.

2.2.1 Design of Experiments

Choosing a good initial DoE is important for the convergence of the optimization. A good DoE will give a good surrogate model of the functional and one seeks to have the least possible points that fill the best the search space. For instance, the Latin Hypercube Sampling (LHS) is widely used for stochastic optimization since it is simple and efficient.

In [11], they used the Enhanced Stochastic Evolutionary (ESE) [12]. In the latter, Jin & Sudjianto proposed an rapid algorithm to optimize a criterion (maximin distance, entropy, CL2..) which can be chosen by the user. Depending on the a priori knowledge of the problems, engineers can choose some others properties (like orthogonality) or design (full factorial, partial factorial,...).

2.2.2 Surrogate Model

Assuming n already evaluated points of a deterministic function in $\mathbb{R}^k$. The vector $\mathbf{x}^i = (x_1^i, ..., x_k^i) \in \mathbb{R}^k$ corresponds to the ith point, $i \in \{1, .., n\}$. The stochastic model [13] can be rewrite as:

$$y(\mathbf{x}^i) = \sum_h \beta_h f_h(\mathbf{x}^i) + \epsilon^i \in \mathbb{R}$$

with ϵ^i is the error term between the true black-box and the model that should be estimated.
In practice, there are three main categories of kriging:

- Simple kriging: $y(\mathbf{x}^i) = \mu + \epsilon^i$, where μ is known as in **Figure 2** with $\mu = 0$.
- Ordinary kriging where μ is constant and unknown.
- Universal kriging where $\mu = \sum_h \beta_h f_h(\mathbf{x}^i)$ is polynomial and unknown.

If y is continuous and knowing that the functions f are selected to be continuous, we deduce that uncertainty terms ϵ are continuous. The closer the data x^i and x^j, the more the errors ϵ^i and ϵ^j are correlated.
Let the weighted distance be:

$$d(x^i, x^j) = \sum_{h=1}^{k} \Theta_h |x_h^i - x_h^j|$$

Thus, the covariance kernel being square exponential: $k(x, x') = \exp(-\frac{1}{2}.d(x, x')^2)$ The parameters Θ_h are the inverse correlation lengths in the h direction. Id est that the more the variable h causes significant changes in the functional, the shorter the correlation length: a displacement of $\frac{1}{\Theta_{h_1}}$in the direction h_1 will have the same impact as a displacement of $\frac{1}{\Theta_{h_2}}$ in the direction h_2. This corresponds to the variable "length_scale" in the **Figure 2**.

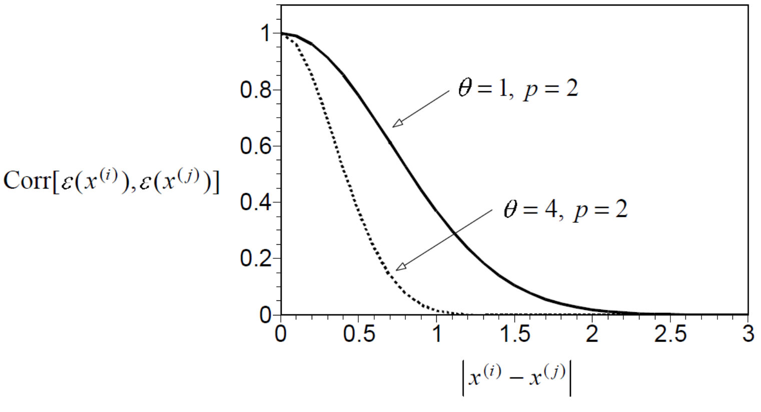


Figure 2.5: Plot of two correlation functions for two activity level Θ [6]

Figure 2.5 comes from [6] and describes the kernel described above ($p = 2$: square). Hence, we obtain a metamodel taking into account the dependencies between the normally distributed random variables $\epsilon(x^i)$. Note that they considered the Pearson correlation: $Corr(X, Y) = \frac{Cov(X,Y)}{\sqrt{Var(X).Var(Y)}}$.
We consider the covariance kernel and not the correlation one as the first can be deduced from the latter (we assume constant variance hyperparameter σ^2).

The Stochastic Process Model (SPM) that they build with this kernel is the DACE (Design and Analysis of Computer Experiments) which has k+2 parameters, what corresponds to ordinary kiging: Θ_k, μ, σ.
Let y be the vector of evaluated values at the points x, and let R be the error correlation matrix between inputs points $R_{ij} = Corr(\epsilon(x^i), \epsilon(x^j))$. The hyperparameters Θ_k are estimated by maximum likelihood of the samples. Then by maximizing the following likelihood

$$L = \frac{1}{(2\pi)^{n/2}(\sigma^2)^{n/2}|R|^{1/2}} \exp - \frac{(y - \mathbb{1}\mu)'R^{-1}(y - \mathbb{1}\mu)}{2\sigma^2}$$

and knowing the hyperparameters, they estimate μ and σ by:

$$\hat{\mu} = \frac{\mathbb{1}'R^{-1}y}{\mathbb{1}'R^{-1}\mathbb{1}} \text{ and } \hat{\sigma}^2 = \frac{(y - \mathbb{1}\mu)'R^{-1}(y - \mathbb{1}\mu)}{n}$$

By substituting these expressions in L and maximizing it, it is possible to recover an estimation for Θ_k and therefore R.
one also seeks to estimate the expected response between points. To do this, the BLUP (Best Linear Unbiaised Predictor) can be used. Let $r_i(x^*) = Corr(\epsilon(x^*), \epsilon(x^i))$, then (BLUP) $\hat{y} = \hat{\mu} + r'R^{-1}(y - \mathbb{1}\hat{\mu})$ and the DACE interpolate data. The Mean Squared Error (MSE) is

$$s^2(x^*) = \sigma^2[1 - r'R^{-1}r + \frac{(1 - \mathbb{1}'R^{-1}r)^2}{\mathbb{1}'R^{-1}\mathbb{1}}]$$

which gives a zero error on data.

Models can be validated using leave-one-out strategies or more generally cross-validation. For example, Jones et al. calculated the number of standard deviations of error between a known value and its prediction by the DACE generated by a leave-one-out strategy [11, 14].
In the case of variance component estimation, the original data set is replaced by a set of contrasts calculated from the data, and the likelihood function is calculated from the probability distribution of these contrasts, according to the model for the complete data set. In particular, Restricted Estimation by Maximum of Likelihood (REML) is used as a method for fitting linear mixed models. In contrast to the earlier maximum likelihood estimation, REML can produce unbiased estimates of variance and covariance parameters [15].

2.2.3 Optimization strategy

Now that we are able to build inexpensive surrogate models, we need to handle the associated uncertainties on those models. In fact, by using the minimum of the current surrogate model , one may get trapped by local minimum as the surrogate is an approximation and can be very uncertain in some regions. This criterion is called Surrogate Based Optimization (SBO). The Bayesian paradigm consists in searching for the best point to evaluate by taking into account the estimation of the uncertainties at every possible point. In theory, we want the point that will have the biggest probability to improve our minimum incumbent.

The first method is to use a Figure of Merit (FoM) consisting on a trade-off functions between:

- Local research which searches into the optimal information by exploiting the minimum zone the metamodel.

- Global research which searchs into the missing information by exploring the zone where we could find a better minimum (a potential minimum is a small amount of standard deviation under the mean prediction).

The first Figure of Merit, widely used in EGO is the Expected Improvement (EI). Let f_{min} be the current incumbent. The EI criterion is given by

$$\text{EI}(x) = \mathbb{E}\left[\max\left(f_{min} - y(x), 0\right)\right].$$

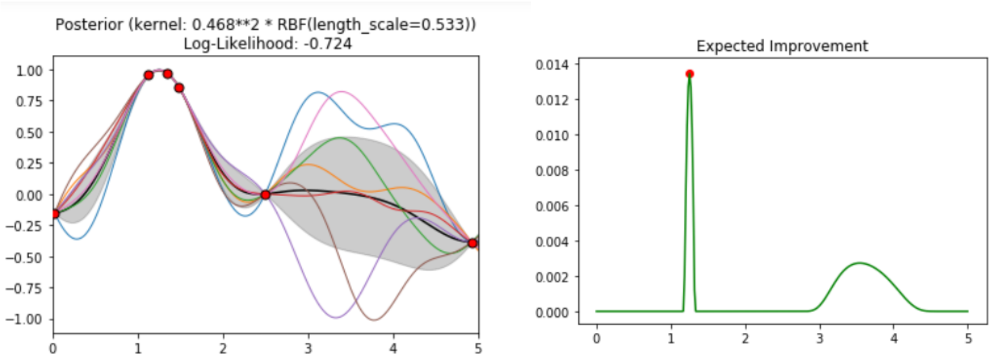


Figure 2.6: DoE (red points) with Surrogate model (Left), Figure of Merit-EI (Right)

As you can see in **Figure 2.6** where we are searching for the maximum, we are choosing exploitation and not exploration because the exploitation peak is higher on the Figure of Merit (the EI on the right).
We talked about SBO and EI but we can talk about other Figure of Merit criteria. Among the widely used we have Upper Confident Bound ($\mu - 3\sigma$), WB2s ($s.EI - \mu$) [11, 14] or the Generalized EI [16].

When a Figure of Merit is defined to be used, the optimization problem becomes:

$$\min_{x \in \mathbb{R}^k} \quad \text{FoM}(GP(x|x^0, .., x^n, f(x^0), .., f(x^n)))$$

A lot of different algorithms can be found in the literature like Evolutionary Algorithms, Gradient-Based algorithms, COBYLA, Simplex, Interior Points, Branch and bounds ...

Another method for global optimization is to use NOMAD (Nonlinear Optimization by Mesh Adaptive Direct Search) [17].

Algorithm 1: Optimizing with NOMAD

```
Result: Minimum of a function
Given a distance and a step $\Delta$, build a mesh grid. Let $\epsilon$ be a threshold value.
Let $y*$ be the current minimal value the algorithm has found.
while $\Delta > \epsilon$ do
    Search better points near the incumbent on the mesh grid according to the mesh (SEARCH
     step).
    if the SEARCH step improve the minimum: $y_{SEARCH} < y*$ then
        $y* \leftarrow y_{SEARCH}$.
        $\Delta \leftarrow \Delta^+$ with $\Delta^+ > \Delta$. If successful, we take a bigger step $\Delta$ for the mesh at the next
         iteration.
    else
        Use a given strategy to explore new zones of the mesh (POLL step).
    end
    if the POLL step improve the minimum: $y_{POLL} < y*$ then
        $y* \leftarrow y_{POLL}$
        $\Delta \leftarrow \Delta^+$ with $\Delta^+ > \Delta$. If successful, we take a bigger step $\Delta$ for the mesh at the next
         iteration.
    else
        $\Delta \leftarrow \Delta^-$ with $\Delta^- < \Delta$. If not successful, we take a smaller step $\Delta$ to tighten the mesh
         and have more possible points to explore at the next iteration.
    end
end
```

2.2.4 Stopping criterion

The new point to compute determined, its value is evaluated through the black-box to optimize and integrated into the DoE. Then, a surrogate model is again built and this whole operation is iterated until reaching a stopping criterion fixed beforehand [18] . For example a long period wiout finding a better point, a progression of less than one percent several times in a row, a Δ bigger than a threshold,... However, the main stopping criterion usually imposed is given by a maximum number of iterations related to a computational budget and not by a convergence criterion.

2.3 Efficient Global Optimization with constraints

To tackle the constraints of the problem (2.1), we have several possible approaches. In NOMAD [19, 20], they use a filter to only keep the feasible region. Another approach is to penalize the Expected Improvement (for example like a Lagrangian) [21]. We can integrate the penalization directly into the Figure of Merit by, like the expected improvement, creating a metamodel but for the expected violation of the constraints [11, 14]. All this methods and some others are summarized in [22].

2.4 Dealing with integer and categorical variables: A method overview

Previously, we have seen how to treat continuous variable bur in order to resolve mixed problems, we need a method to optimize discrete variables taking values in $\mathbb{Z}$ or categorical variables taking

values in a finite set. Integer variables are in an ordered ring but categorical variables have not a mandatory complete order structure, neither a dyadic operation.
For example let z be a integer variable. If z=2, z+1=3>z and let u be categorical such that $u \in \{banana, apple\}$ 'banana' and 'apple' can not be compared and we have no signification for 'banana'+'apple'.
Let u be categorical variable with 4 levels such that $u \in \{Red, Green, Blue, Cyan, Yellow\}$
In label encoding, we are replacing our categorical variable by a integer variable whose possible values are $X \in [1, .., nb\_categories]$, e.g, $X = 1$ is $x =' Red'$ and $X = 4$ is $x =' Cyan'$.
In one-hot encoding, we are replacing our categorical variable by $nb\_categories$ binary variables $(X1, ..., X_{nb\_categories})$ with $X_i = 1$ if $X = i$, e.g. $X = (1, 0, 0, 0, 0)$ is $x =' Red'$ and $X = (0, 0, 0, 1, 0)$ is $x =' Cyan'$.
In the following, x will be the continuous variables, z the integer (order relation) and u the categorical variables.

A categorical variable can take different values called levels. Categories are possible arrangement of levels. Example: shape and color are 2 categorical variables, square, circle, rhombus are 3 shape levels, green, blue are 2 color levels and *blue square*, *blue circle*, *blue rhombus*, *green square*, *green cirle* and *green rhombus* are the 2*3=6 categories.

In EGO, one wants to build a surrogate model which approximate the back-box. The inputs being mixed integer, we need to find new strategies to build mixed surrogate model. Then, we need new optimization strategies based on these models to converge quickly.
When one has an optimization goal, it is necessary to choose a model more specialized for it but this optimization-oriented models are not precise outside the minimal zone. This is why it is interesting when we are after two objectives, precise metamodels and efficient optimization, to have different approaches.
Different strategies from literature are explained in the following. In every case, we have a different type of models and a different type of optimization. Some methods are more oriented toward the model construction and some other are more about the optimization.

2.4.1 Building a mixed model with many sub-models

A Mixed Integer Efficient Global Optimization (AMIEGO) [23] is a mixed EGO optimizer that constructs the global mixed surrogate model as a combination of models for the integers and models for the continuous variables (not for categorical ones). For every mixed function $f(x, z)$, $g(x, z)$ (objective or constraint) they build a different set of metamodels. From now, let's focus on a mixed metamodel for f, the objective function.

Algorithm 2: Building a metamodel with AMIEGO

Result: Mixed Surrogate Model

- Generate a DoE for the integer variables z^k and another DoE for the continuous variables denoted x^k
- $\forall z_i \in z^k$, for each point of this integer DoE, build a continuous model (M_C) over the continous space based on the continuous DoE x^k knowing that the integer variables are equal to the current integer $M_{C_i} = M_C|z = z_i$
- $\forall z_i \in z^k, y_{I_i} = \min_x(M_{C_i})$. For every point in the integer DoE, the corresponding integer output is the minimum according to the continuous model for this integer value (the SBO).
- The integer metamodel is then build, as a continuous model, based on the minima from the different continuous models (z_i^k, y_{I_i}).

For the dimension not to scale to quickly with the dimension, the kriging model is coupled with Partial Least Square regression (see Section 3.1.1) [24].

To optimize the metamodel, they compute a Figure of Merit over the integer model. Then, they used the new infill point z according to the Figure of Merit as a constant and built a new continuous model. The minimum of this new continuous model gives us the y output. The couple (z, y) is then add in the integer DoE and the integer metamodel is refined.
The Figure of Merit used is a generalized EI under constraints. In fact, in 2019 [25], they used a new strategy called Multiple Infill via a Multi-Objective Strategy (MIMOS). This multi-objective formulation seeks to simultaneously maximize three objective functions: the original expected improvement function (balance between exploration and exploitation), the distance from an existing evaluated point (pure exploration), and a particular case of the generalized expected improvement function (see [16]) that reduces to the probability of improvement (pure exploitation). They used a binary-coded multi-objective evolutionary algorithm that also satisfies the integer constraints of the design variables and generates a 3-D Pareto front of non-dominated designs.

The main drawback of this method for model buiding is that the integer model being only a model according to the minimum in the continuous dimension, it can gives us no information about the space elsewhere, far from the continuous minimum. However, this is a really good result for our second pursued objective as the model concentrates over the minimum and then is efficient for the optimization.
Unfortunately, AMIEGO does not tackle problem with categorical variable and is limited in size by the explosion in the number of models to build when there is a lot of continuous variables together with integer ones.

2.4.2 Building a mixed model with continuous relaxations

Garrido-Merchán & Hernández-Lobato (GMHL) [26] proposed a method that uses continuous relaxations of integer and categorical variables to deal with categorical and integer-valued variables in a

BO framework. They distinguishes 3 approaches:

• The Naive approach consists in solving the continuous relaxation of the problem and then to project afterwards to the closer admissible point for evaluation. Naive approach can select several times the same point to evaluate or do not converge to the correct solution because of the rounding which do not guarantee the Figure of Merit to be maximized on the integer or categorical value with the biggest value.

• The basic approach proposed consists in doing the same but rounding into the wrapper that evaluate the black box besides of doing it outside.

• In the best approach proposed by GMHL, they computed discretized gaussian processes by replacing every point where the mean and the variance are evaluated by the nearest valid point before computing the estimate means and correlations so they obtain more realistic results. The consequence of this method is that the Euclidean distance between the points that lead to the same variable configuration becomes zero. This enforces maximum correlation between the function values at those input points under the GP, obtaining the expected constant behavior.

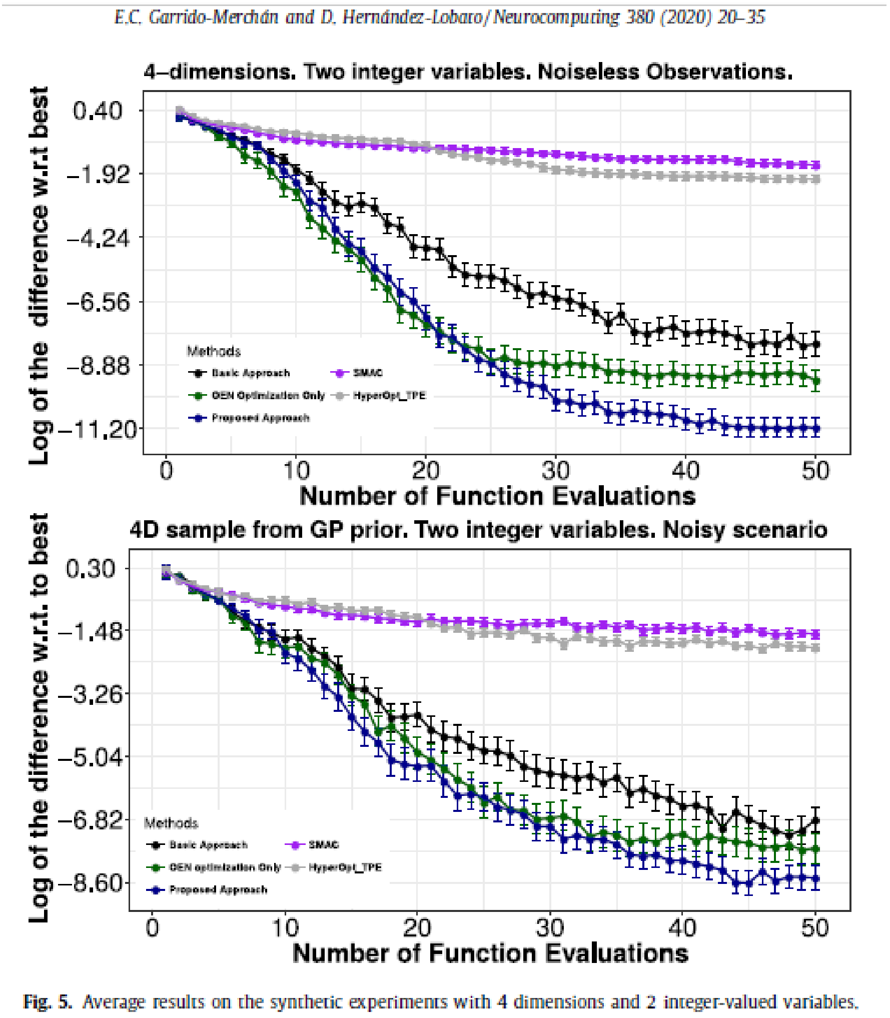
As they used a flat acquisition function (continuous by part Figure of Merit), they optimize via a block coordinate ascent methodology which iterates between optimizing non-real variables (integer-valued and categorical) and real variables by using the grid and the L-BFGS method to optimize all real variables. Then, the One-Exchange Neighborhood (OEN) strategy is used to optimize the transformed integer and categorical variables. The OEN strategy is a greedy method [27, 28].
The algorithm can be summarized in the following way:

Algorithm 3: Building a metamodel with GMHL

Result: Mixed Surrogate Model

- Relax continuously every integer dimension.
- $\forall u_j$ (categorical variables), $\forall i \in L_{u_j}$, $\Omega = \Omega \times [0,1]$ We add a variable u_{j_i} in [0,1] for every level of every categorical variable (one-hot encoding).
- Generate a continuous DoE in Ω with admissible values.
- Create a continuous model with the relaxed inputs that gives us a mean μ and a covariance kernel k.
- Project the continuous model to have a new constant by part model where every point has the value of the closer physically admissible point.
- $\mu'(x) = \mu(T(x))$ and $k'(x, x') = k(T(x, x'))$ with $T(x)$ being the closer value of x in the search space.
- $T(x)$ is x for continuous variable. $T(z)$ rounds z to the closer integer. $\forall i \in L_{u_j}$, $k := \arg\max_{u_{j_i}} \mu(u_{j_i}),\ T(u_{j_k}) = 1, T(u_{j_{i \neq k}}) = 0$, the chosen level (set to 1) is the one with the biggest relaxed value.

To compare their results, they introduced two BO methods: SMAC (Sequential model-based optimization for general algorithm configuration) with randoms forests [29] and TPE (Tree-structured Parzen Estimator Approach) with a generative approach [30]. These methods do not use gaussian processes.



E.C. Garrido-Merchán and D. Hernández-Lobato / Neurocomputing 380 (2020) 20–35

Fig. 5. Average results on the synthetic experiments with 4 dimensions and 2 integer-valued variables.

Figure 2.7: GMHL variants optimizations methods compared with SMAC and TPE [26]

Figure 2.7 compares these different methods in a set of test cases (there are other comparisons in the paper [26]).
GMHL method is really simple and can treat all types of variables but as the space is relaxed continuously, we often explore and optimize a really big space with a lot of flat parts because of the rounding and moreover, the dimension of the space highly increases with the combinatorics when there is several integer or categorical variables. We can do as in AMIEGO and use KPLS to reduce dimension but the strategy is not adapted for high dimension.

2.4.3 Building a model with mixed covariance kernels

Group kernels for Gaussian process metamodels with categorical inputs [31], introduce the notion of categorical kernels. Let $w = (x, u)$ such that $x = (x_1, .., x_I)$ the continuous variables and $u =$

$(u_1, .., u_J)$ the categorical ones. Here, we consider an integer to be a categorical variable. For example, $z \in [-2, 2]$ will be considered as $u \in \{$ "-2","-1","0","1","2"$\}$.
Then the product space is $\mathscr{D} = U \times \prod_{j=1}^{J} \{1, .., L_j\}$ with $U \subset \mathbb{R}^I$ and L_j is the number of levels the jth categorical variable can take and denote m the dimension of the categorical space $m = \prod_{j=1}^{J} L_j$. As before, we denote k the covariance kernel of the gaussian process, $k(w, w') = Cov(\epsilon(w), \epsilon(w')$.
Mixed kernels are defined as combination of continuous and discrete ones:

- Product: $k(w, w') = k_{cont}(x, x')k_{cat}(u, u')$.
- Sum: $k(w, w') = k_{cont}(x, x') + k_{cat}(u, u')$.
- ANOVA: $k(w, w') = (1 + k_{cont}(x, x'))(1 + k_{cat}(u, u'))$.

The question then comes down to constructing a valid kernel on a finite set, which is a positive semidefinite (PSD) matrix, the discrete covariance kernel is then a matrix T. Instead of creating a big matrix of size m for each possibility called Category-Wise kernel (CW), we could combine 1-dimensional kernels with the operations denoted * (Sum, Product or Anova) which preserve positiveness.
For example: $k_{cont}(x, x') = k^1_{cont}(x_1, x'_1) * .. * k^1_{cont}(x_I, x'_I)$ and $k_{cat}(u, u') = [T_1]_{u_1, u'_1} * .. * [T_J]_{u_J, u'_J}$ For Integer, it is proposed to use a transformation on the space and then to apply a discrete kernel.
Pellamati et al. distinguish the homoscedastic case where T is a constant diagonal (constant variance through levels) and the general heteroscedastic case where the variance can vary between dimensions. Usually, PSD matrices are decomposed in $T = PDP^T$ with D diagonal and P an orthogonal matrix decomposed via classic transform (Cayley transform, Eulerian angles, Householder transformations, Givens rotations...) or $T = LL^T$ with Cholesky decomposition. In a homoscedastic case, the columns of L have the same norm and represent points on a sphere in $\mathbb{R}^{L_j}$. For heteroscedastic matrices, one can uses a hypersphere decomposition of the lower triangular matrix L.
An other way to extract hyperparameters without the dimension explosion of the full CW kernel is the compound Symmetry (CS) kernel where it is assumed common correlation for all levels [32]. The CS covariance matrix treats equally all pairs of levels, which is an important limitation, especially when L is big. More flexibility is obtained by considering groups of levels.
Roustant et al. separate the levels of a categorical level into G groups of non constant sizes and consider a PSD matrix T by block.

$$T = \begin{pmatrix} W_1 & B_{1,2} & ... & B_{1,G} \\ ... & ... & ... & ... \\ B_{G,1} & ... & B_{G,G-1} & W_G \end{pmatrix}$$

where the diagonal blocks W_g contain within-group covariances, and the off-diagonal blocks $B_{g,g'}$ are constant matrices containing between-group covariances. one wants the matrices W to be PSD and then $W_g - \bar{W}_g J_g$ to be PSD with $\bar{W}_g$ being the mean of W_g and J_g the matrix of one of the size of the gth group. This modelized the Generalized Compound Symmetry (GCS) matrices.
Let u a categorical variable partitioned in G groups, g/u denotes that the variable u belongs to the group $\mathscr{G}_g$. Considering the following hierarchical Gaussian model:

$$\eta_{g/u} = \mu_g + \lambda_{g/u}, g = 1, .., G, u \in \mathscr{G}_g$$

$\mu \sim \mathscr{N}(0, B^*)$ is the group effect and $\lambda_{g/.} \sim \mathscr{N}(0, W_g^*)$ is the within-group levels effects and it is assumed independence between these variables. $W_g^* = Cov(\lambda_{g/.}|\bar{\lambda}_{g/.} = 0)$ and up to centering conditions on $\lambda_{g/.}$, μ_g is the mean of group g. Hence, B^* is interpreted as the between group means covariance. Similarly, $\lambda_{g/.}$ is the within-group effect around the group mean. This justifies the notations B^* and W_g^*.

Roustant et al. give two theorems and a method for building a generalized compound symmetry matrix. See Appendix B.

In Efficient global optimization of constrained mixed variable problems [33], they used product of continuous and discrete kernels like before [31]. In order to ensure that the chosen kernel is valid, it is necessary for the covariance function to be symmetric and PSD.

Pellamati et al. proposed several kernels models, the trivial one is the Category-Wise kernel (CW), which is a correlation matrix with $\prod_{j=1}^{J} L_j(L_j + 1)/2$ parameters (due to symmetry).

As mentioned previously, the hyperparameters are calculated via an hypersphere decomposition first applied within the scope of mixed variable GP by Zhou et al. [34]. The Figure of Merit used is the Infill Criterion, which is the product between the Expected Improvement and the Probability of Feasibility (PoF). The latter is optimized by relying on a mixed continuous/discrete Genetic Algorithm (GA) similar to the one presented by Stelmack et al. [35].

A more reasonable way is to consider only the small matrices between levels and then do the product what was called combining 1-dimensional kernels.

As in Roustant et al., they decompose the kernel into $k_{cat}(u, u') = \prod_{j=1}^{J} [T_j]_{u_j, u'_j}$.In this case, the heteroscedatic kernel by level with an hypersphere decomposition (He_HS) has $\sum_{j=1}^{J} L_j(L_j + 1)/2$ parameters and the homoscedatic one (Ho_HS) has $\sum_{j=1}^{J} L_j(L_j - 1)/2$ parameters. The compound symmetry model has 2 L_j parameters with a variance and a covariance parameter by level. One particular case of this simple model allows us to rewrite the kernel as a mixed one between continuous and discrete space with the Gower distance [36].

Thus obtaining the mixed kernel:

$$k_{Gower\ CS}(w, w') = \sigma^2 exp\left[-\sum_{i=1}^{I}\theta_i\left(\frac{\frac{|x_i - x'_i|}{\Delta x_i}}{I + J}\right)^{p_i} - \sum_{j=1}^{J}\theta_{j+I}\left(\frac{\mathbb{1}_{u_j \neq u'_j}}{I + J}\right)^{p_{j+I}}\right]$$

Logically He_HS should be the best model but covariance between categories are badly calculated because they are calculated over the whole continuous space. In fact, the more the model is general and complex, the more it is efficient and precise but the more it has hyperparameters. Hence, for a given budget, having a lot of hyperparameters is not a good choice because the bayesian estimated values of these parameters are less precise. Nevertheless, having few parameters means oversimplified models which have well-estimated parameters but are not realistic. In practical cases, Ho_Hs tends to be the one which seems to works the best.

The procedure belows summarized the usage of a kernel in the building of a metamodel once we have chosen and built the structure of our categorical/discrete kernel (GCS, CS, Ho_Hs, He_Hs, CW,...):

Algorithm 4: Building a metamodel with group kernels

Result: Mixed Surrogate Model

- Generate a DoE for the continuous variable x^k and a DoE for the categorical variables (integer are considered as categorical) u^k.
- Only for the integer variables in u^k, we have, $[T_j]_{z_j,z_j^2} = k_j(F_j(z), F_j(z^2))$. So, before calculating the matrix (Step 3), we transform the input with a metafunction (hyperparameters to optimize) which is continuous and strictly increasing (can be linear by part, quadratic,...). This function distorts the inputs to take into account the order relation that exists between integers before calculating the covariances.
- $k_{cat}(U, U^2) = \prod_{j=1}^{J} [T_j]_{u_j,u_j^2}, T_j \in M_{L_j}$. The categorical kernel is the product of the matrices of "sub-kernels" represented on correlation matrices form for every categorical variable.
- Build a mixed kriging model, with covariance kernel $k(w, w^2) = k_{cont}(x, x^2).k_{cat}(u, u^2)$

The correlations between the categories are calculated over the whole continuous space and independently. This can lead to badly evaluated categorical correlations for the minimal continuous zone. It is a global model contrary to AMIEGO which is a good model only on the minimal regions. By choosing a more complete problem, we have a better representation but as we have a lot of hyperparameters, there are badly optimize whereas with few hyperparameters they are way better optimized but lead to a too simple model. Together with this trade-off between models, if we choose the Generalized Compound Symmetry kernel, then, there is a lot of clustering work to choose the block in the matrices. In a concrete case, it is almost impossible to predict which model will be the best.

2.4.4 Adapting Nonlinear Optimization by Mesh Adaptative Direct Search to Mixed Variable Programming

Pattern search algorithms are derivative-free direct search algorithms [37]. In Audet [19], they filtered the non feasible and non-optimal solution, according to the best feasible incumbent [17]. They used the Mixed Variable Programming (MVP) optimization algorithm, a pattern search for bound constraints. Originally, they only guaranteed convergence to a stationary point when the objective function was continuously differentiable in the continuous variables, but Abramson's thesis has extended this result using the Clarke calculus (1990) as in Audet and Dennis [17] for the continous case [38]. In Abramson, linear constraints for mixed variable programming are treated using the idea presented in Lewis and Torczon (2000). It involves setting the objective function value equal to +inf when one of them is violated, and when the current iterate is sufficiently close to the boundary, the polling directions must contain directions tangent to the boundary [39].

The Audet-Dennis Generalized Pattern Search (GPS) algorithm for bound constrained mixed variable optimization problems is extended to problems with general nonlinear constraints by incorporating a filter, in which new iterates are accepted whenever they decrease the incumbent objective function value or constraint violation function value. Additionally, the algorithm can exploit any available derivative information (or rough approximation thereof) to speed convergence without sacrificing the flexibility often employed by GPS methods to find better local optima. In generalizing existing

GPS algorithms, the new theoretical convergence results reduce seamlessly to existing results for more specific classes of problems. While no local continuity or smoothness assumptions are made, a hierarchy of theoretical convergence results is given by Kokkolaras, Audet and Dennis, in which the assumptions dictate what can be proved about certain limit points of the algorithm [38].

The MVP code was used to solve a thermal insulation problem [38, 39]. In this problem, there is a temperature variable (continuous) for each intercept and a distance (continuous) between every two following intercepts. The discrete variables are: the number of intercept (integer) and the type of insulator (categorical) between Nylon(N), Teflon(T) and fiber-glass Epoxy(E).
The discrete neighbourhood are manually defined as such:

- Any of the existing intercepts and the insulator to its left are removed.
- A new intercept together with an insulator to its right are added.
- The type of insulator between two intercepts is changed.

Then the NOMAD MVP is operated with this sequence:

```
Algorithm 5: Optimizing with NOMAD MVP
Result: Minimum of a mixed function

Given a distance and a step $\Delta$, build a mesh grid. Let $\epsilon$ be a threshold value.
Let $y^*$ be the current minimal value the algorithm has found.
while $\Delta > \epsilon$ do
    • Do the search step as usual on the mesh grid.
    • If no new minimum is found. Do a poll step on every continuous dimension.
    • If no new minimum is found. Do a discrete poll on user-defined discrete neighbourhood.
    • If no new minimum is found. Do an extended poll: when the minimum found on a discrete
      neighbourhood is close to the incumbent, do a continuous poll on this discrete value.
    • If no new minimum is found: $\Delta \leftarrow \Delta^-$.
    • When a new minimum is found: $\Delta \leftarrow \Delta^+$ and update the incumbent $y^*$.
end
```

This MVP needs to have good prior knowledge to manually define neighbourhoods. Although it could work really well, it could be long to execute in high dimension.

2.4.5 An heuristic for surrogate Optimization

Surrogate Optimization-Mixed Integer(SO-MI) [40] is an heuristic to optimize a mixed integer function but only with a defined distance, so it is not really adapted to categorical variables.

The algorithm works as follows:

Algorithm 6: Optimizing with SO-MI

Result: Minimum of a mixed function

- Create 4 groups of candidates by perturbing (continuous variables) or (discrete variables) or (both) of the incumbent or by uniform sampling over the domain.
- Calculate a Score for every candidate in his group.
- Select the best candidate of each group for enrichment.
- Add the four new points into the surrogate model.

Let W be the score value. K_j is the j-th candidate of one of the four groups K. The exploration score is defined with the minimal euclidean distance D from a sampling point to its closer already sampled point. $D(K_j) = \min_{i=0..n} ||K_j - x_i||$ Considering the K-th group, $D_{min} = \min_j D(K_j)$, $D_{max} = \max_j D(K_j)$ Hence, the distance score of a sample relative to its group denoted V_D is $V_D = \frac{D_{max}-D(K_j)}{D_{max}-D_{min}} \mathbb{1}_{D_{min} \neq D_{max}} + \mathbb{1}_{D_{min}=D_{max}}$
Similarly, the exploitation score is defined with the estimated surrogate value Sf. $S_{min} = \min_j Sf(K_j)$, and thus, the surrogate response of a sample relative to its group is $V_R = \frac{Sf(K_j)-S_{min}}{S_{max}-S_{min}} \mathbb{1}_{S_{min} \neq S_{max}} + \mathbb{1}_{S_{min}=S_{max}}$
Therefore, they defined the trade-off score value of a point as $W(K_j) = \omega_R V_R(K_j) + \omega_D V_D(K_j)$ with some rules for the weight parameters.
The weight at iteration k is ω^k.

- $\forall k, \ \omega_R^k \geq 0$ and $\omega_D^k \geq 0$
- $\forall k, \ \omega_R^k + \omega_D^k = 1$
- $\omega_D^0 = 1, \omega_R^0 = 1$, initially, we want to explore the search space.
- $\forall k, \ \omega_D^k > 0 \rightarrow \omega_D^k < \omega_D^k$, the more the model has point, the more one wants to exploit it.
- if $\omega_D^k = 0$ then $\omega_D^{k+1} = 1$. The weights cycle to avoid converging into a local minimum.

There is also a penalization based on the constraint violation for the constrained case.
Theorem: The algorithm SO-MI is asymptotically complete, i.e. assuming an indefinitely long runtime and exact computations, a global minimum of the optimization problem will be found with probability one.

This random heuristic is not really efficient in very large spaces because of the curse of the dimensionality which creates more and more distant points in high dimension. The random sampling strategy could not work efficiently in high dimension where we could prefer pseudo-random heuristic, for example. This heuristic does not work for the categorical variables and has several tuning parameters to optimize.

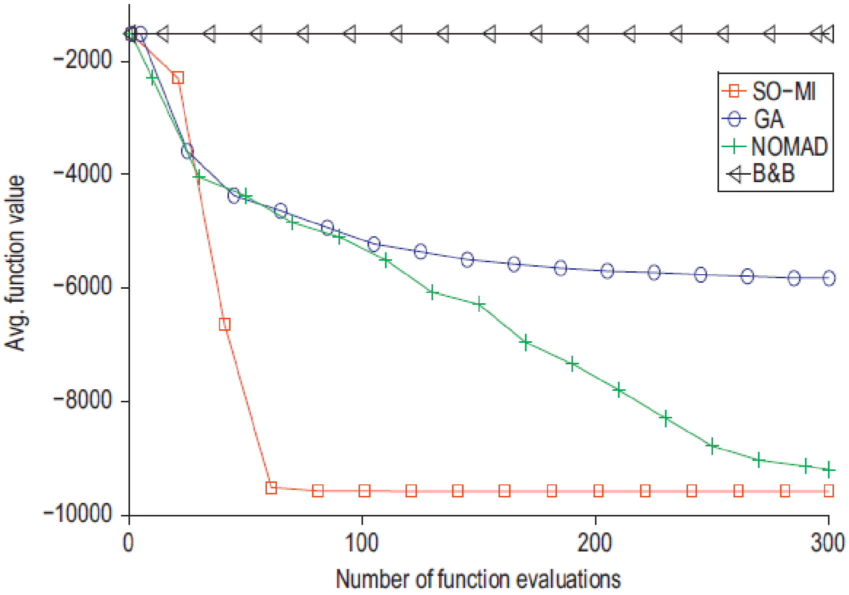


Fig. 2. Objective function value for feasible points averaged over 30 trials vs. number of function evaluations; test problem 12, box-constrained minimization problem.

Figure 2.8: SO-MI, GA, NOMAD and B&B optimizations on a dimension 15 case (with 5 integers) [40]

Figure 2.8 compare SO-MI with a Genetic Algorithm (GA), a Branch & Bound algorithm (B&B) [41] and NOMAD for integer. Other tests cases are presented on the paper [40]. In this particular case, the method performs really well. As expected, NOMAD is more efficient that non-adapted methods like GA or B&B.

2.4.6 Optimizing a mixed model with an adaptation of Nonlinear Optimization by Mesh Adaptive Direct search algorithm for a Figure of Merit optimization

The original Categorical Efficient Global Optimization (Cat-EGO) algorithm combines different methods [42]. In fact, Zuniga and Sinoquet have developed a new automatic heuristic for using NOMAD without physics a priori (cf. section 2.4.4) on the structure of the problem which is the core improvement but in their algorithm, they use the mixed NOMAD to optimize the Expected Improvement.

Algorithm 7: Optimizing with Cat-EGO

Result: Minimum of a mixed function

- As in NOMAD, start by separating the continuous, integer and categorical variables.
- For continuous and integer variables, do the same procedure as in a classic NOMAD because the neighbourhoods are well-defined, with an euclidean distance.
- When the ordered NOMAD SEARCH and POLL failed on the ordered variables x and z, do an extended poll on the categorical variables u. To do this, one has to generate with an heuristic a probability law Z_k at iteration k. The probability law is constructed as such:
  - Let $p_{k,i} = \mathbb{P}(Z_k = c_i)$ and U be the number of categories.
  - Z_0 is initialized as an equiprobability for changing to another category ($\mathbb{P}(Z_k^{Poll} = c_i | Z_k^{Current} = c_j, Z_k^{Poll} \neq c_j) = \frac{1}{U-1}$ so that each category is at the same "probability distance".
  - At each iteration, the probabilities of being in a known state $p_{k,i}$ are updated and then, the probabilities of changing to another state are defined as: $\mathbb{P}(Z_k^{Poll} = c_i | Z_k^{Current} = c_j, Z_k^{Poll} \neq c_j) = \frac{p_{k,i}}{1-p_{k,j}}$
  - The probabilities to be in a specific category at iteration k decomposed into $p_{k,i} = \alpha_k p^g_{k,i} + (1-\alpha_k) p^m_{k,i}$.
  - The exploration term is approximated by $p^{\hat{g}}_{k,i} = 1 - (\frac{n_{k,i}}{n_k})^{1/2}$ with n_k being the number of point known at iteration k and $n_{k,i}$ being the number of points known at iteration k in the category i. If we have evaluated no points in category i at iteration k, $p^g_{k,i} = 1$ and if we have evaluated only points in category i, $p^g_{k,i}$=0.
  - The exploitation term is approximated by $p^{\hat{m}}_{k,i} = 1 - \frac{S^R_{k,i}}{\sum\limits_{j=1}^{U} S^R_{k,j}}$ with $S^R_{k,i}$ being an approximation of the mean number of standard deviation improvement we can expect on the cumulative distribution function of the minimum of our surrogate knowing that $z = c_i$ (the exploitation term in the Expected Improvement).
- Let V_k be the realization of a simulated uniform distribution over [0,1]. And let $u = c_j$ (the current category is the j-th) if $u \neq c_1$ and $v \in [0, \frac{p_{k,1}}{1-p_{k,j}}] \rightarrow u := c_1$

  else $\forall i \in \{2..U\}, i \neq j$, if $v \in [\sum\limits_{l=0}^{i-1} \frac{p_{k,l}}{1-p_{k,j}}, \sum\limits_{l=0}^{i} \frac{p_{k,l}}{1-p_{k,j}}] \rightarrow u := c_i$

The space can now be explored in case of no prior information in NOMAD and we still have a good convergence with NOMAD but the heuristic can takes a lot of time to converge as SO-MI. This algorithm is less expensive in high dimension than a classic pure NOMAD as it combine different approaches.

The list of presented methods is not exhaustive, the goal was to introduce different types of approaches to deal with mixed integer to give an overview of the different ideas used. To cite one article not presented in this review, we can talk about Bandit-BO [43] which uses a Multi-Armed

Bandit (MAB) and, like for Cat-EGO (cf. section 2.4.6), deduces a probability law over the categories but by considering the minimum of every continuous model, like AMIEGO (cf. section 2.4.1).
We have presented totally different approaches and their functioning but to conclude on this section, we will summarize the characteristics of these approaches for our two objectives: the modelling and the optimization.
AMIEGO, GMHL, Group kernels, MVP, SO-MI and Cat-EGO refer to methods presented in the sections 2.4.1 to 2.4.6.

2.4.7 Models Summary

In this section we will summarize the interests of the methods for the construction of a metamodel. The first criterion that arises is *does the model can manage categorical variables?*
Some mixed EGO methods do not take into account the non ordered variables and therefore can not use them without a defined distance. This is the case for AMIEGO and SO-MI, for example. In the first case, there is no way to build a metamodel without distance and in the second case, the only possible method could be by random sampling what is totally not appropriate to this particular problem. One method which is not designed for categorical variable but can still managed them is MVP but in this case, one needs prior information to manually define categorical neighbourhood. In Group Kernel, the opposite problem arises because integers are treated as categorical variables but this problem is solved by adding a meta-function to correct the distance between the points. On the particular case of the Compound Symmetry, they found that the model is equivalent to the Gower Distance [36] which means that the natural distance between two categorical variables should be the relative Hamming distance defined as the number of different levels between two categories. Let u and u' be two categories corresponding to J categorical variables then $D_{Hamming}(u,u') = \sum\limits_{j=1}^{J} \frac{\mathbb{1}_{u_j \neq u'_j}}{J}$.
This distance is in $[0,1]$ and this is also the case in the two last methods, GMHL and Cat-EGO having adapted distance for categorical variables.
Cat-EGO is based on group kernels models and brings innovations only for the optimization like SO-MI, based on a Radial Basis Function (RBF) kernel. NOMAD can optimize a mixed integer problem but the only underlying metamodel is the continuous mesh grid and there is no mixed model.
Therefore, we have three types of models to approximate the relationship between categories:

- Models based on a probability law which define a transition probability between two categories (CAT-EGO).
- Models based on a covariance kernel with a geometric distance between categories (GMHL, MVP).
- Models based on a covariance kernel with a discrete matrix (Group Kernels).

In all the presented methods, only the Group kernels and GMHL can be used for building a metamodel. In the case of GMHL, the distance being continuous and the categorical variables being relaxed in $[0,1]$, we came back to an Hamming distance when evaluating the distance between categories.
Now, it could be interesting to distinguish models that could be used for optimization purpose.

2.4.8 Optimization Summary

In this section we will summarize the interests of the methods for the optimization of an expensive black-box.
The second question that interest us is *does the model can be used to efficiently optimize the underlying black-box ?*
As previously said, two steps NOMAD strategy or Figures of Merit can be used to have a criterion or a procedure that fulfill the global aspect of the optimization goal. We then have several methods to optimize these different criteria. For mixed integer, probability-based methods can also be used to choose between different discrete possibilities. Another strategy could be to optimize a model based on the optimization of a sub-model like in AMIEGO where the integer model is built by optimizing several other continuous models. Some classic solvers like evolutionary algorithm can also be used once the model is built, for example to optimize the Figure of Merit [33, 35]. To finish with, CAT-EGO and SO-MI are probability-based methods, CAT-EGO works with NOMAD to optimize the EI of the model and SO-MI uses four different strategies in parallel (an exploration by random sampling and three exploration with different perturbations). The efficiency of CAT-EGO seems to indicate that combining different approaches is relevant to optimize quickly.

For its simplicity, I have chosen to test GMHL and then to implement this particular method into SMT. This method using continuous relaxations, it can be optimized once the model is build with a classic continuous optimizer. Another possibility would have been to implement a group kernel variant, for example Compound Symmetry which have really few hyperparameters but then it would have been necessary to develop a relevant mixed optimization method. CAT-EGO, for example, seems really promising.

Chapter 3

Developed Algorithm

3.1 Models and optimizers used in the Surrogate Modeling Toolbox

The surrogate modeling toolbox (SMT) is a Python package developed by ISAE-Supaero, ONERA, NASA and the University of Michigan [44]. It contains a collection of surrogate modeling methods, sampling techniques, and benchmarking functions. This package provides a library of surrogate models that is simple to use and facilitates the implementation of additional methods. SMT is different from existing surrogate modeling libraries because of its emphasis on derivatives, including training derivatives used for gradient-enhanced modeling, prediction derivatives, and derivatives with respect to the training data. It also includes new surrogate models: kriging by partial-least squares reduction (KPLS, KPLSK) which scales in inputs and energy-minimizing spline interpolation (RMTS) which scales in training points. SMT is documented using custom tools for embedding automatically-tested code and dynamically-generated plots to produce high-quality user guides with minimal effort from contributors.
SMT library can be downloaded and installed from this link: https://www.github.com/SMTorg/smt.
The developments made during this internship were implemented in SMT and are now available on the official version with the previous link. The dedicated documentation can be found at https://smt.readthedocs.io/en/latest/_src_docs/applications/mixed_integer.html.

3.1.1 Kriging with Partial Least Square

Partial Least Squares (PLS) is a statistical method to analyze the variations of a quantity of interest with respect to underlying variables. In the same spirit as the Principal Component Analysis (PCA) which gives directions that maximize the inertia of data, PLS gives directions that maximize the variation of the quantity of interest. In [24], they used the PLS to define subspaces to make high-dimensional Kriging more efficient. The first step is to make an analysis on the dataset, define the number of principal components h we are interested in, then apply projections of datasets on the found principal components. Doing so, they proposed a solution to reduce the dimension of the problem while keeping a maximum of explicative data.
Therefore, Kriging with Partial Least Square regression (KPLS) consists into projecting the hyperparameters space into a smaller space of dimension h. The h first axis are called principal components and used to span this new small space of lower dimension and only h hyperparameters are optimized).

$$\prod_{l=1}^{nx} \exp\left(-\theta_l \left(x_l^{(i)} - x_l^{(j)}\right)^2\right), \qquad \prod_{l=1}^{nx}\prod_{k=1}^{h} \exp\left(-\theta_k \left(w_l^{(k)} x_l^{(i)} - w_l^{(k)} x_l^{(j)}\right)^2\right) \quad \forall\, \theta_l, \theta_h \in \mathbb{R}^+$$

Standard Gaussian correlation function **PLS-Gaussian correlation function**

Figure 3.1: Formulas from: https://smt.readthedocs.io/

3.1.2 Kriging with Partial Least Square+K

KPLSK is a KPLS-based model. As before, we project into the small space of dimension h spanned by principal components. Then we optimize the reduced number of hyperparameters in the small space [24]. Thereafter, the optimized reduced hyperparameters are projected back into the full original space and this projected reduced optimal are used as a starting point for a local optimization of all the hyperparameters.

The models KPLS and KPLSK could be easily coupled with GMHL. For instance to reduce the dimension. The criterion qEI can also be coupled with GMHL (cf. Appendix C).

3.2 Super Efficient Global Optimizer with Mixture Of Experts

Adaptive surrogate models have become more and more popular in engineering optimization problems. The SuperEGO (SEGO) algorithm with a mixture of experts model (MOE) well suited to high-dimensional problems is known as SEGOMOE [1, 11, 14, 24]. They start by constructing the best surrogate model by combining automatic clustering and best expert selection. Then, they approach the solution iteratively by balancing the exploration/exploitation phase. It leads to a restricted number of calls of time-consuming high-fidelity models (so called budget).

SEGOMOE was not used during this internship but would have been be interesting to work with for both constraints handling, as in Equation (2.1) and high-dimensional purposes.

3.3 Study and implementation of continuous relaxations in the Surrogate Modeling Toolboox to deal with integer and categorical variables

3.3.1 Continuous relaxation preliminary tests of variants and coding

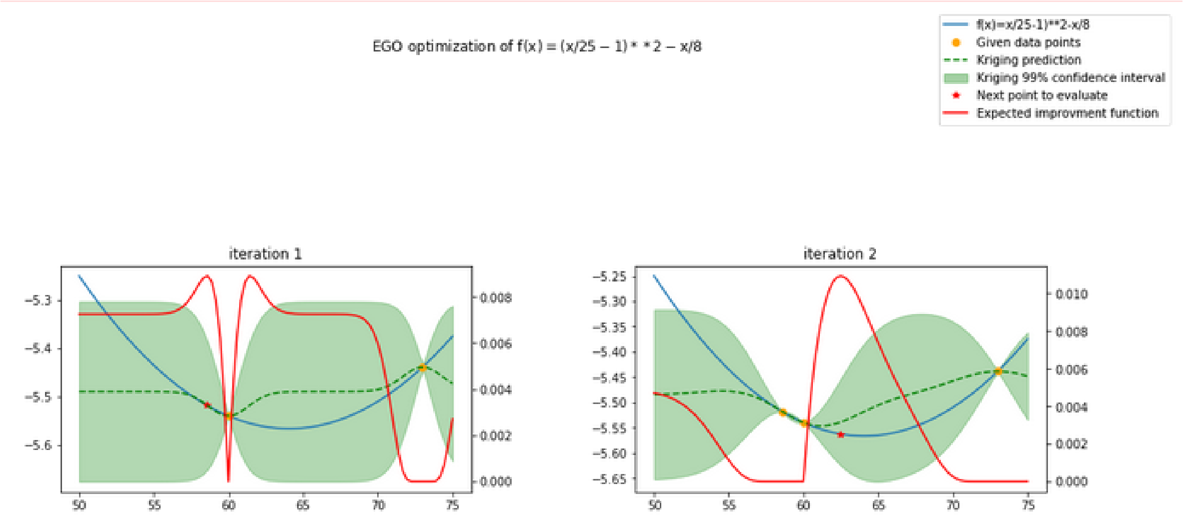


Figure 3.2: SMT optimization results

The left scale is for the function value and the right scale is for the Expected Improvement value.

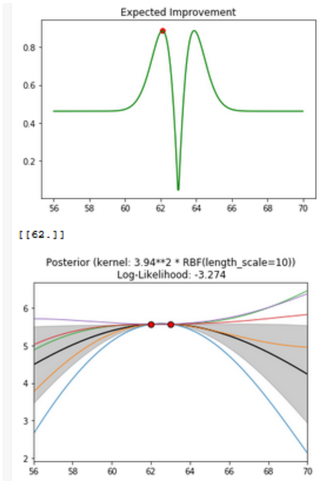
Figure 3.2 presents a step of EGO optimization of a continuous quadratic functional. The problem $\max_x EI(x)$ is solved using Sequential Least Squares Programming (SLSQP).

The first method I have implemented is what GMHL called Naive approach for integers. The In this method, we round to the nearest integer the result of an ego continuous relaxation, as only integers have a value through the black-box. The infill point is

$$x^* = \lfloor \max_x EI(\mu(x), \sigma(x)) \rfloor$$

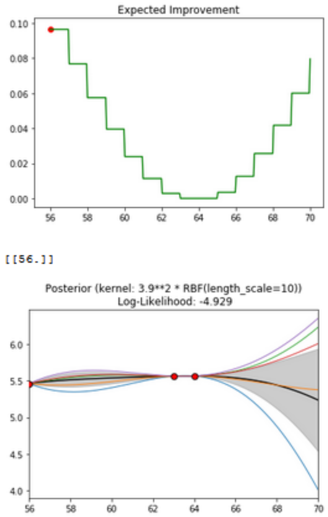
where $\lfloor \ \rfloor$ denotes the floor function, also called the greatest integer function. In a second approach, it was tested to round the expected improvement. In fact, when evaluating the next point to evaluate, as only integer have defined value, we should take, not the best point of the continuous relaxation but the best integer in term of Figure of Merit. Hence, for a given Figure of Merit, e.g EI, we optimize $FoM(floor(x))$. The problem with this method is that a continuous by part function is harder to optimize as we can't use gradient-based methods, in this case, the solver used is no longer Sequential least squares quadratic programming (SLSQP) but Constrained optimization by linear approximation (COBYLA). The infill point is

$$x^* = \max_{\lfloor x \rfloor} EI(\mu(x), \sigma(x))$$



```
[[62.]]
```

Figure 3.3: GMHL naive approach



```
[[56.]]
```

Figure 3.4: GMHL with rounded EI

Figure 3.3 presents the naive approach and **Figure 3.4** presents the method but with rounded expected improvement. The functional is the same as in **Figure 3.2** but with an integer variable $z \in [56, 70]$. We have in both case two points and we see the associated expected improvement and its maximum corresponding to the last point that was added to the model. The black line corresponds to the mean, the gray zone is 3 standard deviation around the mean. As we can see, with the rounded EI, we have an EI of 0 on the segment [63,64] which is coherent with the fact that we have already evaluated the two integers 63 and 64.

The best approach from GMHL is to compute discretized gaussian processes. To do this, we replace every point where we evaluate the mean and the variance by the nearest integer before computing the estimate means and covariances. The infill point is $x^* = \max_x EI(\mu(\lfloor x \rceil), \sigma(\lfloor x \rceil))$
The discrete gaussian processes are more relevant, the variances are better calculated and the continous expected improvement helps the gradient-based optimization algorithm, but the problem is that we can select a point already evaluated because we round after optimizing the Figure of Merit to find the best newer point. A workaround implemented was to use tunneling penalization [45]. With this, we penalized the Figure of Merit in the neighbourhood of already evaluated integers. At iteration i, the DoE is the k points we have already evaluate $x_{DoE} = (x_0, ...x_k)$

$$FoM_{penalized}(x) := \max(\ FoM(x) - \sum_{i=0}^{k} penalization(x, x_k), 0\ )$$

with $penalization(x, x_k) = \max(\ \frac{FoM(x) - FoM(x_k)}{||x - x_k||^{2\lambda}}, 0\ )$

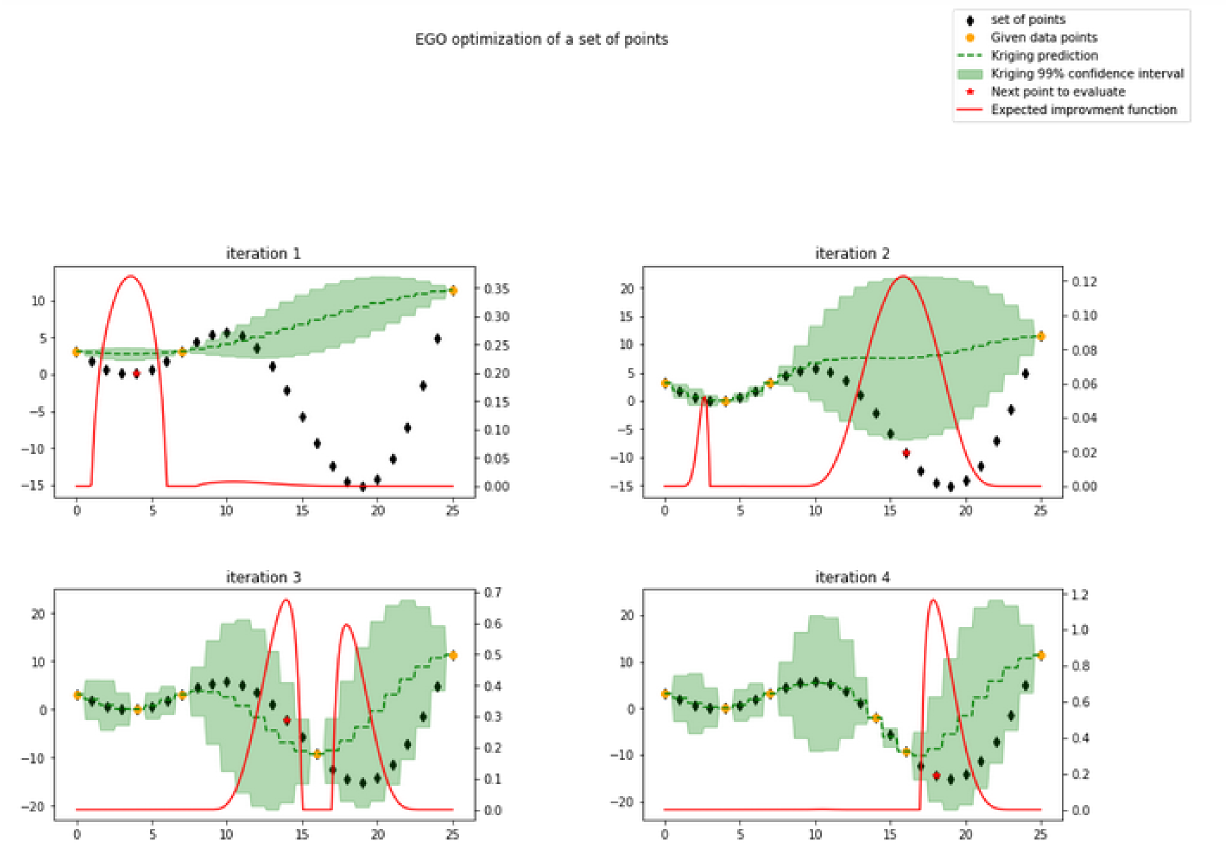


Figure 3.5: GMHL for integer with tunneling

Figure 3.3 shows 4 iterations of the best approach. Now, it is the model which is discrete and not the Expected Improvement we will optimize. And at iteration 3, you can clearly see the effect of the tunneling around $z = 16$ but the definition should give a continuous expected improvement. In this case, $\lambda = 2$.

For categorical variables, we will use a one-hot encoding. As before, we round the values of the prediction to one of the possible outcomes. The biggest component of the relaxed optimization will be set to 1 and the others to zero. For example, if (0.9,0.8,0.7) is the point to evaluate, we will use the prediction for the value (1,0,0). Therefore, the corresponding black-box objective is expected to be constant in those regions of the input space that correspond to the same configuration.

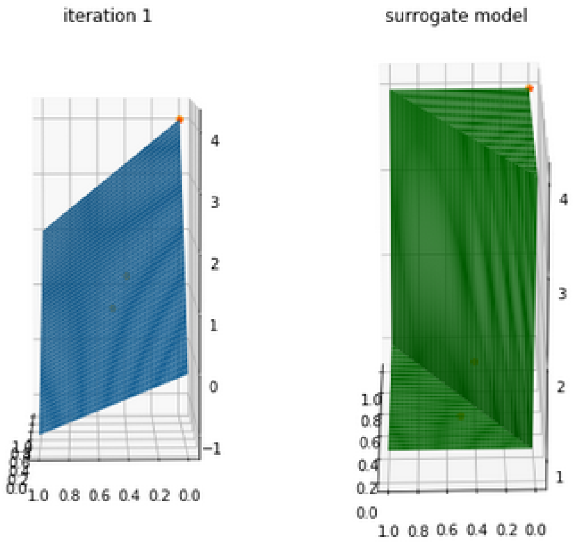


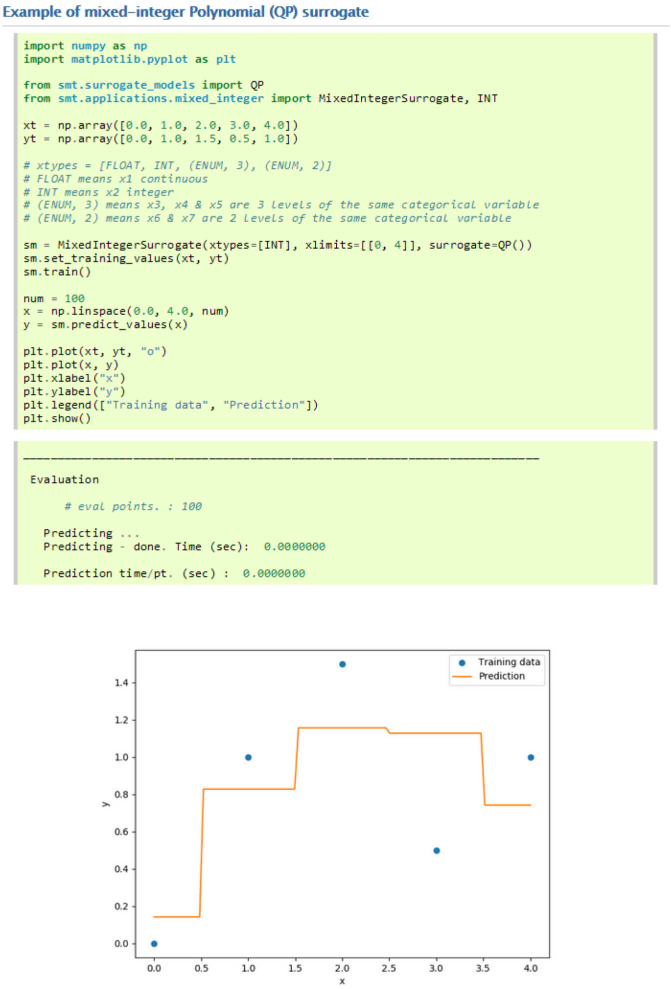
Figure 3.6: GMHL for categorical variables

Figure 3.6 shows what the rounding operation does on a 2D example. On the left, the blue model represents the continuous case and on the right is the rounded case. As we have a categorical variable, the only possible values are (1,0): we choose the first level or (0,1): we choose the second level. Therefore, we have an oblique separation when we round to the closer possible admissible value. We observe that the uncertainty is equal to zero after just collecting one observation that corresponds to the True value and just one observation that corresponds to the False value.

By testing the latter on different test cases, we notice that the penalization is too expensive for high dimension problems but the tunneling option is still available in SMT. This conclude the section of the investigation of the different continuous relaxation algorithms. In SMT, we will only implement the best approach called GMHL. This implementation is presented in the next section.

3.3.2 Implementation in the toolbox

Later on, I have implemented a first model in SMT based on these results.
The toolbox is built around the models. Hence, the first step was to build mixed surrogate models. To do this, we specified the type of every variable and then we relaxed them in an expanded space and we project to admissible value a posteriori.
The second step consists in implementing the tunneling in the FoM, allowing different models in the optimizer (for example KPLS or KPLSK), transforming the mixed inputs for building the metamodels in the relaxed space and transforming the outputs of the FoM for evaluating points in the tight original space and assigning labels to categorical variables(e.g x_3 should correspond to "blue").
Two pull requests were submitted and implemented in the last version of SMT. The later code was then refactored by Rémi Lafage (ONERA/DTIS). Therefore, the following presents the last version of the developments.

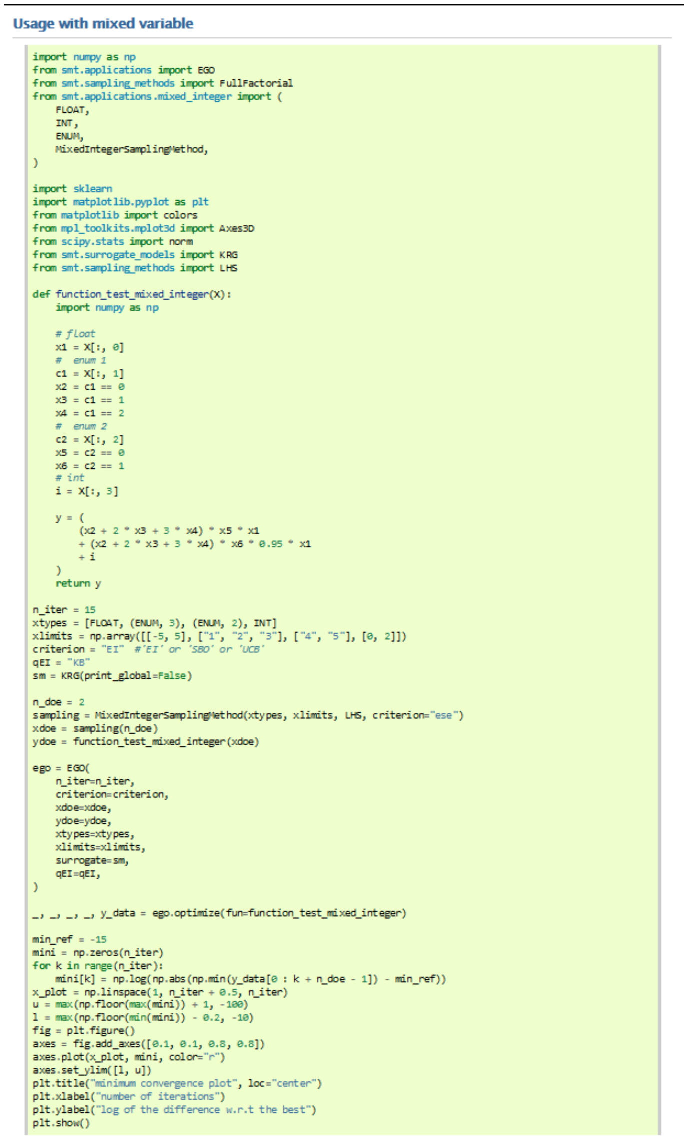


Example of mixed–integer Polynomial (QP) surrogate

```
import numpy as np
import matplotlib.pyplot as plt

from smt.surrogate_models import QP
from smt.applications.mixed_integer import MixedIntegerSurrogate, INT

xt = np.array([0.0, 1.0, 2.0, 3.0, 4.0])
yt = np.array([0.0, 1.0, 1.5, 0.5, 1.0])

# xtypes = [FLOAT, INT, (ENUM, 3), (ENUM, 2)]
# FLOAT means x1 continuous
# INT means x2 integer
# (ENUM, 3) means x3, x4 & x5 are 3 levels of the same categorical variable
# (ENUM, 2) means x6 & x7 are 2 levels of the same categorical variable

sm = MixedIntegerSurrogate(xtypes=[INT], xlimits=[[0, 4]], surrogate=QP())
sm.set_training_values(xt, yt)
sm.train()

num = 100
x = np.linspace(0.0, 4.0, num)
y = sm.predict_values(x)

plt.plot(xt, yt, "o")
plt.plot(x, y)
plt.xlabel("x")
plt.ylabel("y")
plt.legend(["Training data", "Prediction"])
plt.show()
```

```
___________________________________________________________________________

 Evaluation

      # eval points. : 100

   Predicting ...
   Predicting - done. Time (sec):  0.0000000

   Prediction time/pt. (sec) :  0.0000000
```

Figure 3.7: Build a mixed metamodel with SMT-GMHL



Usage with mixed variable

```
import numpy as np
from smt.applications import EGO
from smt.sampling_methods import FullFactorial
from smt.applications.mixed_integer import (
    FLOAT,
    INT,
    ENUM,
    MixedIntegerSamplingMethod,
)

import sklearn
import matplotlib.pyplot as plt
from matplotlib import colors
from mpl_toolkits.mplot3d import Axes3D
from scipy.stats import norm
from smt.surrogate_models import KRG
from smt.sampling_methods import LHS

def function_test_mixed_integer(X):
    import numpy as np

    # float
    x1 = X[:, 0]
    #  enum 1
    c1 = X[:, 1]
    x2 = c1 == 0
    x3 = c1 == 1
    x4 = c1 == 2
    #  enum 2
    c2 = X[:, 2]
    x5 = c2 == 0
    x6 = c2 == 1
    # int
    i = X[:, 3]

    y = (
        (x2 + 2 * x3 + 3 * x4) * x5 * x1
        + (x2 + 2 * x3 + 3 * x4) * x6 * 0.95 * x1
        + i
    )
    return y

n_iter = 15
xtypes = [FLOAT, (ENUM, 3), (ENUM, 2), INT]
xlimits = np.array([[-5, 5], ["1", "2", "3"], ["4", "5"], [0, 2]])
criterion = "EI"  #'EI' or 'SBO' or 'UCB'
qEI = "KB"
sm = KRG(print_global=False)

n_doe = 2
sampling = MixedIntegerSamplingMethod(xtypes, xlimits, LHS, criterion="ese")
xdoe = sampling(n_doe)
ydoe = function_test_mixed_integer(xdoe)

ego = EGO(
    n_iter=n_iter,
    criterion=criterion,
    xdoe=xdoe,
    ydoe=ydoe,
    xtypes=xtypes,
    xlimits=xlimits,
    surrogate=sm,
    qEI=qEI,
)

_, _, _, _, y_data = ego.optimize(fun=function_test_mixed_integer)

min_ref = -15
mini = np.zeros(n_iter)
for k in range(n_iter):
    mini[k] = np.log(np.abs(np.min(y_data[0 : k + n_doe - 1]) - min_ref))
x_plot = np.linspace(1, n_iter + 0.5, n_iter)
u = max(np.floor(max(mini)) + 1, -100)
l = max(np.floor(min(mini)) - 0.2, -10)
fig = plt.figure()
axes = fig.add_axes([0.1, 0.1, 0.8, 0.8])
axes.plot(x_plot, mini, color="r")
axes.set_ylim([l, u])
plt.title("minimum convergence plot", loc="center")
plt.xlabel("number of iterations")
plt.ylabel("log of the difference w.r.t the best")
plt.show()
```

Figure 3.8: Optimize with SMT-GMHL

Figure 3.7 presents the building of a mixed metamodel with SMT. **Figure 3.8** presents the optimization of a mixed metamodel with SMT.

In the following, I will explain how the code works and the different functions implemented into the final version. A new class named **mixed_integer** was implemented.

Inputs

- The input "xtype" is an array of strings representing the type of the variables (continuous, integer or categorical) and the number of levels for categorical variables.
- The input "xlimits" gives the bounds for every variable. For example xlimits= [[0.0,2.5],[-2,2], ["blue","red"]] for a continuous, an integer and a categorical variable.

Functions

- "unfold_with_continuous_limits" expands xlimits to add continuous dimensions. Each categorical level corresponds to a new continuous dimension in [0,1]. Integer dimensions are relaxed continuously.

- "cast_to_discrete_values" projects continuously relaxed values to their closer admissible values. For instance, if an input dimension is typed ["blue", "red", "green"] in xlimits a sample/row of the input x may contain the values (or mask) [..., 0, 0, 1, ...] to specify "green" for this original dimension.
- "fold_with_enum_index" reduces inputs from the discrete unfolded space to initial x dimension space where categorical x dimensions are valued by the index in the corresponding enumerate list. For instance, if an input dimension is typed ["blue", "red", "green"] a sample/row of the input x may contain the mask [..., 0, 0, 1, ...] which will be contracted in [..., 2, ...] meaning the "green" value.
- "unfold_with_enum_mask" expand categorical inputs from initial x dimension space where categorical x dimensions are valued by the index in the corresponding enumerate list to the discrete unfolded space. This function is the opposite of "fold_with_enum_index".
- "cast_to_enum_value" return enumerate "green" for the enumerate index 2.

Usage

Algorithm 8: Generating a mixed DoE

Result: Mixed DoE

vartype = ["int",("cate",2)]
xlimits= [[-2,2],["red","blue"]]

- Transform vartype into a relaxed array which define the size of the relaxed space. vartype'= [1,2,2]
- Relax the limits to have the bounds of this relaxed space. xlimits'= [[-2.0,2.0],[0.0,1.0],[0.0,1.0]]
- Generate a continuous DoE in the 3D space. DoE'= [(-1.2,0.2,0.3), (2.51,0.8,0.6)]
- Project the value of the DoE to admissible values. DoE'= [(-1.0,0.0,1.0), (3.0,1.0,0.0)]
- Assign labels to have a valid DoE. DoE = [(-1,"blue"),(3,"red")]

Algorithm 9: Building the model

Result: Mixed metamodel for the expensive black-box

Given vartype and xlimits.

- Transform vartype into a relaxed array which define the size of the relaxed space.
- Relax the limits to have the bounds of this relaxed space.
- Generate a mixed DoE like in the algorithm 8 and denote y the value of these points through the black-box or use a DoE if given.
- Use the projected DoE in the relaxed space and their corrresponding value to fit a continuous kriging model in the relaxed space.
- Project all values of the model to the prediction at their closer admissible value.

Algorithm 10: Mixed integer EGO

```
Result: Minimum of an expensive black-box
while not stopping criterion do
    • Compute the Figure of Merit (and the qEI) with projected values of mean and standard
      deviation in the relaxed space based on the metamodel.
    • Find the new point to evaluate with a continuous optimizer for the EI.
    • Project this new point to evaluate and label it.
    • Evaluate the point through the black-box.
    • Add the new value into the relaxed DoE at its corresponding coordinates.
    • Recalculate the mixed metamodel with a bigger DoE into the relaxed space.
end
```

Chapter 4

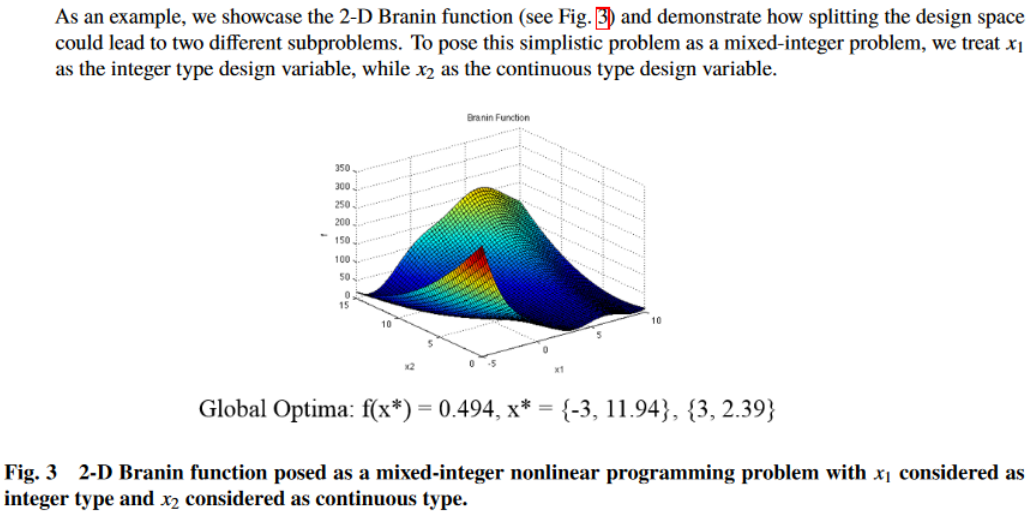
Results

The previously implemented algorithm was tested on different tests cases. Firstly we wanted to validate the mixed models implementation. We have three models which worked properly, Kriging, KPLS and KPLSK.

To begin with, it was tested on a set of test cases extracted from state of the art papers. Thereafter, it was tested on a real test case in which the Bayesian Optimization (BO) was coupled with Muldisciplinary Design Analysis (MDA) [46].

4.1 Analytic validation

I have summarized these cases in the webpage : https://mixed-optimization-benchmark.github.io/index/



As an example, we showcase the 2-D Branin function (see Fig. 3) and demonstrate how splitting the design space could lead to two different subproblems. To pose this simplistic problem as a mixed-integer problem, we treat x_1 as the integer type design variable, while x_2 as the continuous type design variable.

Branin Function
350
300
250
200
150
100
50
0
f
15
10
5
0
x2
-5
0
5
10
x1

Global Optima: f(x*) = 0.494, x* = {-3, 11.94}, {3, 2.39}

Fig. 3 2-D Branin function posed as a mixed-integer nonlinear programming problem with x_1 considered as integer type and x_2 considered as continuous type.

Figure 4.1: Case 5: the Branin testcase [25]

Figure 4.1 presents the test case called Branin 5. It consists in a Branin function but the dimension x_2 is considered as an integer one. Therefore, we still have the three local minima.

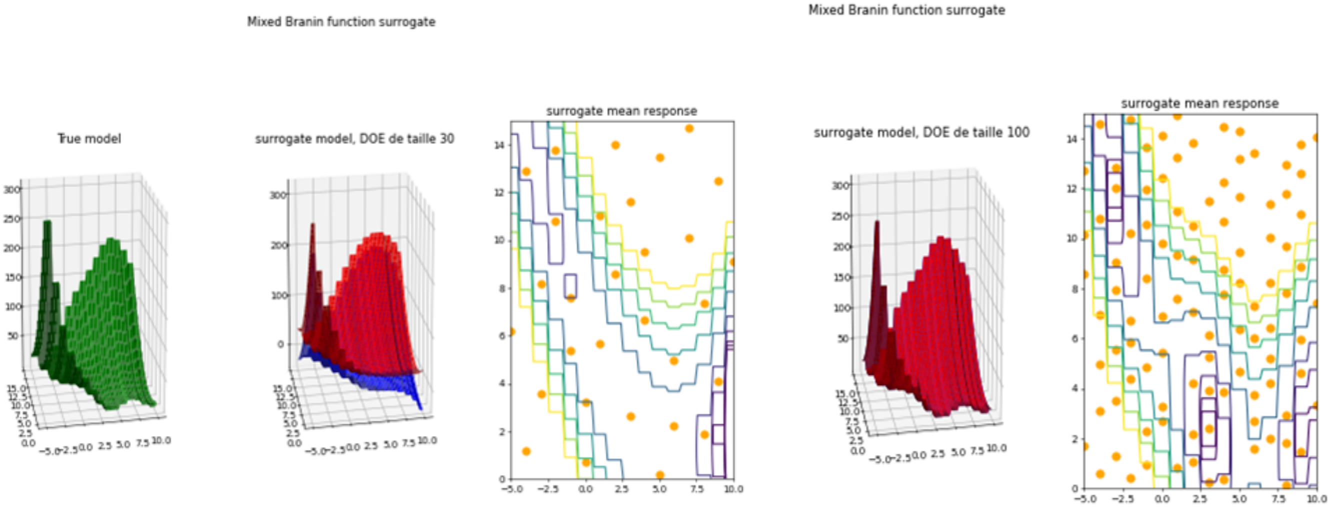
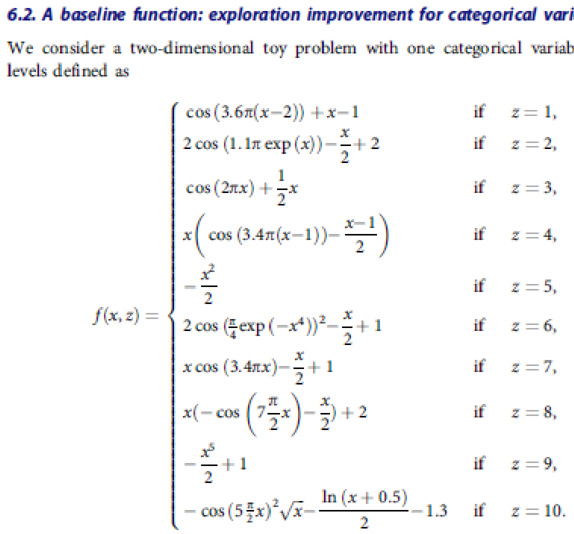


Figure 4.2: Branin_5 models visualization

On the left of **Figure 4.2**, in green, is plot the true function to approximate. For two sizes of DoE, we have the plots of $\mu + 3\sigma$ in red and $\mu - 3\sigma$ in blue. The interest of the Bayesian optimization is to estimate this uncertainty. Obviously, on the diagram for a DoE of 100 points, on the right, the blue and red curves are really close because we have a good model approximation but with only 30 points, we can clearly see that the model has some uncertainties. On the contour plot with 100 points, we can clearly see the three local minima in purple. Even with 30 points, we found the structure of the valley.



6.2. A baseline function: exploration improvement for categorical variables

We consider a two-dimensional toy problem with one categorical variable with 10 levels defined as

$$f(x,z) = \begin{cases} \cos(3.6\pi(x-2)) + x - 1 & \text{if} \quad z = 1, \\ 2\cos(1.1\pi\exp(x)) - \dfrac{x}{2} + 2 & \text{if} \quad z = 2, \\ \cos(2\pi x) + \dfrac{1}{2}x & \text{if} \quad z = 3, \\ x\left(\cos(3.4\pi(x-1)) - \dfrac{x-1}{2}\right) & \text{if} \quad z = 4, \\ -\dfrac{x^2}{2} & \text{if} \quad z = 5, \\ 2\cos(\frac{\pi}{4}\exp(-x^4))^2 - \dfrac{x}{2} + 1 & \text{if} \quad z = 6, \\ x\cos(3.4\pi x) - \dfrac{x}{2} + 1 & \text{if} \quad z = 7, \\ x(-\cos\left(7\dfrac{\pi}{2}x\right) - \dfrac{x}{2}) + 2 & \text{if} \quad z = 8, \\ -\dfrac{x^5}{2} + 1 & \text{if} \quad z = 9, \\ -\cos(5\frac{\pi}{2}x)^2\sqrt{x} - \dfrac{\ln(x+0.5)}{2} - 1.3 & \text{if} \quad z = 10. \end{cases}$$

Figure 4.3: Set 1 case

Figure 4.3 shows an other test case in dimension 11 in which there is a categorical variable with 10 levels and a continuous variable.

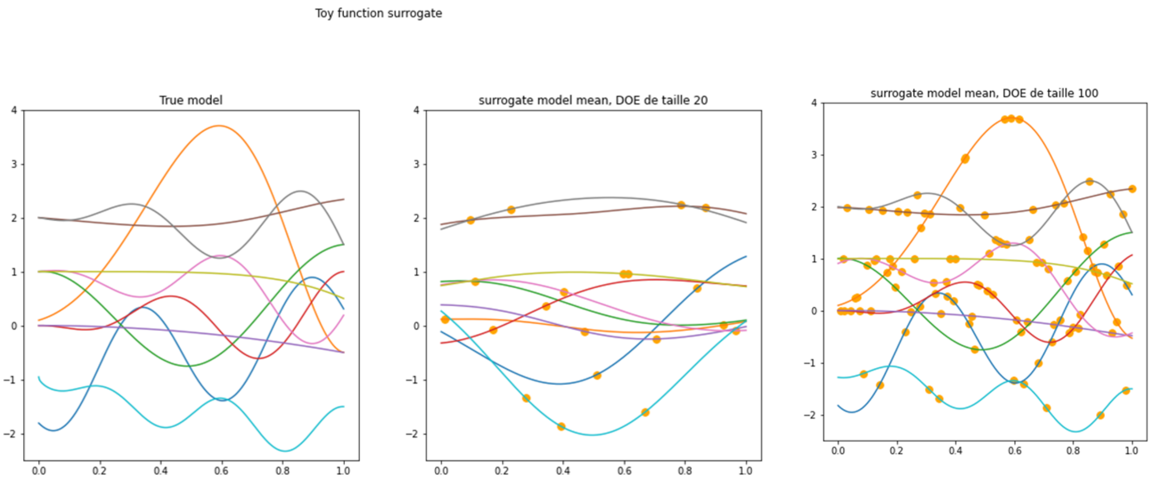


Figure 4.4: Set_1 models visualization

On the left of **Figure 4.4**, it is the true problem where each continuous function curve corresponds to a given level. Then, in the middle and on the right, two models are build with DoE of size 20 and 100. We see that the more we have points to build the model, the closer the model is to the true function. The methods to generate the mixed DoE should gives approximately the same amount of point for every level.

For the two functions Branin 5 and Set 1, in **Figure 4.5** and **Figure 4.6** and for different DoE sizes, we have plot two quantitative values for model validation.

On the left, there is an estimated Root Mean Square Error defined as $RMSE = \sqrt{\sum_{i=1}^{n} \frac{1}{n}(\hat{y}_i - y_i)}$.

- For a given DoE of size n, we generate a Latin Hypercube Sampling of size $2.n$.
- Half of this points are used for building the model with n points and are called the data set.
- The other n points are used to calculate the RMSE and are called the validation set.

On the right, the plot represents the surrogate models minima for different DoE size and in dotted line the true minimum. These minima are computed with the SBO criterion on the models and optimize with SLSQP.

For the values to be consistent, they are calculated over 20 runs and represented by boxplots. The yellow line is the median, the box is delimited by the first and the third quartile.

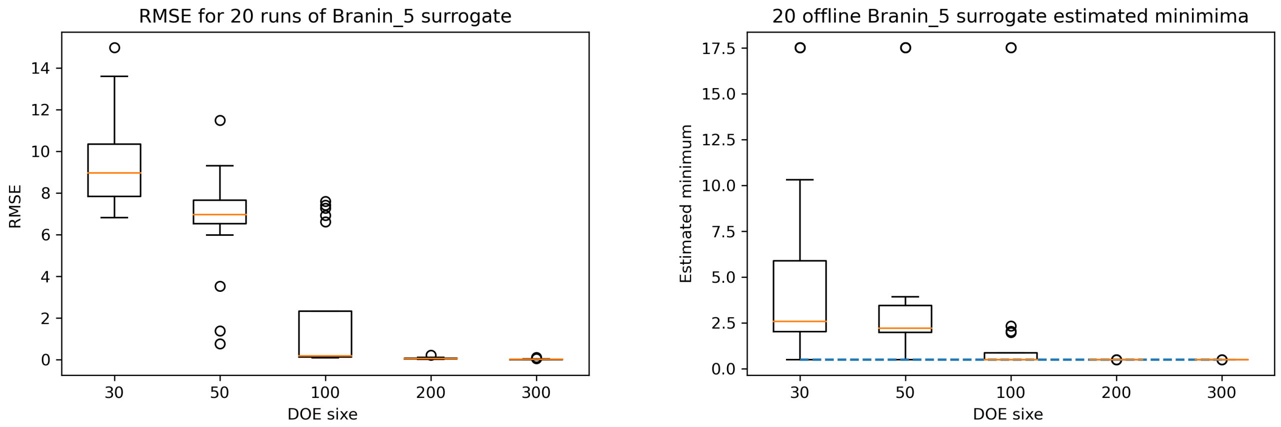


Figure 4.5: Branin_5 model validation

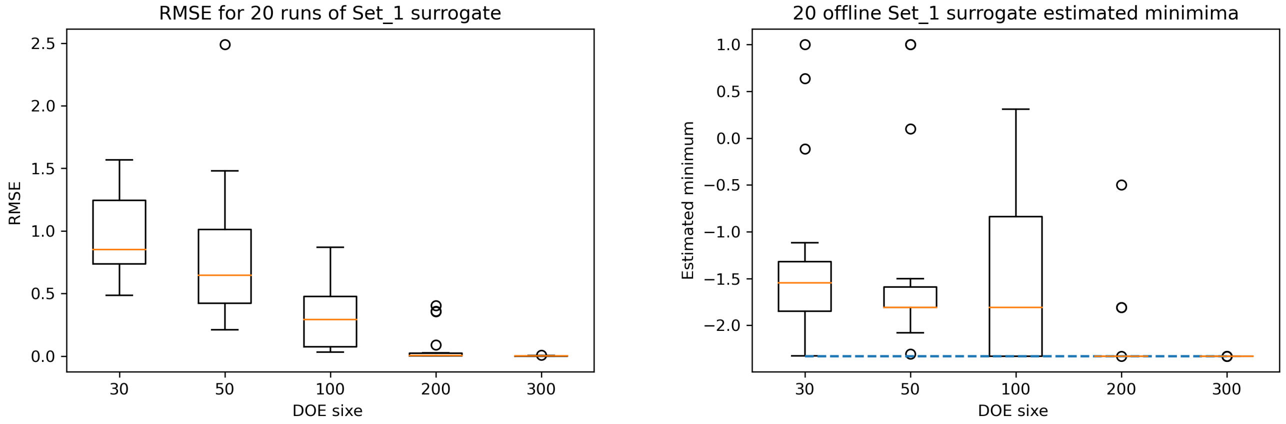


Figure 4.6: Set_1 model validation

Figure 4.5 and **Figure 4.6** show that the RMSE converges to 0 when the number of points in the DoE increase and that the estimated minimum converge to the true minimum. Another effect is that the biggest the DoE the less the variation between the 20 runs is important and that is reflected by the fact that the boxplot tends to become flat.

Now, we want to test the optimization of the metamodels by choosing adaptive enrichment points instead of picking a lot of points. In this case, we want to have the best estimated minimum with the least evaluation points.

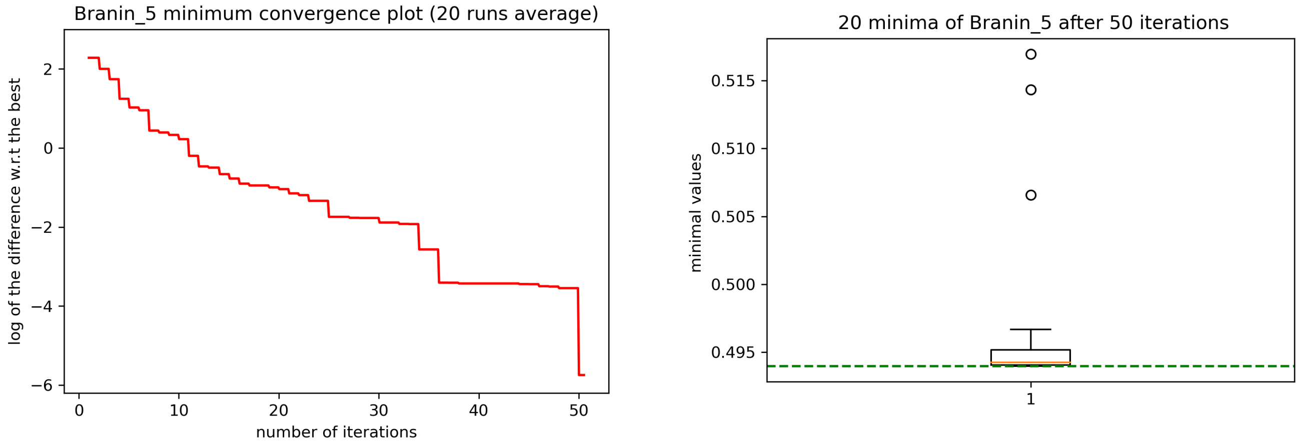


Figure 4.7: Branin_5 optimization validation

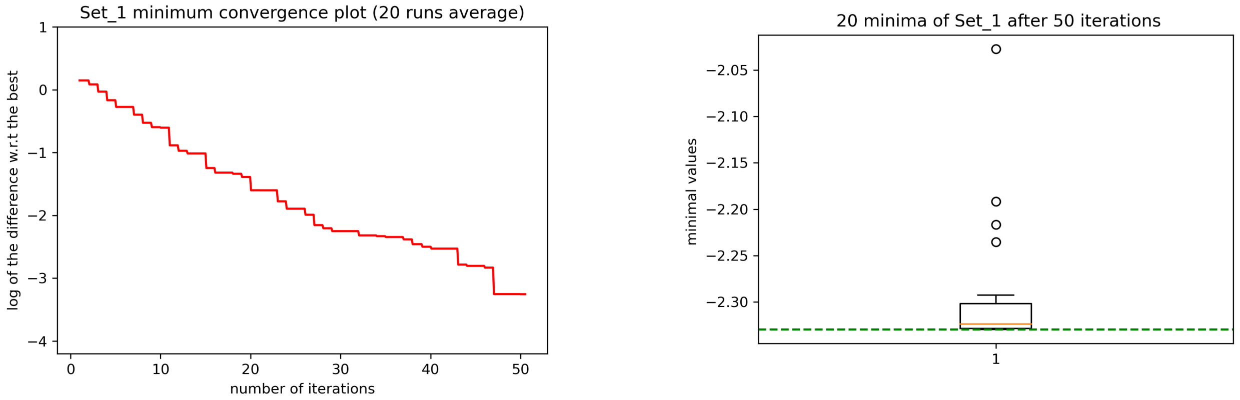


Figure 4.8: Set_1 optimization validation

In **Figure 4.7** and **Figure 4.8**, on the left, it is the log of the difference between the incumbent and the actual minimum on average over 20 runs for different budget and on the right, we show boxplots of the 20 incumbent after 50 iterations, the true minimum still being the dotted line. By considering a log of difference of -2, the Branin function converges after 33 iterations on average and the set of 10 curves converges after 27 iterations.

An interesting result is that the final variation, after 50 runs, between the 20 minima founded for Branin 5 is of the order of 0.02 whereas it is of the order of .33 for the function Set 1. Hence, when an optimization is not precise on its minimum, error of e^{-4} for Set 1 contrary to an error of e^{-6} for Branin 5, then, the best point found can vary a lot from one optimization to another. In practice, we do not know the real true minimum but if we obtain important variations between two runs of the same optimization, it is a good indicator that the optimization has not converged.

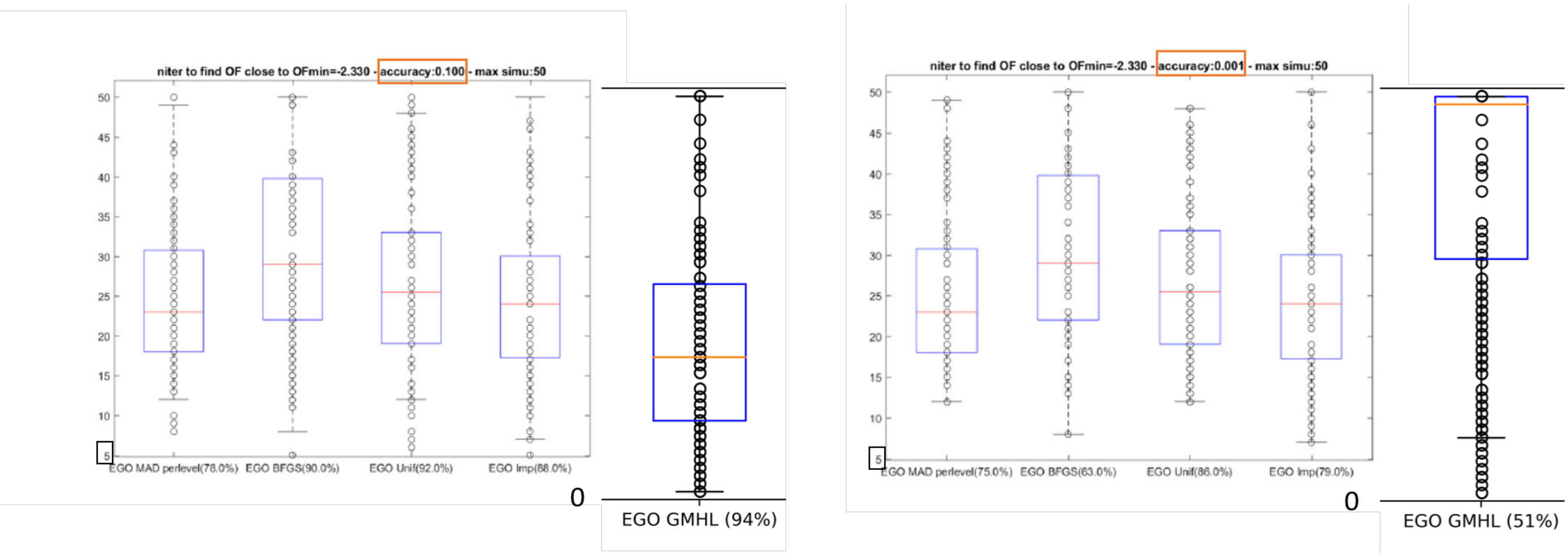
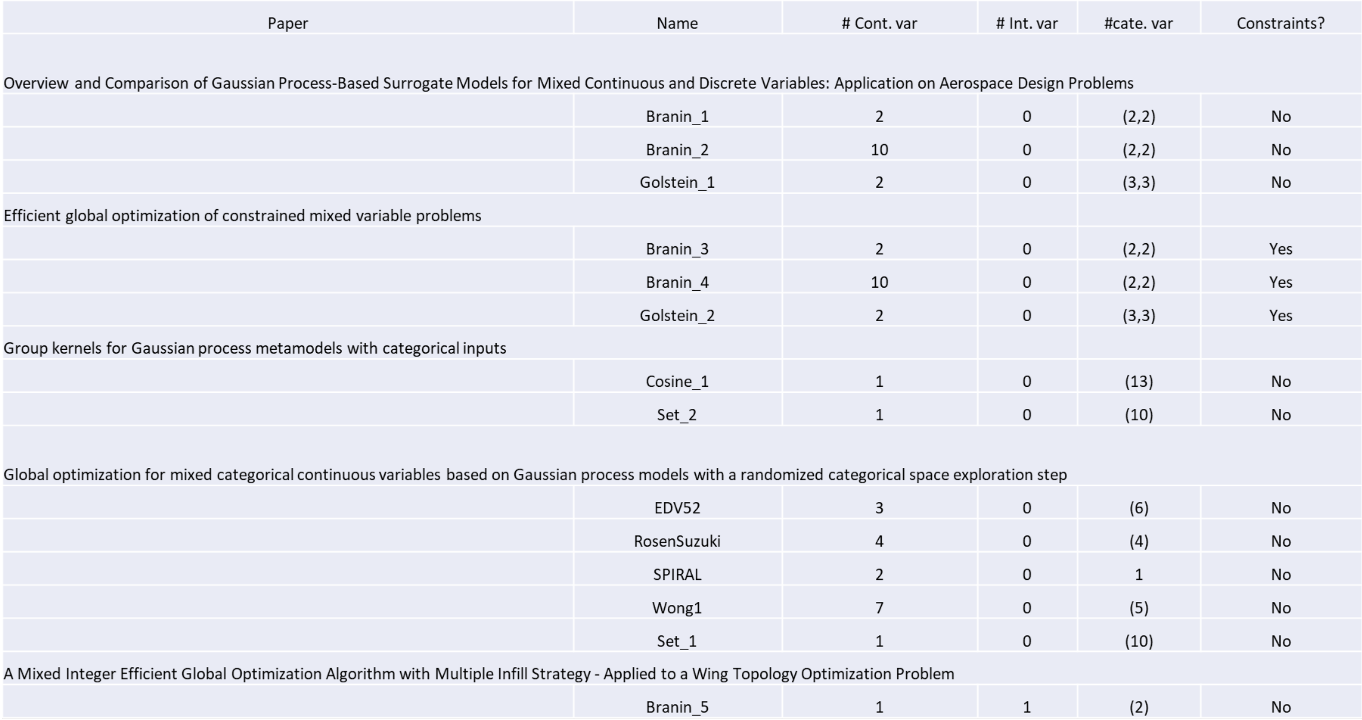


Figure 4.9: CAT-EGO and GMHL optimizations compared on Set_1 test case

Figure 4.9 compare 4 variants of the method CAT-EGO with the GMHL implementation on the test case Set_1. In CAT-EGO, the model used is the covariance matrices mixed kernel [31]. The four methods correspond to different Expected Improvement sub-optimizations: NOMAD with Uniform Sampling (EGO Uni), NOMAD with improved sampling (EGO Imp) which corresponds to the method developed in CAT-EGO [42] and two methods with a continuous optimization for each of the 10 possible levels, BFGS algorithm (EGO BFGS) or NOMAD (EGO MAD per level). Starting with 100 DoE of 5 points, for each method, the boxplots shows the numbers of iterations needed to converge to a given accuracy (0.1 on the left, 0.001 on the right) with a maximal budget of 50 iterations. For every method, the number between brackets is the number of optimizations that converged to the correct optimal categorical level. For the accuracy 0.1, GMHL converged more quickly than every other CAT-EGO method and 94 GMHL optimizations found the correct level. On the other hand, when the requested accuracy is 0.001, only half the iteration converged with GMHL what is worse than every other variant of CAT-EGO. The GMHL algorithm uses a new dimension for every possible level, in this case, the KPLS models reduce the total dimension from 11 to 5 but this makes the algorithm exploring more the space and, in particular, the categorical dimensions. As the model is more complex, it can find quickly a promising region. This is also verified on the Branin test case where the GMHL algorithm find a local optimum in approximately 5 iterations. Notwithstanding this good exploration at the start, when it comes to find a good accuracy, the algorithm is not adapted anymore because of three drawbacks : the increased number of dimensions led to a large unfold region to explore, the continuous by part model makes it hard to find a good descent direction for local optimization and the continuous expected improvement deceive the algorithm and makes it partly searching into non-significant space (for example level 1.3 besides of level 1.2 that will both be rounded to level 1). Compared to CAT-EGO, GMHL algorithm is better to explore the space and find an approximation of the minimum but it becomes harder to find a precise value of this minimum and, in the last case, CAT-EGO is a more robust method. The proposed CAT-EGO method (EGO Imp) and NOMAD perlevel gives similar results but the latter used 10 times more evaluations. EGO Imp seems the best of the compared methods for finding a minimal value with an high accuracy.

In the previous sections, two test cases were presented but you can find more problems in Appendix A where you can find all the problems definition and the resulting curves and boxplots.
This table (**Figure 4.10**) summarizes the test cases used that you can find in Appendix but the cases with constraints were not evaluated during this internship.



Paper	Name	# Cont. var	# Int. var	#cate. var	Constraints?
Overview and Comparison of Gaussian Process-Based Surrogate Models for Mixed Continuous and Discrete Variables: Application on Aerospace Design Problems					
	Branin_1	2	0	(2,2)	No
	Branin_2	10	0	(2,2)	No
	Golstein_1	2	0	(3,3)	No
Efficient global optimization of constrained mixed variable problems					
	Branin_3	2	0	(2,2)	Yes
	Branin_4	10	0	(2,2)	Yes
	Golstein_2	2	0	(3,3)	Yes
Group kernels for Gaussian process metamodels with categorical inputs					
	Cosine_1	1	0	(13)	No
	Set_2	1	0	(10)	No
Global optimization for mixed categorical continuous variables based on Gaussian process models with a randomized categorical space exploration step					
	EDV52	3	0	(6)	No
	RosenSuzuki	4	0	(4)	No
	SPIRAL	2	0	1	No
	Wong1	7	0	(5)	No
	Set_1	1	0	(10)	No
A Mixed Integer Efficient Global Optimization Algorithm with Multiple Infill Strategy - Applied to a Wing Topology Optimization Problem					
	Branin_5	1	1	(2)	No

Figure 4.10: Test cases overview

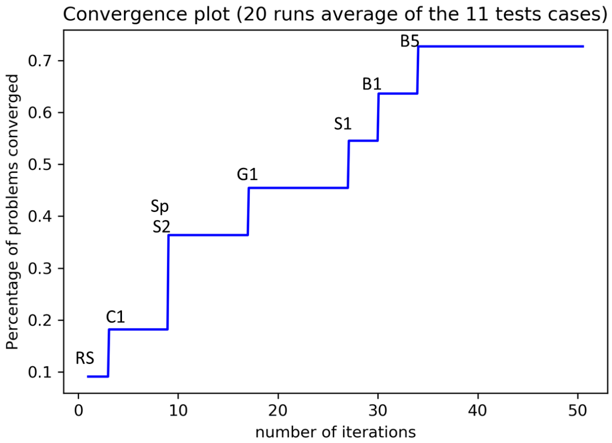


Figure 4.11: Mean percentage of test cases converged for a given budget

Based on the 11 problems without constraints, **Figure 4.11** shows the number of iterations required in mean over 20 runs to converge ($\log |f_{min} - f^*_{true}| \leq -2$). The letter corresponds to abbreviation of the name of the cases (S1= Set 1, RS= Rosen-Suzuki, B5= Branin 5,...).

4.2 Preliminary results on a aerospace engineering application

To face the increasing environmental footprint of commercial aviation, industrial and research efforts have been focusing on exploring unconventional configurations and new propulsion paradigms, mostly based on electric technology. Such explorations require Overall Aircraft Design (OAD) that has to be performed in an integrated multidisciplinary design environment [47].
Such design environments are often limited to multidisciplinary analysis, adapted for a single aircraft configuration or require an important effort to be mastered. FAST-OAD (Future Aircraft Sizing Tool-OAD) is a software program developed by ONERA and ISAE-SUPAERO for aircraft sizing analysis and optimization with emphasis on user friendliness and modularity. It is an aircraft sizing code based on Multidisciplinary Design Optimization (MDO) [2, 5] techniques and the point mass approach to estimate the required fuel and energy consumption for a given set of Top Level Aircraft Requirements (TLAR). This software FAST-OAD is an evolution of the original software program, called FAST [48] based on OpenMDAO [3].

TLAR defined possible configurations and constraints. We want to select between configurations.
To do so, we used the Multidisciplinary Design Optimizer of FAST-OAD. This optimization problem that aims at minimizing the fuel consumption for the mission (objective), with respect to the wing position (design variables), subject to a static margin (constraints).
Sometimes, the constraints can not be respected because of the initial configuration so we should have to optimize with constraints, the MDA is used as a black box function and we do not touch the wing position variables, the "black-box function" does it. See the **Figure 4.12** . The interface has been set up by Nathalie Bartoli with the help of Christophe David (ONERA/DTIS).

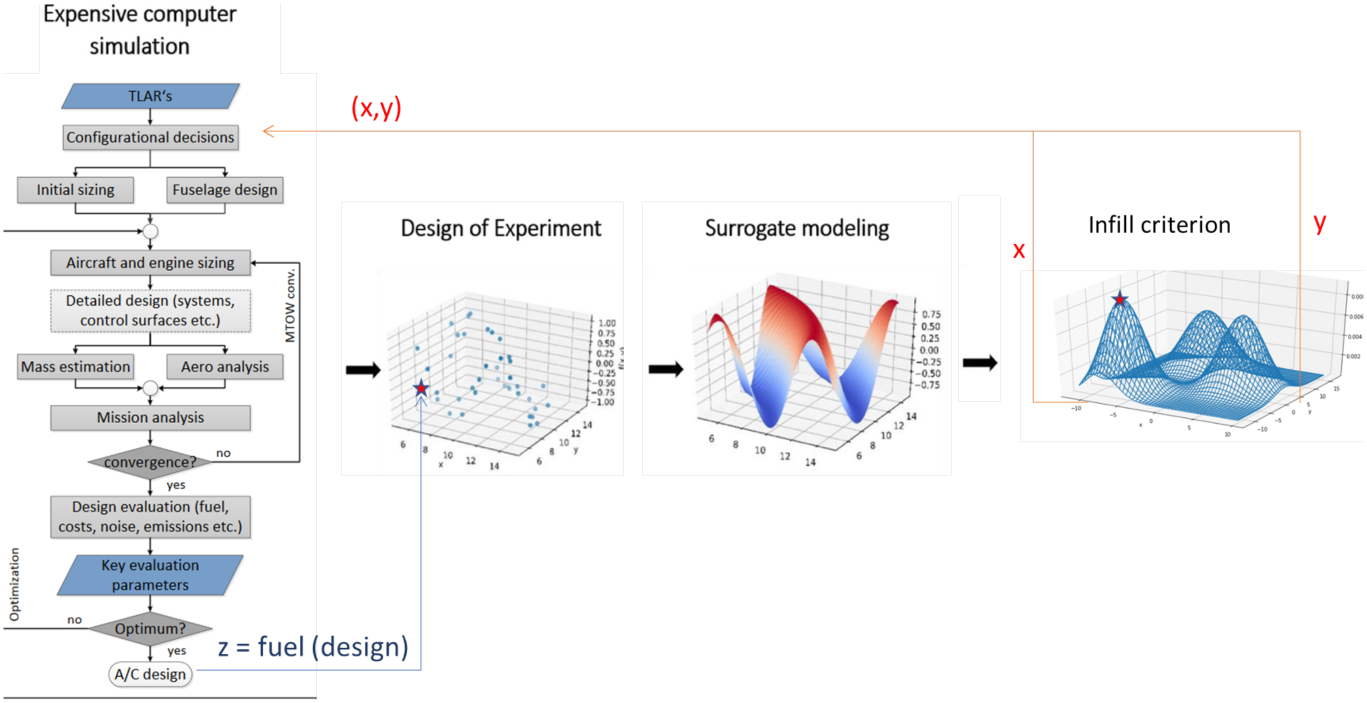
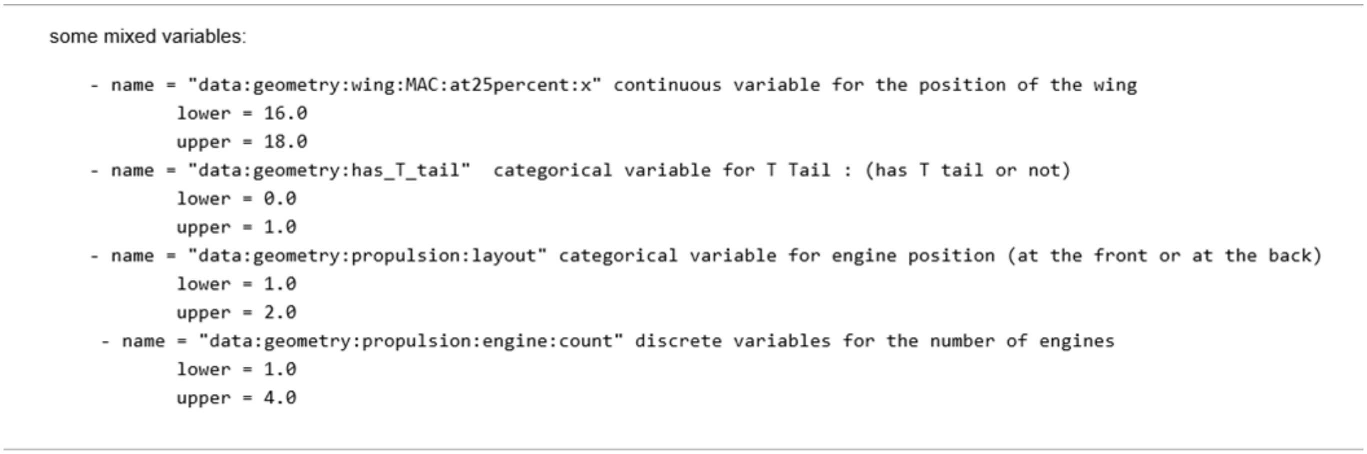


Figure 4.12: FAST-OAD coupled with SMT for the optimization

Technically, the surrogate model of the fuel consumption for the mission depending on the initial configuration should be influenced by the hidden wing position variables so it should mimic the outputs of the underlying optimizer.

The treated test case is the optimization of the Short Range Reference Aircraft CeRAS (CEntral Reference Aircraft data System) based on a Airbus A320. The details can be found at: https://w3.onera.fr/ceas-tcad2014/sites/w3.onera.fr.ceas-tcad2014/files/06_reference_aircraft_s1_a_rwth.pdf
The variables of the SMT-GMHL optimizer are:

- The number of engines *between 1 and 4*,
- The layout for the engine position: *under the wing* or *on rear fuselage*,
- If the horizontal tail is *attached to fuselage* or *on top of vertical tail*,
- The Mean Average Chord at 25 percent for the wing in the x dimension *between 16 and 18 meters*.



some mixed variables:

```
 - name = "data:geometry:wing:MAC:at25percent:x" continuous variable for the position of the wing
         lower = 16.0
         upper = 18.0
 - name = "data:geometry:has_T_tail"  categorical variable for T Tail : (has T tail or not)
         lower = 0.0
         upper = 1.0
 - name = "data:geometry:propulsion:layout" categorical variable for engine position (at the front or at the back)
         lower = 1.0
         upper = 2.0
  - name = "data:geometry:propulsion:engine:count" discrete variables for the number of engines
         lower = 1.0
         upper = 4.0
```

Figure 4.13: Mixed variables for the BO

Figure 4.13 shows the definition in FAST-OAD of the variables, their name, their meaning and their bounds.

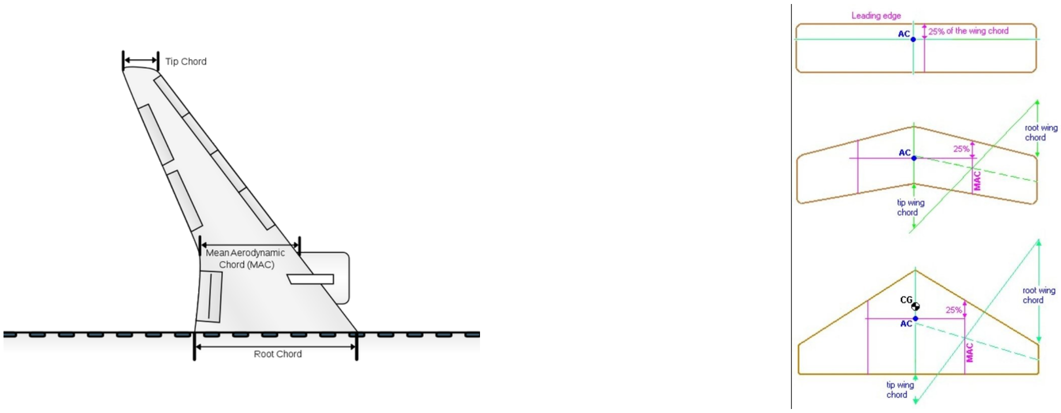


Figure 4.14: FAST test case continuous variable

Figure 4.14 shows the meaning in aircraft design of the continuous variable. On **Figure 4.16**, we can see that the tail of the aircraft and its size change from one configuration to another. The problem that we will resolve in SMT is, without constraint:

$$\underset{x,\ layout,\ T\_tail,\ \#\ of\ engines}{\text{minimize}} \quad \text{Mission Fuel}(FAST-OAD(x, layout, T\_tail, \#\ of\ engines))$$

For every 16 configurations (2 layouts, with or without t-tail, 1-4 number of engines), I ran a 1D continuous optimization with the x_mac_at_25_percent variable between 16 and 18 and display the outputs.

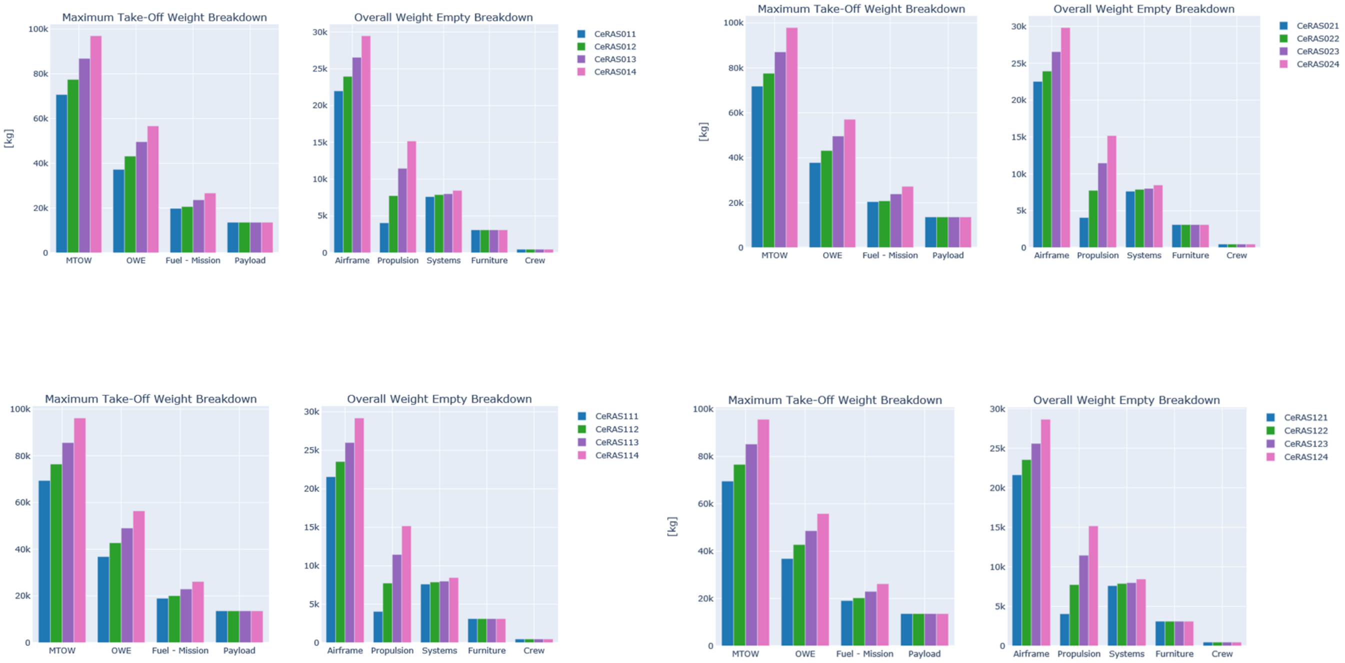


Figure 4.15: Data of the converged CeRAS configurations

Figure 4.15 shows for every configuration, the outputs of the MDO optimization with FAST-OAD for the value minimizing the *Fuel - Mission* output. The legend "CeRASxyz" reads CeRAS configuration with T_tail x in layout y and with z engines.

The 4 engines configurations are thus in pink and we see that the Maximum Take-Off Weight Breakdown (MTOWB) scales with the number of engines. The aircraft minimizing the fuel consumption is the one with 1 engine under the wing and an horizontal tail attached to the top of the vertical tail. Obviously, this configuration is unstable and the static margin is negative.

By running 16 optimizations purely continuous in 1D, we need around 2 points in DoE and 3 iterations to converge every time for a total of 48 iterations and 32 points DoE: **80 evaluations** of the expensive black-box function.

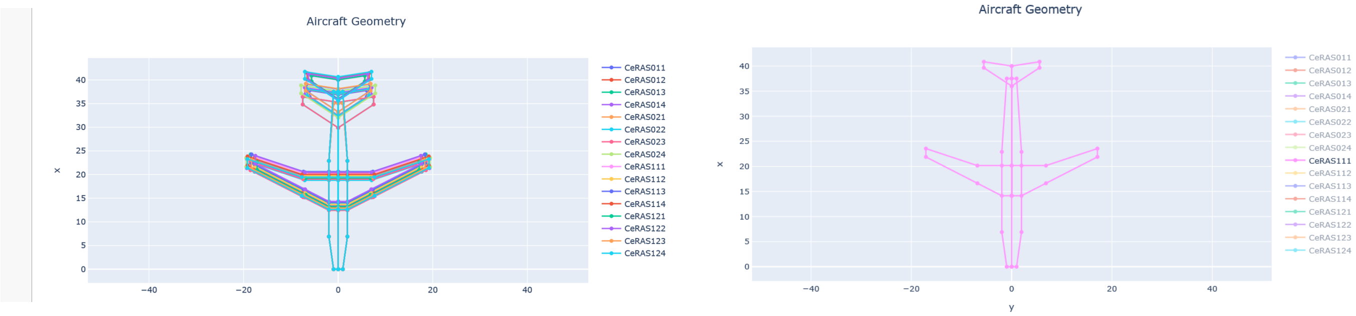


Figure 4.16: Geometries of the CeRAS configurations with the optimal on the right

Figure 4.16 is the contour of geometry of the airplane for every continuously optimized configuration. The optimal configuration is CeRAS 111 (layout 1 with a T_tail and 1 engine) for MAC at 25% of 16 meters.

With the new mixed code, with 10 points in the initial DoE, we assure to find the same optimal configuration in 5 iterations which is a lot faster to compute: **15 evaluations** (instead of 80 evaluations) of the expensive black-box function. The minimal fuel consumption is 18799 kg.
The results are based on the optimization done with FAST-OAD. FAST-OAD is still in a development phase and the mixed optimization was done without constraint. Therefore, the previous preliminary results are only a proof of concept and should be considered as such. The airplane with only one central engine is not physically relevant.

Chapter 5

Conclusion

To conclude, this internship allow us to do a brief review of the literature on the mixed integer efficient global optimization (bayesian optimization).
It concludes really well my Master II of the Ecole Nationale de l'Aviation Civile in Operations Research and my Engineering diploma of the Institut National des Sciences Appliquées de Toulouse.
It allowed me to go further from the courseworks Advanced Combinatory Optimization (OCA) of Capelle and Berghman and to overtake the courseworks of the INSA Computer Experiments with the article of Roustant et al.
This internship at ISAE-Supaero and ONERA with N. Bartoli, Y. Diouane, T. Lefebvre and J. Morlier was then a perfect opportunity to achieve my second cycle studies with strong relations with both INSA and ENAC.

On the personal aspect, this internship was in an excellent environment of research, with a lot of researchers and students and the supervisors helped me a lot to achieve better, especially during the covid-19 lockout and I want to thanks them all in this conclusion.
They gave me the opportunity to make two presentations of the preliminary works of this internship with M. Kokkolaras (McGill University, Canada) and D. Sinoquet (IFPEN, France).

About the results, I have still works to do but we can talk about the implementation into SMT, an open-source Python library, of the GMHL methods to deal with integer and categorical variables with the help of R. Lafage (ONERA/DTIS). After investigation of the method, both mixed models and mixed optimization were implemented.
It was started to do a base of problems for mixed integer comparaison of methods and the complete results of the methods are in appendix A. We treated the CeRAS problem with the ONERA toolbox FAST-OAD with the help of C. David and S. Defoort (ONERA/DTIS).

This internship will end on the twenty-five of September and will lead to a thesis to tackle the high dimension "High dimensional multidisciplinary design optimization for aircraft design". In this PhD thesis, we will investigate efficient parameterization tools to significantly reduce the number of design variables by using active learning technics. An extension of the method could be also proposed to handle mixed continuous and categorical inputs using some previous works on low dimensional problems. Practical implementations within the OpenMDAO framework (an open source MDO framework developed by NASA) are expected. This will continue the previous PhD works of

M. A Bouhlel and R. Priem.

This internship led to correct results. We have still some improvement to do like comparison with other methods, including new models based on the mixed kernels.... The results we have now are a first step toward optimized mixed integers problems which can be presented to the European project AGILE 4.0 and we are looking for generalize it to high dimension and constraints within the solver SEGOMOE.

Bibliography

[1] N. Bartoli, T. Lefebvre, S. Dubreuil, M. Panzeri, R. D'Ippolito, K. Anisomov, and A. Savelyev. "Robust nacelle optimization design investigated in the AGILE european project". In: *19th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. 2018. DOI: 10.2514/6.2018-3250.

[2] A. Lambe and J. R. R. A. Martins. "A unified description of MDO architectures". In: *9th World Congress on Structural and Multidisciplinary Optimization*. 2011.

[3] J. S. Gray, J. T. Hwang, J. R. R. A. Martins, K. T. Moore, and B. A. Naylor. "OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization". *Structural and Multidisciplinary Optimization* 59 (2019), pages 1075–1104. DOI: 10.1007/s00158-019-02211-z.

[4] W. Sun, X. Wang, L. Wang, and X. Song. "Multidisciplinary design optimization of tunnel boring machine considering both structure and control parameters under complex geological conditions". *Structural and Multidisciplinary Optimization* 54 (2016), pages 1073–1092. DOI: 10.1007/s00158-016-1455-9.

[5] A. Lambe and J. R. R. A. Martins. "Extensions to the design structure matrix for the description of multidisciplinary design, analysis, and optimization processes". *Structural and Multidisciplinary Optimization* 46 (2012), pages 273–284. DOI: 10.1007/s00158-012-0763-y.

[6] D. R. Jones, M. Schonlau, and W. J. Welch. "Efficient Global Optimization of Expensive Black-Box Functions". *Journal of Global Optimization* 13 (1998), pages 455–492. DOI: 10.1023/A:1008306431147.

[7] J. Močkus. "On bayesian methods for seeking the extremum". In: *Optimization Techniques IFIP Technical Conference Novosibirsk*. 1974. DOI: 10.1007/978-3-662-38527-2_55.

[8] A. Forrester, A. Sobester, and A. Keane. *Engineering Design via Surrogate Modelling: A Practical Guide*. Wiley, 2008. ISBN: 9780470770795.

[9] S. H. Kim and F. Boukouvala. "Surrogate-Based Optimization for Mixed-Integer Nonlinear Problems". *Computers & Chemical Engineering* 140 (2020). DOI: 10.1016/j.compchemeng.2020.106847.

[10] C. E. Rasmussen and J. Quiñonero-Candela. "A Unifying View of Sparse Approximate Gaussian Process Regression". *Journal of Machine Learning Research* 6 (2005), pages 1939–1959.

[11] N. Bartoli, T. Lefebvre, S. Dubreuil, R. Olivanti, R. Priem, N. Bons, J. R. R. A. Martins, and J. Morlier. "Adaptive modeling strategy for constrained global optimization with application to aerodynamic wing design". *Aerospace Science and technology* 90 (2019), pages 85–102. DOI: 10.1016/j.ast.2019.03.041.

[12] R. Jin, W. Chen, and A. Sudjianto. "An efficient algorithm for constructing optimal design of computer experiments". *Journal of Statistical Planning and Inference* 134 (2005), pages 268–287. DOI: 10.1016/j.jspi.2004.02.014.

[13] D. Duvenaud. "Automatic model construction with Gaussian processes". PhD thesis. University of Cambridge, 2014.

[14] M. Bouhlel, N. Bartoli, R.G. Regis, A. Otsmane, and J. Morlier. "Efficient Global Optimization for high-dimensional constrained problems by using the Kriging models combined with the Partial Least Squares method". *Engineering Optimization* 50 (2018), pages 2038–2053. DOI: 10.1080/0305215X.2017.1419344.

[15] E. Mehdad and J. P. C. Kleijnen. "Efficient global optimisation for black-box simulation via sequential intrinsic Kriging". *Journal of the Operational Research Society* 69 (2018), pages 1725–1737. DOI: 10.1080/01605682.2017.1409154.

[16] W. Ponweiser, T. Wagner, and M. Vincze. "Clustered multiple generalized expected improvement: A novel infill sampling criterion for surrogate models". In: 2008. DOI: 10.1109/CEC.2008.4631273.

[17] C. Audet and J. E. Dennis. “Pattern Search Algorithms for Mixed Variable Programming”. *SIAM Journal on Optimization* 11 (2000), pages 573–594. DOI: 10.1137/S1052623499352024.

[18] A. Scotto Di Perrotolo. “A Theoretical Framework for Bayesian Optimization Convergence”. Master’s thesis. KTH School of Engineering Sciences, 2018.

[19] C. Audet, J. E. Dennis, A. Booker, P. Frank, and D. Moore. “A Surrogate-Model-Based Method For Constrained Optimization”. In: *8th Symposium on Multidisciplinary Analysis and Optimization.* 2000. DOI: 10.2514/6.2000-4891.

[20] N. Amaioua an C. Audet, A. R. Conn, and S. Le Digabel. “Efficient solution of quadratically constrained quadratic subproblems within the mesh adaptive direct search algorithm”. *European Journal of Operational Research* 268 (2018), pages 13–24. DOI: 10.1016/j.ejor.2017.10.058.

[21] R. Michael and L. Torczon. “A Globally Convergent Augmented Lagrangian Pattern Search Algorithm For Optimization With General Constraints And Simple Bounds”. *SIAM Journal on Optimization* 12 (2000), pages 075–1089. DOI: 10.1137/S1052623498339727.

[22] S. Bagheri, W. Konen, R. Allmendinger, J. Branke, K. Deb, J. Fieldsend, D. Quagliarella, and K. Sindhya. “Constraint handling in efficient global optimization”. In: *Proceedings of the Genetic and Evolutionary Computation Conference.* 2017. DOI: 10.1145/3071178.3071278.

[23] S. Roy, W. A. Crossley, J. T. Hwang, J. R. R. A. Martins, K. T. Moore, and J. S. Gray. “A Mixed Integer Efficient Global Optimization Algorithm for the Simultaneous Aircraft Allocation-Mission-Design Problem”. In: *AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference.* 2017. DOI: 10.2514/6.2017-1305.

[24] A. M. Bouhlel, N. Bartoli, A. Otsmane, and J. Morlier. “Improving kriging surrogates of high-dimensional design models by Partial Least Squares dimension reduction”. *Structural and Multidisciplinary Optimization* 53 (2016), pages 935–952. DOI: 10.1007/s00158-015-1395-9.

[25] S. Roy, W. A. Crossley, B. K. Stanford, K. T. Moore, and J. S. Gray. “A Mixed Integer Efficient Global Optimization Algorithm with Multiple Infill Strategy - Applied to a Wing Topology Optimization Problem”. In: *AIAA Scitech 2019 Forum.* 2019. DOI: 10.2514/6.2019-2356.

[26] E. C. Garrido-Merchán and D. Hernández-Lobato. “Dealing with categorical and integer-valued variables in Bayesian Optimization with Gaussian processes”. *Neurocomputing* 380 (2020), pages 20–35. DOI: 10.1016/j.neucom.2019.11.004.

[27] J. Lévesque, A. Durand, C. Gagné, and R. Sabourin. “Bayesian optimization for conditional hyperparameter spaces”. In: *2017 International Joint Conference on Neural Networks (IJCNN).* 2017. DOI: 10.1109/IJCNN.2017.7965867.

[28] F. Hutter. “Automated Configuration of Algorithms for Solving Hard Computational Problems”. PhD thesis. The University Of British Columbia, 2009.

[29] F. Hutter, H.H. Hoos, and K. Leyton-Brown. “Sequential Model-Based Optimization for General Algorithm Configuration”. In: *International Conference on Learning and Intelligent Optimization.* 2011. DOI: 10.1007/978-3-642-25566-3_40.

[30] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. “Algorithms for Hyper-Parameter Optimization”. In: *25th Annual Conference on Neural Information Processing Systems.* hal-00642998. 2011.

[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”. *arXiv e-prints* (2018). arXiv: 1802.02368v2 [math.ST].

[32] J. Pinheiro and D. M. Bates. “Unconstrained parametrizations for variance-covariance matrices”. *Statistics and Computing* 6 (1996), pages 289–296. DOI: 10.1007/BF00140873.

[33] J. Pelamatti, B. Loïc, M. Balesdent, E.-G Talbi, and Y. Guerin. “Efficient global optimization of constrained mixed variable problems”. *Journal of Global Optimization* 73 (2019), pages 583–613. DOI: 10.1007/s10898-018-0715-1.

[34] Q. Zhou, P. Z.G Qian, and S. Zhou. “A Simple Approach to Emulation for Computer Models With Qualitative and Quantitative Factors”. *Technometrics* 53 (2011), pages 266–273. DOI: 10.2307/23210402.

[35] M. A. Stelmack, N. Nakashima, and S. M. Batill. “Genetic Algortihms For Mixed Discrete/Continuous Optimization In Multidisciplinary Design”. In: *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization.* 1998. DOI: 10.2514/6.1998-4771.

[36] M. Halstrup. "Black-Box Optimization of Mixed Discrete-Continuous Optimization Problems". PhD thesis. TU Dortmund, 2016.

[37] J. E. Dennis and V. Torczon. "Direct Search Methods on Parallel Machines". *SIAM Journal on Optimization* 1 (1991), pages 448–474. DOI: 10.1137/0801027.

[38] M. Kokkolaras, C. Audet, and J. E. Dennis. "Mixed Variable Optimization of the Number and Composition of Heat Intercepts in a Thermal Insulation System". *Optimization and Engineering* 2 (2001), pages 5–29. DOI: 10.1023/A:1011860702585.

[39] M.A. Abramson. "Pattern Search Algorithms for Mixed Variable General Constrained Optimization Problems". PhD thesis. Rice University, 2003.

[40] J. Mueller, C. Shoemaker, and R. Piché. "SO-MI: A surrogate model algorithm for computationally expensive nonlinear mixed-integer black-box global optimization problems". *Computers and Operations Research* 40 (2013), pages 1383–1400. DOI: 10.1016/j.cor.2012.08.022.

[41] I. P. Androulakis, C. D. Maranas, and C.A. Floudas. "αBB: A global optimization method for general constrained nonconvex problems". *Journal of Global Optimization* 7 (1995), pages 337–363. DOI: 10.1007/BF01099647.

[42] M. M. Zuniga and 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), pages 1–32. DOI: 10.1080/03155986.2020.1730677.

[43] D. Nguyen, S. Gupta, S. Rana, A. Shilto, and S. Venkatesh. "Bayesian Optimization for Categorical and Category-Specific Continuous Inputs" (2020). DOI: 10.1609/aaai.v34i04.5971.

[44] M. A. Bouhlel, J. T. Hwang, N. Bartoli, R. Lafage, J. Morlier, and J. R. R. A. Martins. "A Python surrogate modeling framework with derivatives". *Advances in Engineering Software* 135 (2019). DOI: 10.1016/j.advengsoft.2019.03.005.

[45] S. Zhang and J. Norato. "Finding Better Local Optima in Topology Optimization via Tunneling". In: *ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. 2018. DOI: 10.1115/DETC2018-86116.

[46] T. Lefebvre, N. Bartoli, S. Dubreuil, M. Panzeri, R. Lombardi, W. Lammen, M. Zhang, I. van Gent, and P. D. Ciampa. "A clustered and surrogate-based MDA use case for MDO scenarios in AGILE project". In: *2018 Multidisciplinary Analysis and Optimization Conference*. 2018. DOI: 10.2514/6.2018-3252.

[47] S. Delbecq, C. David, S. Defoort, P. Schmollgruber, E. Benard, and V. Pommier-Budinger. "From FAST to FAST-OAD: An open source framework for rapid Overall Aircraft Design". In: *10th EASN Virtual International Conference on Innovation in Aviation & Space to the Satisfaction of the European Citizens*. 2020.

[48] P. Schmollgruber, N. Bartoli, J. Bedouet, S. Defoort, Y. Gourinat, E. Benard, R. Lafage, and A. Sgueglia. "Use of a certification constraints module for aircraft design activities". In: *17th AIAA Aviation Technology, Integration, and Operations Conference*. 2017. DOI: 10.2514/6.2017-3762.

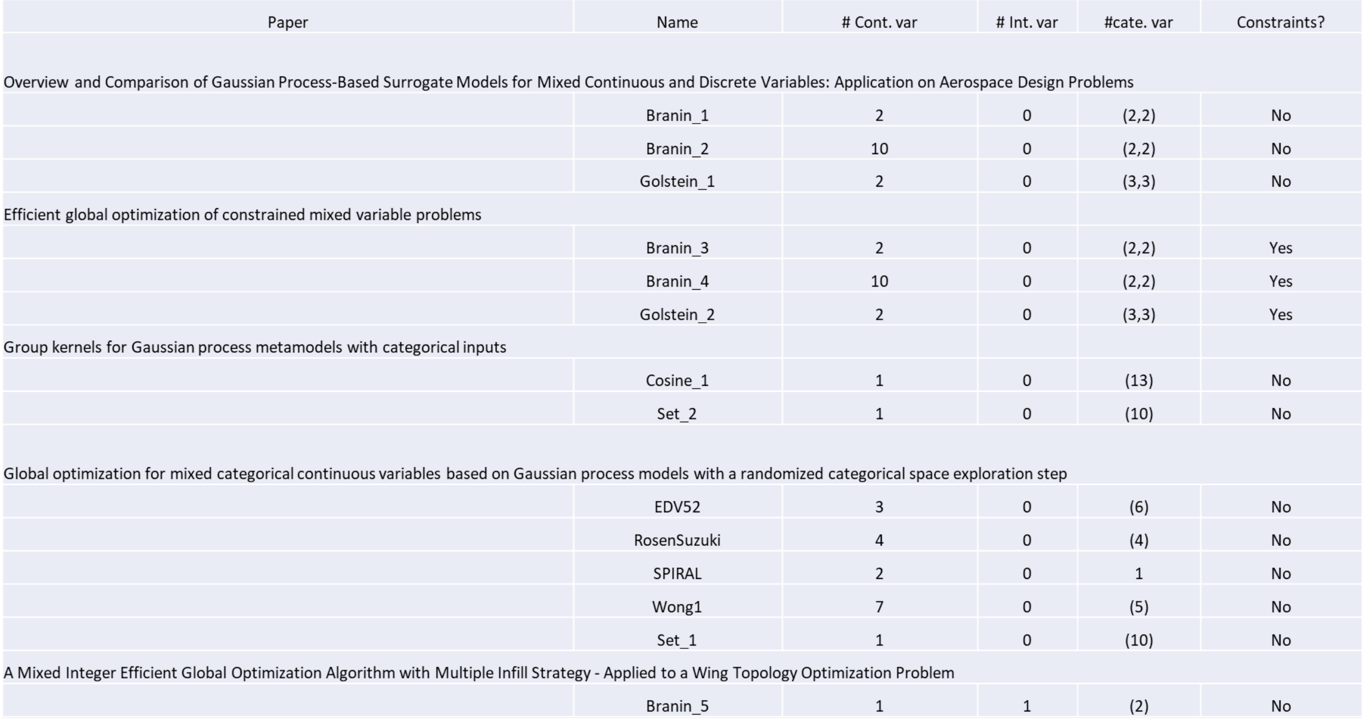
[49] D. Ginsbourger, R. Le Riche, and L. Carraro. "Kriging is well-suited to parallelize optimization". In: *Computational Intelligence in Expensive Optimization Problems*. Springer Berlin Heidelberg, 2010. ISBN: 978-3-642-10701-6.

Appendices

Appendix A

Detailed Results

The first part of this appendix will present the 11 test cases and their definition. Then, in a second part, the models and optimization results will be presented.



Paper	Name	# Cont. var	# Int. var	#cate. var	Constraints?
Overview and Comparison of Gaussian Process-Based Surrogate Models for Mixed Continuous and Discrete Variables: Application on Aerospace Design Problems					
	Branin_1	2	0	(2,2)	No
	Branin_2	10	0	(2,2)	No
	Golstein_1	2	0	(3,3)	No
Efficient global optimization of constrained mixed variable problems					
	Branin_3	2	0	(2,2)	Yes
	Branin_4	10	0	(2,2)	Yes
	Golstein_2	2	0	(3,3)	Yes
Group kernels for Gaussian process metamodels with categorical inputs					
	Cosine_1	1	0	(13)	No
	Set_2	1	0	(10)	No
Global optimization for mixed categorical continuous variables based on Gaussian process models with a randomized categorical space exploration step					
	EDV52	3	0	(6)	No
	RosenSuzuki	4	0	(4)	No
	SPIRAL	2	0	1	No
	Wong1	7	0	(5)	No
	Set_1	1	0	(10)	No
A Mixed Integer Efficient Global Optimization Algorithm with Multiple Infill Strategy - Applied to a Wing Topology Optimization Problem					
	Branin_5	1	1	(2)	No

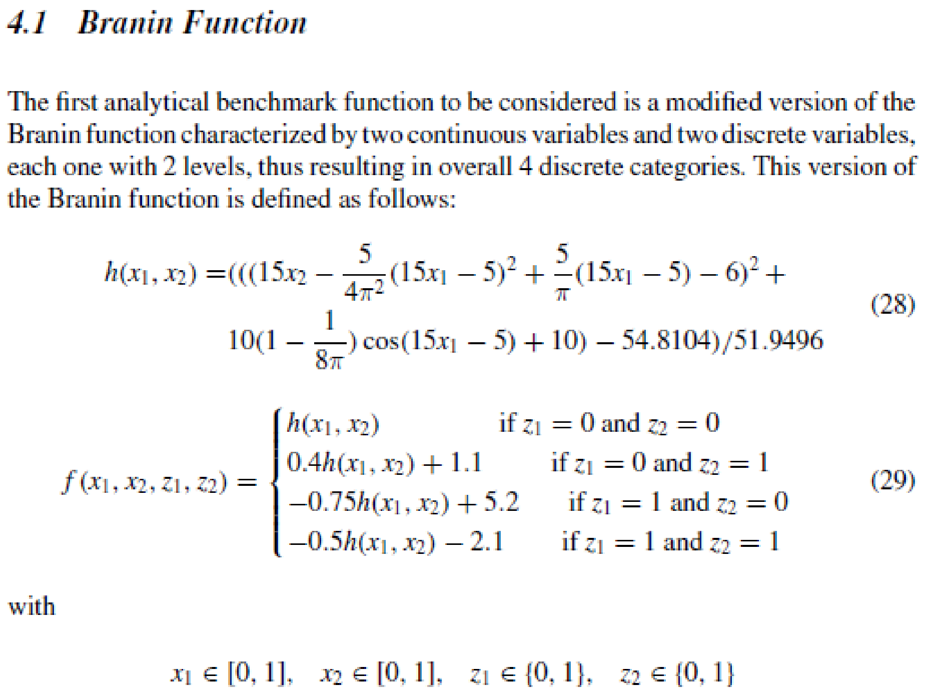
Figure A.1: Test cases overview

This table (**Figure A.1**) summarizes the test cases used that you can find in this Appendix but the cases with constraints were not evaluated during this internship.

A.1 All test cases

Branin 1

From [33]
Reference solution: -4.517207270143658



4.1 Branin Function

The first analytical benchmark function to be considered is a modified version of the Branin function characterized by two continuous variables and two discrete variables, each one with 2 levels, thus resulting in overall 4 discrete categories. This version of the Branin function is defined as follows:

$$h(x_1,x_2) = (((15x_2 - \frac{5}{4\pi^2}(15x_1-5)^2 + \frac{5}{\pi}(15x_1-5)-6)^2 + 10(1-\frac{1}{8\pi})\cos(15x_1-5)+10) - 54.8104)/51.9496 \tag{28}$$

$$f(x_1,x_2,z_1,z_2) = \begin{cases} h(x_1,x_2) & \text{if } z_1 = 0 \text{ and } z_2 = 0 \\ 0.4h(x_1,x_2)+1.1 & \text{if } z_1 = 0 \text{ and } z_2 = 1 \\ -0.75h(x_1,x_2)+5.2 & \text{if } z_1 = 1 \text{ and } z_2 = 0 \\ -0.5h(x_1,x_2)-2.1 & \text{if } z_1 = 1 \text{ and } z_2 = 1 \end{cases} \tag{29}$$

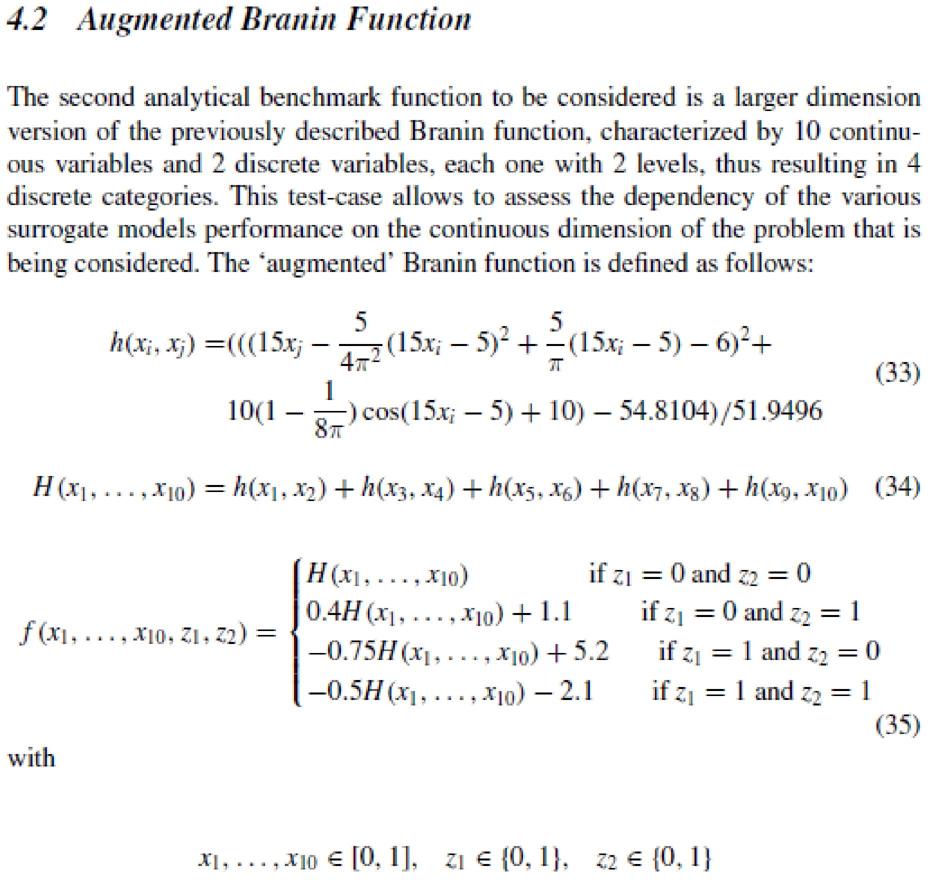
with

$$x_1 \in [0,1], \quad x_2 \in [0,1], \quad z_1 \in \{0,1\}, \quad z_2 \in \{0,1\}$$

Figure A.2: Branin_1 case

Branin 2

From [33]
Reference solution: Reference : -14.186036350718284



4.2 *Augmented Branin Function*

The second analytical benchmark function to be considered is a larger dimension version of the previously described Branin function, characterized by 10 continuous variables and 2 discrete variables, each one with 2 levels, thus resulting in 4 discrete categories. This test-case allows to assess the dependency of the various surrogate models performance on the continuous dimension of the problem that is being considered. The 'augmented' Branin function is defined as follows:

$$\begin{aligned} h(x_i, x_j) =& (((15x_j - \frac{5}{4\pi^2}(15x_i - 5)^2 + \frac{5}{\pi}(15x_i - 5) - 6)^2 + \\ & 10(1 - \frac{1}{8\pi}) \cos(15x_i - 5) + 10) - 54.8104)/51.9496 \end{aligned} \tag{33}$$

$$H(x_1, \dots, x_{10}) = h(x_1, x_2) + h(x_3, x_4) + h(x_5, x_6) + h(x_7, x_8) + h(x_9, x_{10}) \tag{34}$$

$$f(x_1, \dots, x_{10}, z_1, z_2) = \begin{cases} H(x_1, \dots, x_{10}) & \text{if } z_1 = 0 \text{ and } z_2 = 0 \\ 0.4H(x_1, \dots, x_{10}) + 1.1 & \text{if } z_1 = 0 \text{ and } z_2 = 1 \\ -0.75H(x_1, \dots, x_{10}) + 5.2 & \text{if } z_1 = 1 \text{ and } z_2 = 0 \\ -0.5H(x_1, \dots, x_{10}) - 2.1 & \text{if } z_1 = 1 \text{ and } z_2 = 1 \end{cases} \tag{35}$$

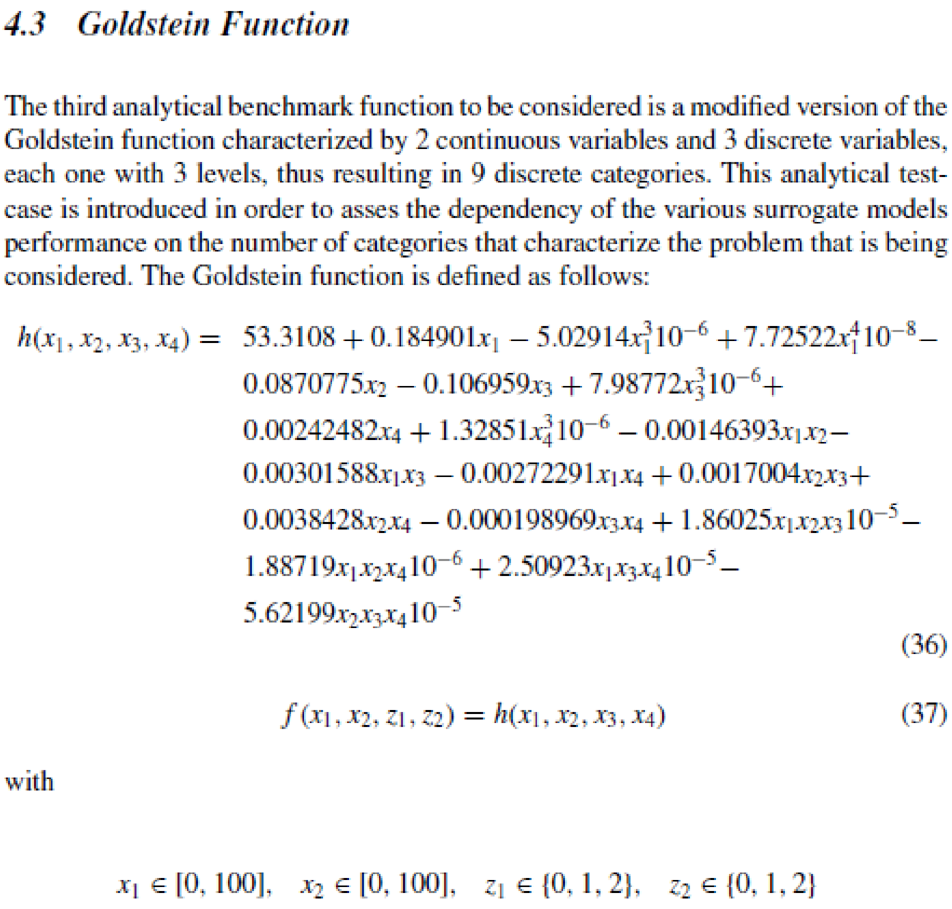
with

$$x_1, \dots, x_{10} \in [0, 1], \quad z_1 \in \{0, 1\}, \quad z_2 \in \{0, 1\}$$

Figure A.3: Branin_2 case

Goldstein 1

From [33]
Reference solution : 38.08474427



4.3 Goldstein Function

The third analytical benchmark function to be considered is a modified version of the Goldstein function characterized by 2 continuous variables and 3 discrete variables, each one with 3 levels, thus resulting in 9 discrete categories. This analytical test-case is introduced in order to asses the dependency of the various surrogate models performance on the number of categories that characterize the problem that is being considered. The Goldstein function is defined as follows:

$$\begin{aligned} h(x_1, x_2, x_3, x_4) = \; & 53.3108 + 0.184901x_1 - 5.02914x_1^3 10^{-6} + 7.72522x_1^4 10^{-8} - \\ & 0.0870775x_2 - 0.106959x_3 + 7.98772x_3^3 10^{-6} + \\ & 0.00242482x_4 + 1.32851x_4^3 10^{-6} - 0.00146393x_1x_2 - \\ & 0.00301588x_1x_3 - 0.00272291x_1x_4 + 0.0017004x_2x_3 + \\ & 0.0038428x_2x_4 - 0.000198969x_3x_4 + 1.86025x_1x_2x_3 10^{-5} - \\ & 1.88719x_1x_2x_4 10^{-6} + 2.50923x_1x_3x_4 10^{-5} - \\ & 5.62199x_2x_3x_4 10^{-5} \end{aligned} \tag{36}$$

$$f(x_1, x_2, z_1, z_2) = h(x_1, x_2, x_3, x_4) \tag{37}$$

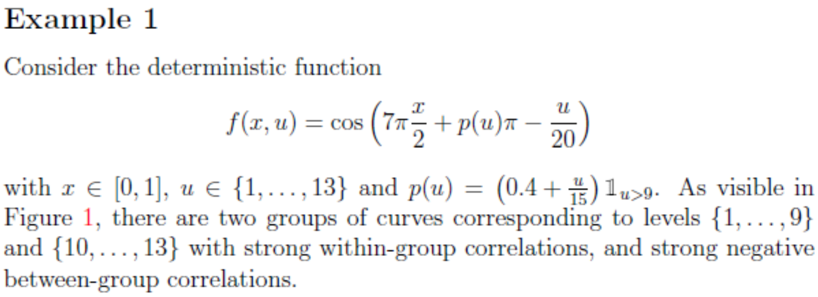
with

$$x_1 \in [0, 100], \quad x_2 \in [0, 100], \quad z_1 \in \{0, 1, 2\}, \quad z_2 \in \{0, 1, 2\}$$

Figure A.4: Goldstein_1 case

Cosine 1

From [31]
Reference solution: -1



Example 1

Consider the deterministic function

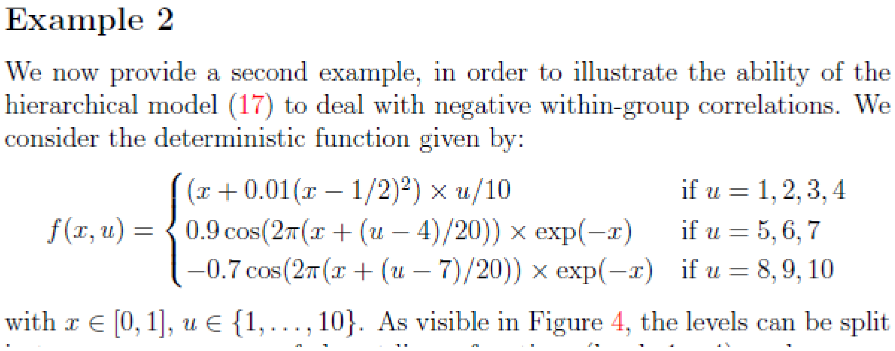
$$f(x, u) = \cos\left(7\pi\frac{x}{2} + p(u)\pi - \frac{u}{20}\right)$$

with $x \in [0, 1]$, $u \in \{1, \dots, 13\}$ and $p(u) = \left(0.4 + \frac{u}{15}\right)\mathbb{1}_{u>9}$. As visible in Figure 1, there are two groups of curves corresponding to levels $\{1, \dots, 9\}$ and $\{10, \dots, 13\}$ with strong within-group correlations, and strong negative between-group correlations.

Figure A.5: Cosine_1 case

Set 2

From [31]
Reference solution: -0.6657395614066074



Example 2

We now provide a second example, in order to illustrate the ability of the hierarchical model (17) to deal with negative within-group correlations. We consider the deterministic function given by:

$$f(x,u) = \begin{cases} (x + 0.01(x-1/2)^2) \times u/10 & \text{if } u = 1,2,3,4 \\ 0.9\cos(2\pi(x+(u-4)/20)) \times \exp(-x) & \text{if } u = 5,6,7 \\ -0.7\cos(2\pi(x+(u-7)/20)) \times \exp(-x) & \text{if } u = 8,9,10 \end{cases}$$

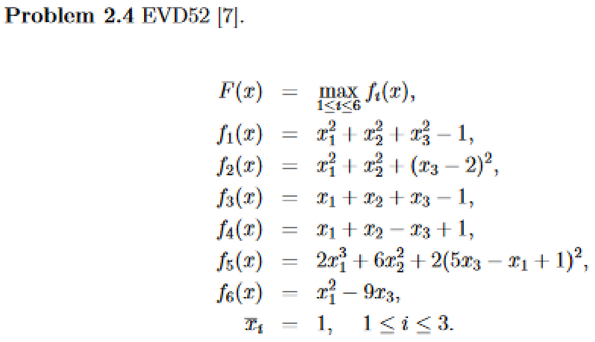
with $x \in [0,1]$, $u \in \{1,\ldots,10\}$. As visible in Figure 4, the levels can be split

Figure A.6: Set_2 case

EVD52

From [42]
The modification made by Zuniga and Sinoquet is to considerate a categorical variable which select between f_i function.
We made the assumption that the continuous variables are between [-5,5] because the original problem was left unbounded.

Reference solution :-249.43247170013404



Problem 2.4 EVD52 [7].

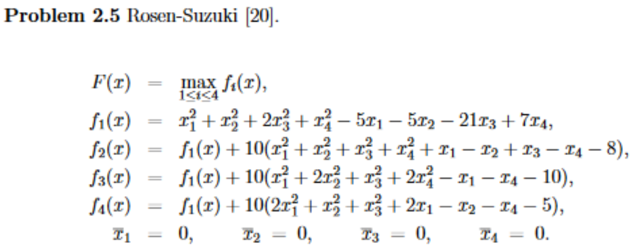
$$\begin{aligned} F(x) &= \max_{1\le i\le 6} f_i(x), \\ f_1(x) &= x_1^2 + x_2^2 + x_3^2 - 1, \\ f_2(x) &= x_1^2 + x_2^2 + (x_3-2)^2, \\ f_3(x) &= x_1 + x_2 + x_3 - 1, \\ f_4(x) &= x_1 + x_2 - x_3 + 1, \\ f_5(x) &= 2x_1^3 + 6x_2^2 + 2(5x_3 - x_1 + 1)^2, \\ f_6(x) &= x_1^2 - 9x_3, \\ \bar{x}_i &= 1, \quad 1 \le i \le 3. \end{aligned}$$

Figure A.7: EDV52 case

Rosen-Suzuki

From [42]
The modification made by Zuniga and Sinoquet is to considerate a categorical variable which select between f_i function.
We made the assumption that the continuous variables are between [-5,5] because the original problem was left unbounded.

Reference solution: 245



Problem 2.5 Rosen-Suzuki [20].

$$\begin{aligned}
F(x) &= \max_{1\leq i\leq 4} f_i(x),\\
f_1(x) &= x_1^2+x_2^2+2x_3^2+x_4^2-5x_1-5x_2-21x_3+7x_4,\\
f_2(x) &= f_1(x)+10(x_1^2+x_2^2+x_3^2+x_4^2+x_1-x_2+x_3-x_4-8),\\
f_3(x) &= f_1(x)+10(x_1^2+2x_2^2+x_3^2+2x_4^2-x_1-x_4-10),\\
f_4(x) &= f_1(x)+10(2x_1^2+x_2^2+x_3^2+2x_1-x_2-x_4-5),\\
\overline{x}_1 &= 0, \qquad \overline{x}_2 = 0, \qquad \overline{x}_3 = 0, \qquad \overline{x}_4 = 0.
\end{aligned}$$

Figure A.8: Rosen-Suzuki case

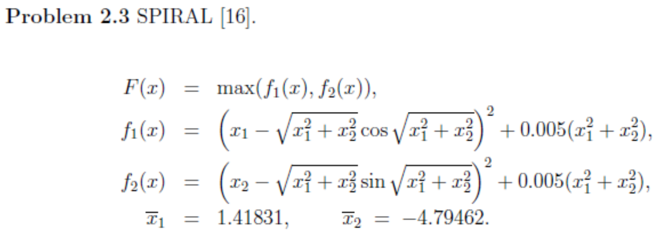
Spiral

From [42]

The modification made by Zuniga and Sinoquet is to considerate a categorical variable which select between f_i function.

We made the assumption that the continuous variables are between [-5,5] because the original problem was left unbounded.

Reference solution: 0



Problem 2.3 SPIRAL [16].

$$\begin{aligned}
F(x) &= \max(f_1(x), f_2(x)),\\
f_1(x) &= \left(x_1-\sqrt{x_1^2+x_2^2}\cos\sqrt{x_1^2+x_2^2}\right)^2+0.005(x_1^2+x_2^2),\\
f_2(x) &= \left(x_2-\sqrt{x_1^2+x_2^2}\sin\sqrt{x_1^2+x_2^2}\right)^2+0.005(x_1^2+x_2^2),\\
\overline{x}_1 &= 1.41831, \qquad \overline{x}_2 = -4.79462.
\end{aligned}$$

Figure A.9: Spiral case

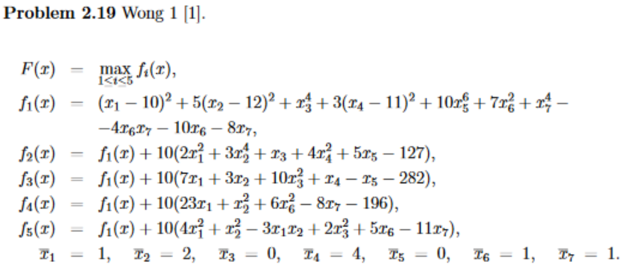
Wong 1

From [42]

The modification made by Zuniga and Sinoquet is to considerate a categorical variable which select between f_i function.

We made the assumption that the continuous variables are between [-5,5] because the original problem was left unbounded.

Reference solution: -2389.627479486391



Problem 2.19 Wong 1 [1].

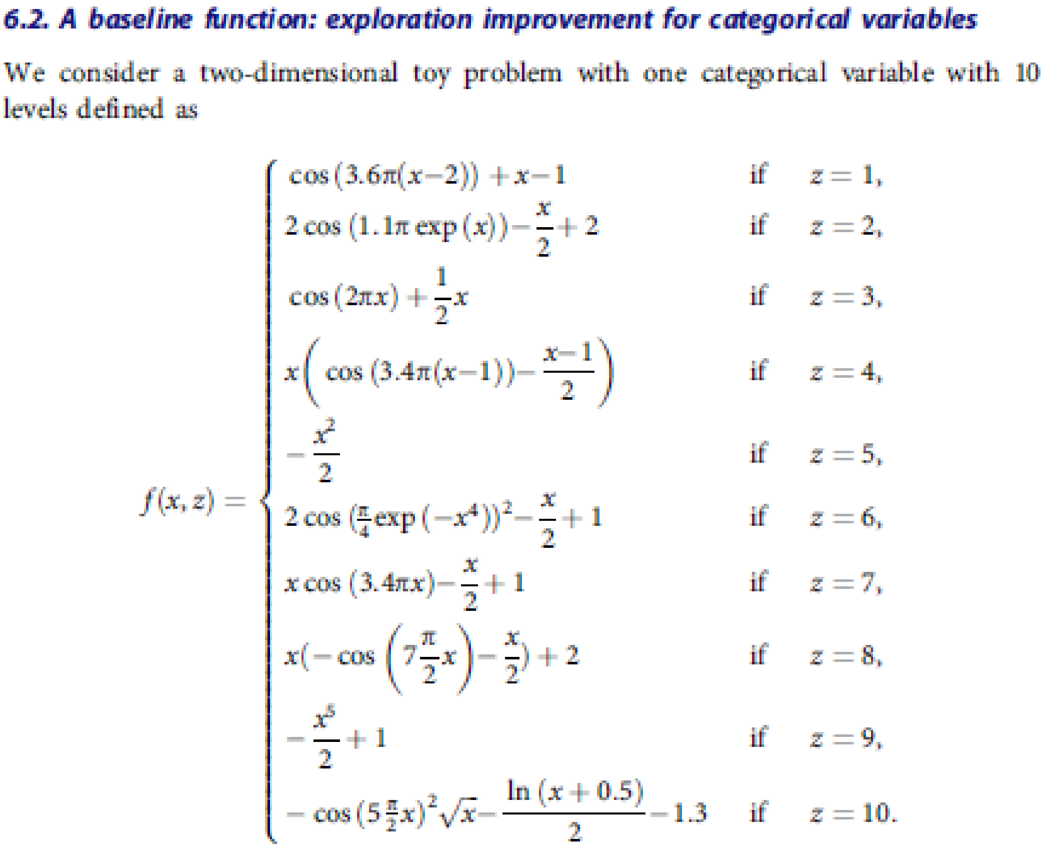
$$
\begin{aligned}
F(x) &= \max_{1<i<5} f_i(x), \\
f_1(x) &= (x_1-10)^2+5(x_2-12)^2+x_3^4+3(x_4-11)^2+10x_5^6+7x_6^2+x_7^4- \\
&\quad -4x_6x_7-10x_6-8x_7, \\
f_2(x) &= f_1(x)+10(2x_1^2+3x_2^4+x_3+4x_4^2+5x_5-127), \\
f_3(x) &= f_1(x)+10(7x_1+3x_2+10x_3^2+x_4-x_5-282), \\
f_4(x) &= f_1(x)+10(23x_1+x_2^2+6x_6^2-8x_7-196), \\
f_5(x) &= f_1(x)+10(4x_1^2+x_2^2-3x_1x_2+2x_3^2+5x_6-11x_7), \\
\overline{x}_1 &= 1, \quad \overline{x}_2 = 2, \quad \overline{x}_3 = 0, \quad \overline{x}_4 = 4, \quad \overline{x}_5 = 0, \quad \overline{x}_6 = 1, \quad \overline{x}_7 = 1.
\end{aligned}
$$

Figure A.10: Spiral case

Set 1

From [42]
The modification made by Zuniga and Sinoquet is to considerate a categorical variable which select between f_i function.
We made the assumption that the continuous variables are between [-5,5] because the original problem was left unbounded.

Reference solution: -2.3296038673969974



6.2. A baseline function: exploration improvement for categorical variables

We consider a two-dimensional toy problem with one categorical variable with 10 levels defined as

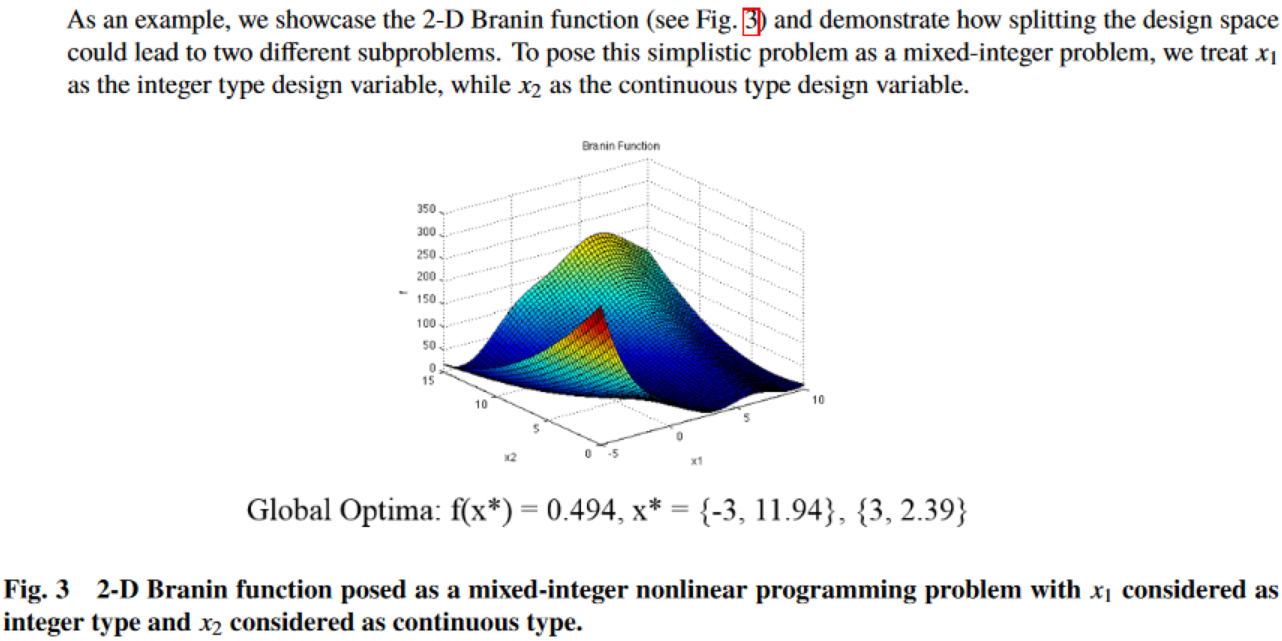
$$
f(x,z) = \begin{cases}
\cos(3.6\pi(x-2)) + x - 1 & \text{if} \quad z=1, \\
2\cos(1.1\pi \exp(x)) - \frac{x}{2} + 2 & \text{if} \quad z=2, \\
\cos(2\pi x) + \frac{1}{2}x & \text{if} \quad z=3, \\
x\left(\cos(3.4\pi(x-1)) - \frac{x-1}{2}\right) & \text{if} \quad z=4, \\
-\frac{x^2}{2} & \text{if} \quad z=5, \\
2\cos\left(\frac{\pi}{4}\exp(-x^4)\right)^2 - \frac{x}{2} + 1 & \text{if} \quad z=6, \\
x\cos(3.4\pi x) - \frac{x}{2} + 1 & \text{if} \quad z=7, \\
x(-\cos\left(7\frac{\pi}{2}x\right) - \frac{x}{2}) + 2 & \text{if} \quad z=8, \\
-\frac{x^5}{2} + 1 & \text{if} \quad z=9, \\
-\cos\left(5\frac{\pi}{2}x\right)^2\sqrt{x} - \frac{\ln(x+0.5)}{2} - 1.3 & \text{if} \quad z=10.
\end{cases}
$$

Figure A.11: Set 1 case

Branin 5

From [23]

Reference value: 0.49398053



As an example, we showcase the 2-D Branin function (see Fig. 3) and demonstrate how splitting the design space could lead to two different subproblems. To pose this simplistic problem as a mixed-integer problem, we treat x_1 as the integer type design variable, while x_2 as the continuous type design variable.

Branin Function
350
300
250
200
150
100
50
0
f
15
10
5
0
x2
-5
0
5
10
x1

Global Optima: f(x*) = 0.494, x* = {-3, 11.94}, {3, 2.39}

Fig. 3 2-D Branin function posed as a mixed-integer nonlinear programming problem with x_1 considered as integer type and x_2 considered as continuous type.

Figure A.12: Branin 5 case

A.2 All results

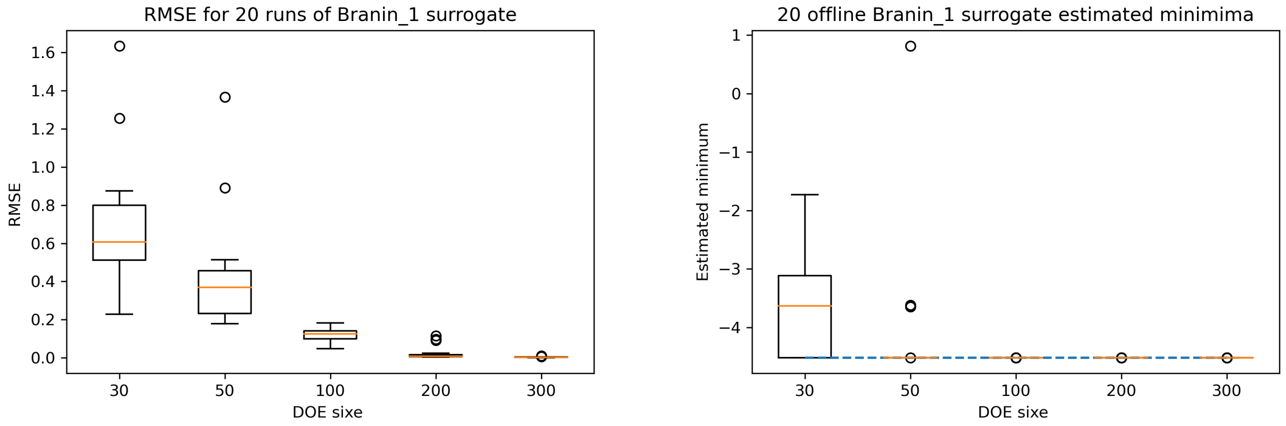


Figure A.13: Branin_1 Model

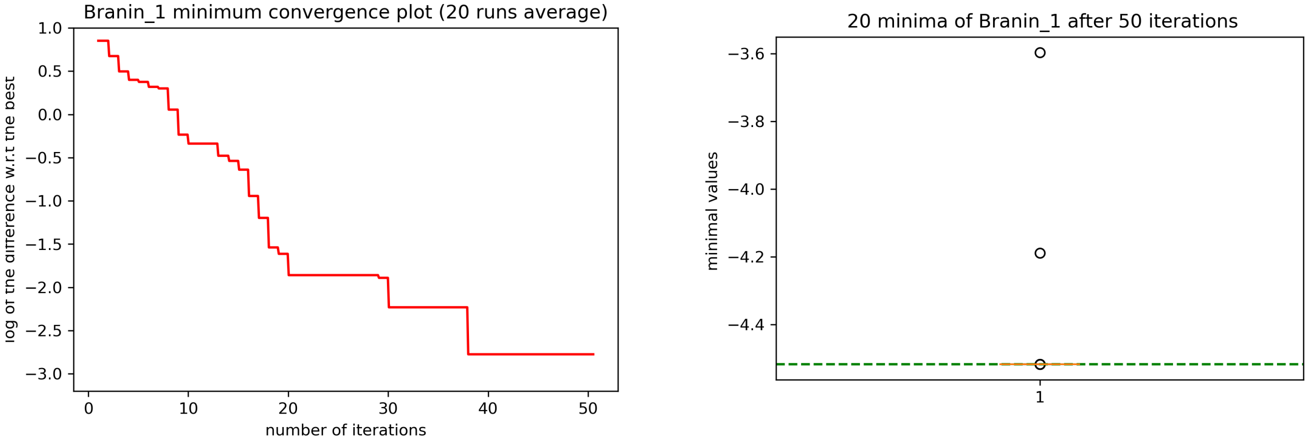


Figure A.14: Branin_1 Optimization

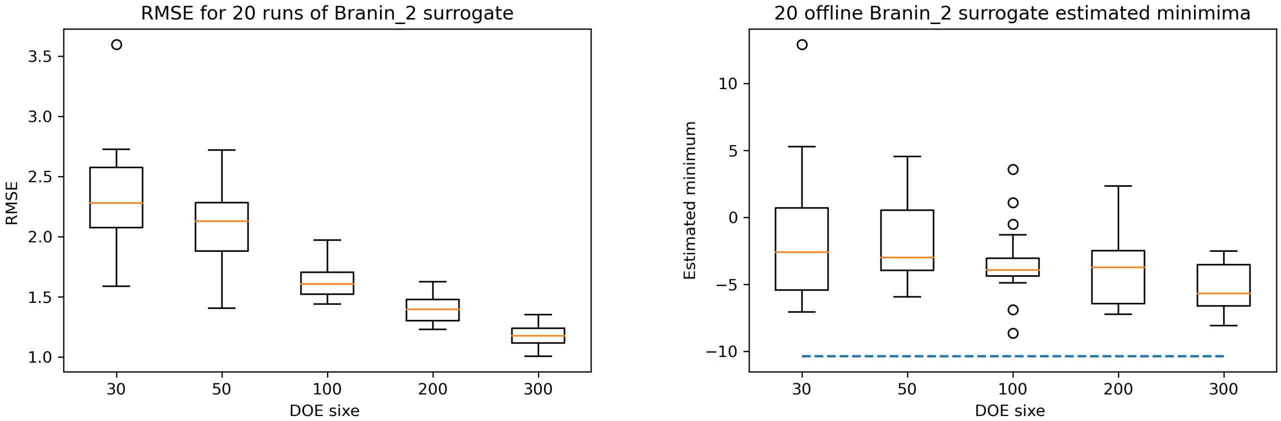


Figure A.15: Branin_2 Model

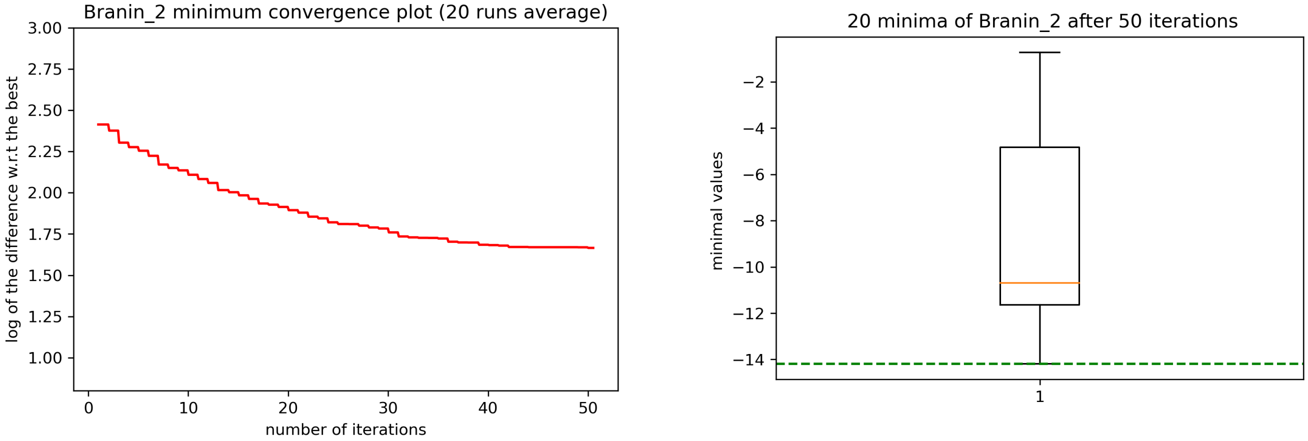


Figure A.16: Branin_2 Optimization

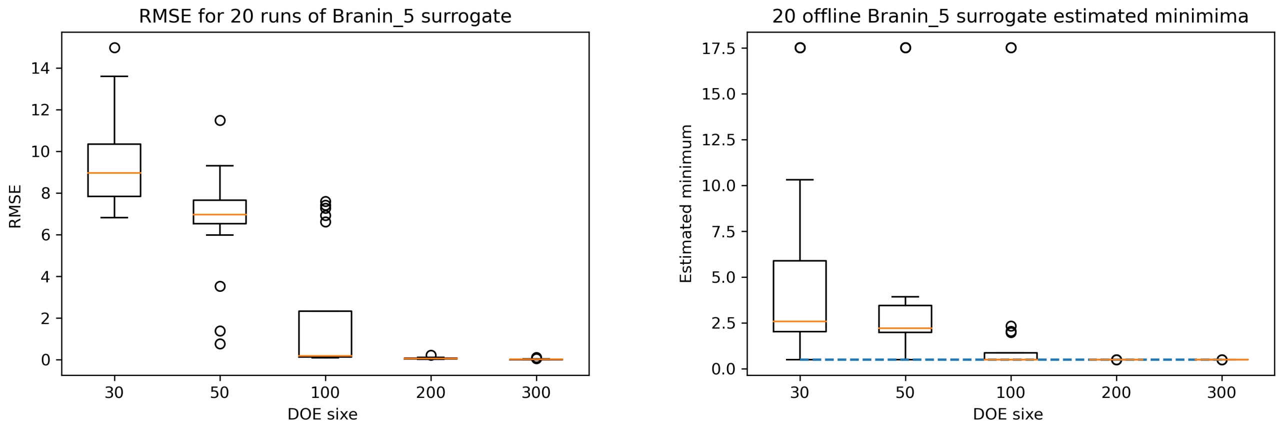


Figure A.17: Branin_5 Model

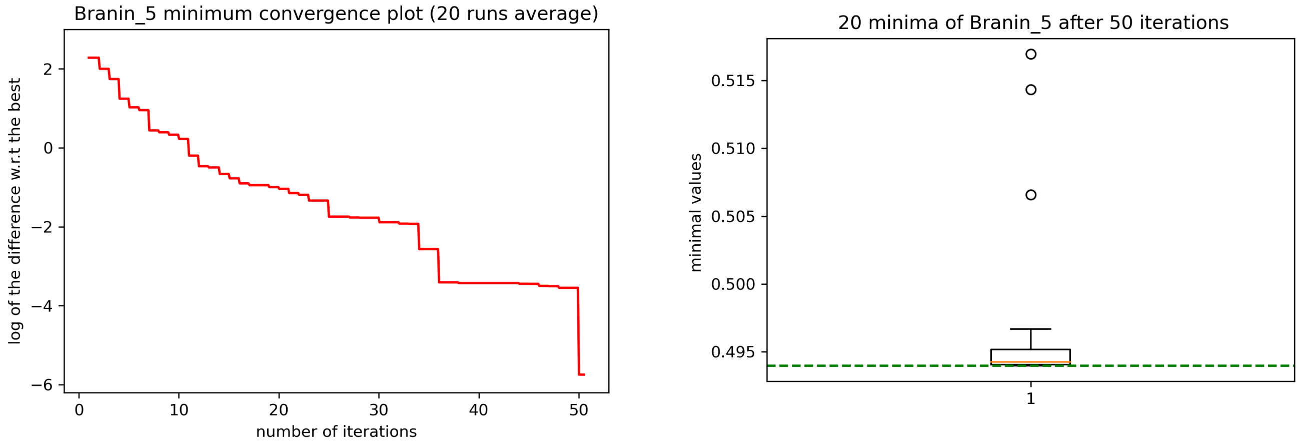


Figure A.18: Branin_5 Optimization

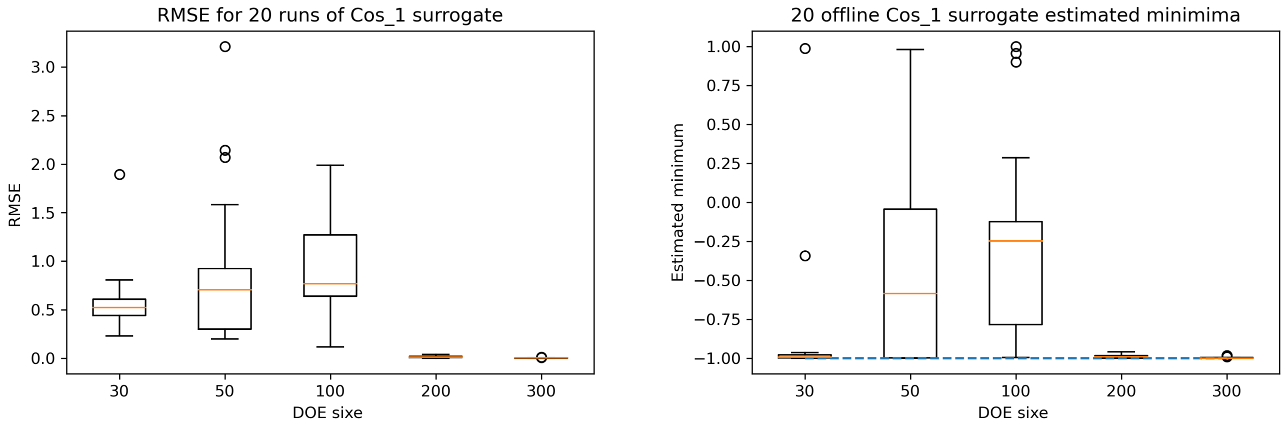


Figure A.19: Cosine_1 Model

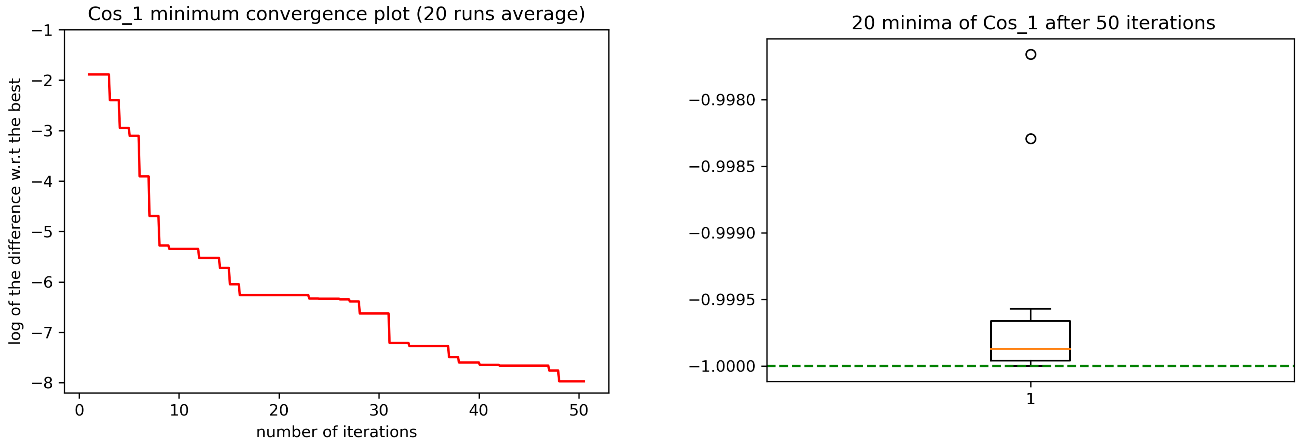


Figure A.20: Cosine_1 Optimization

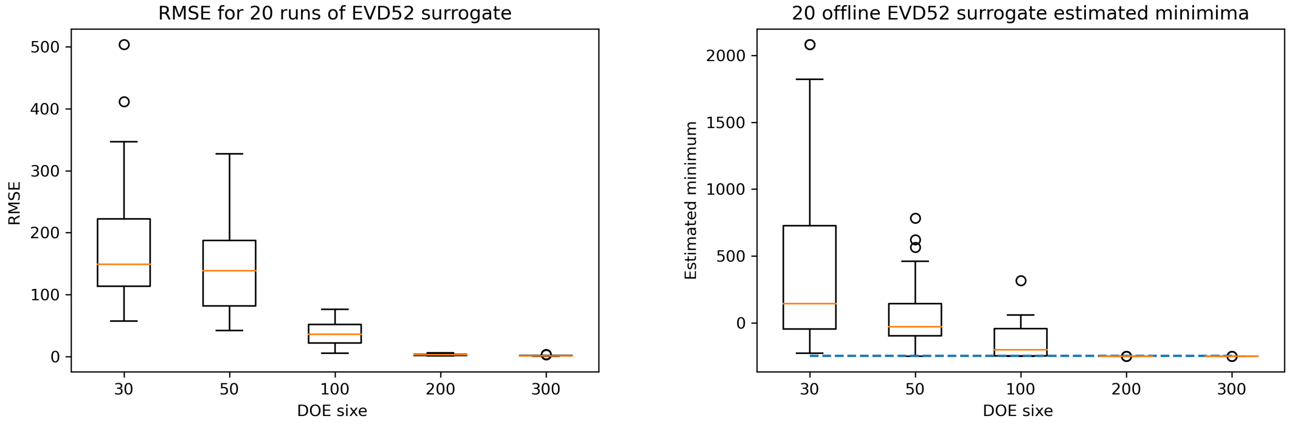


Figure A.21: EVD52 Model

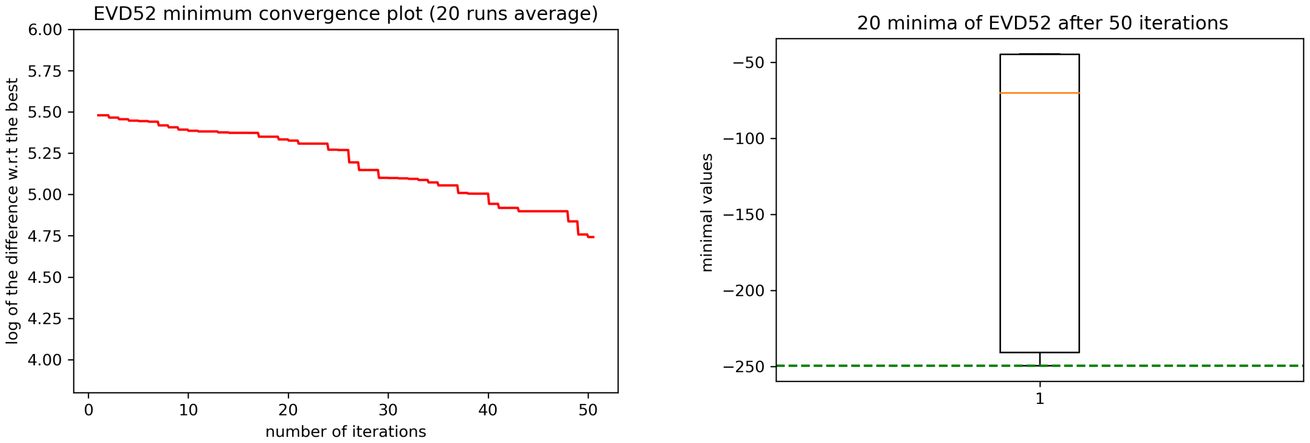


Figure A.22: EVD52 Optimization

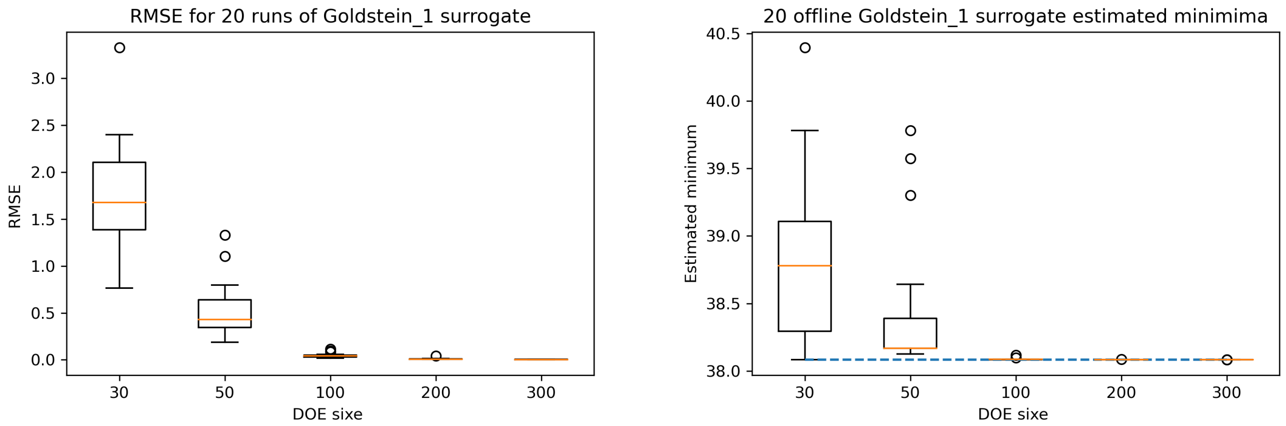


Figure A.23: Goldstein_1 Model

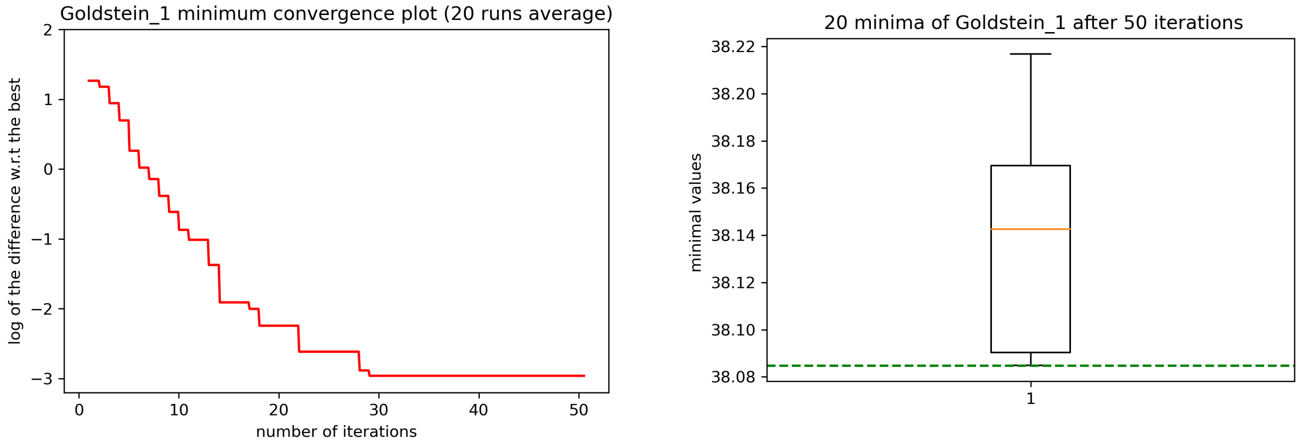


Figure A.24: Goldstein_1 Optimization

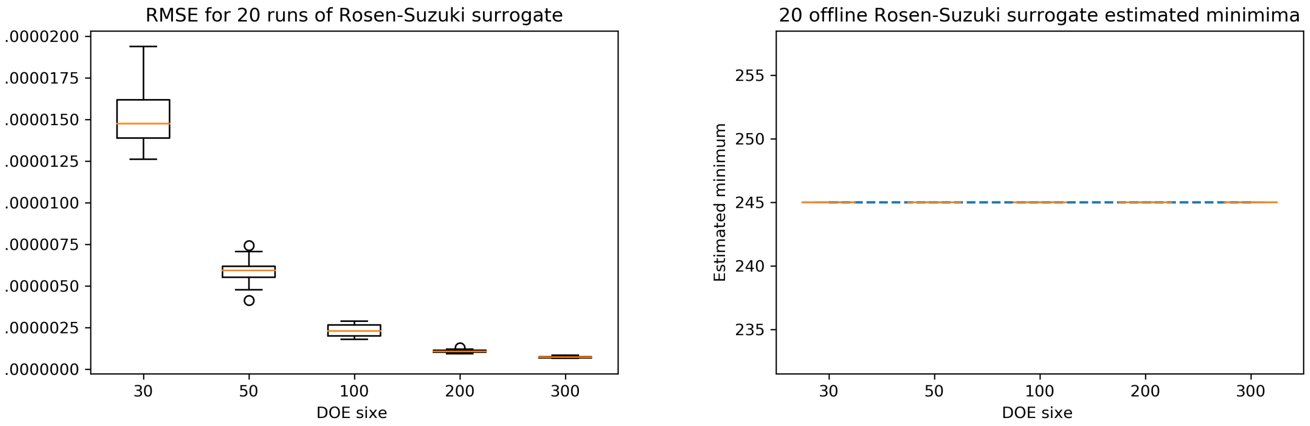


Figure A.25: Rosen-Suzuki Model

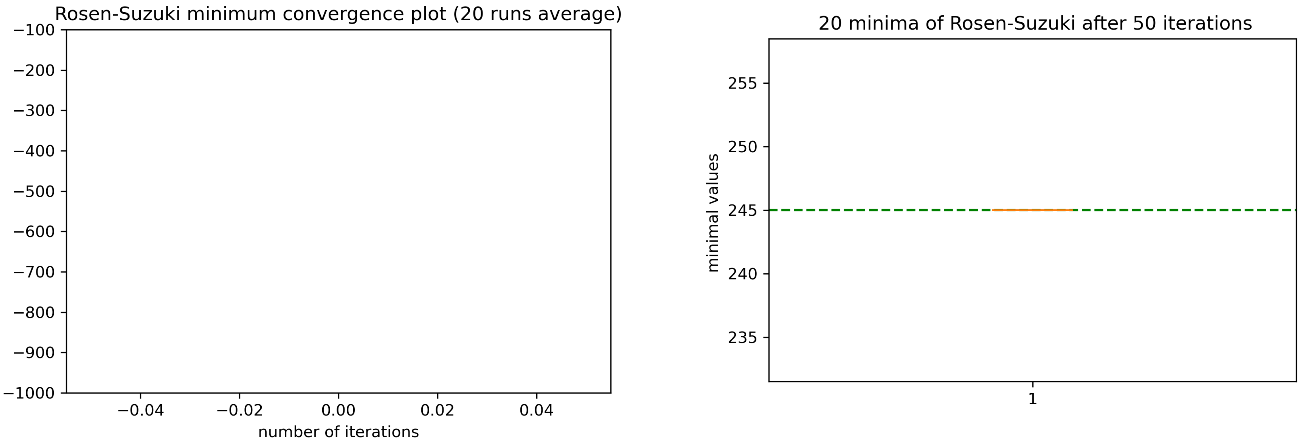


Figure A.26: Rosen-Suzuki Optimization

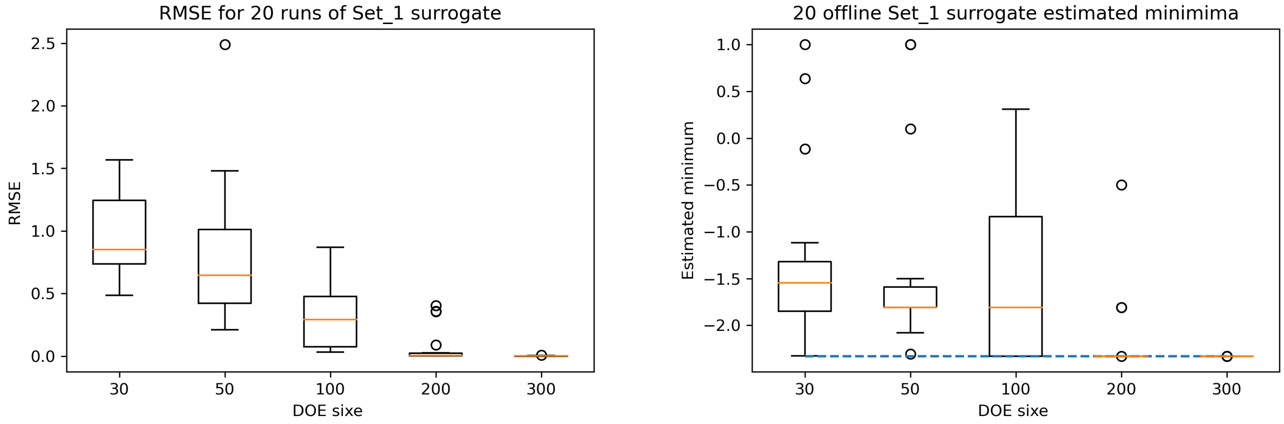


Figure A.27: Set_1 Model

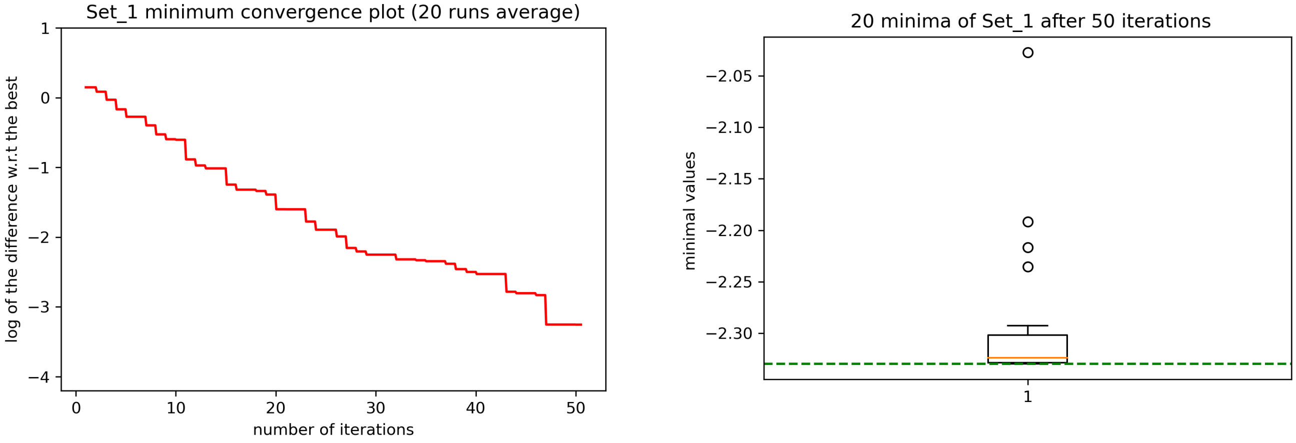


Figure A.28: Set_1 Optimization

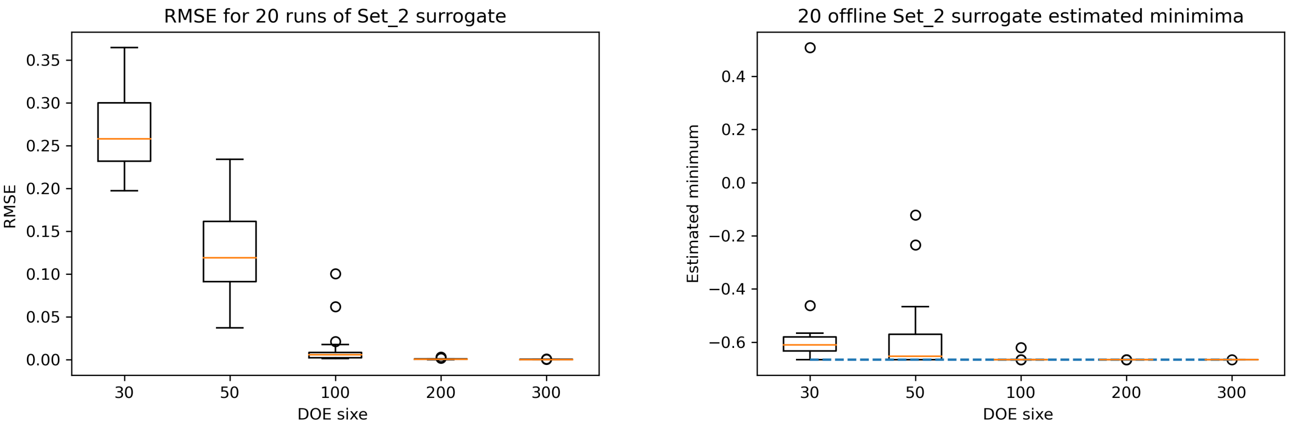


Figure A.29: Set_2 Model

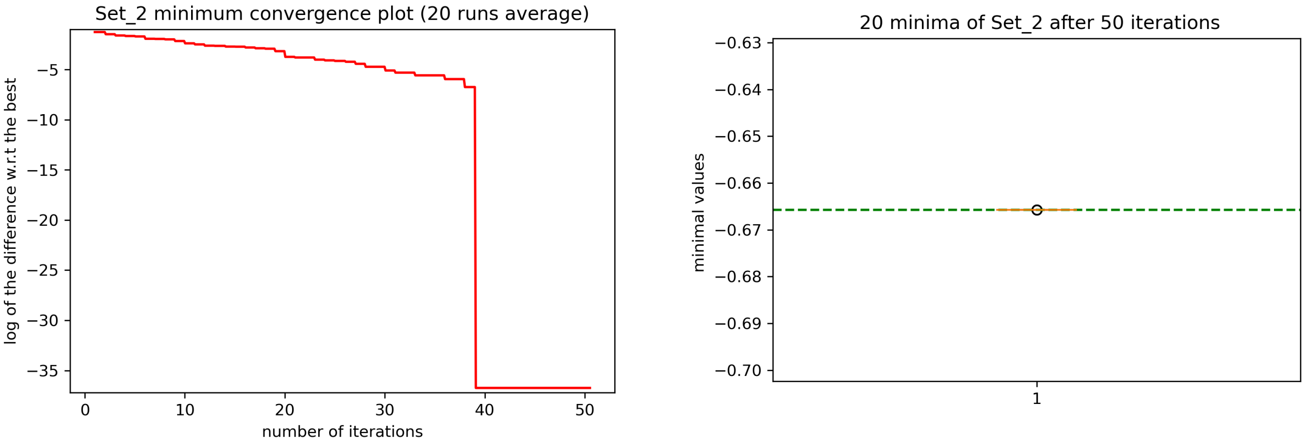


Figure A.30: Set_2 Optimization

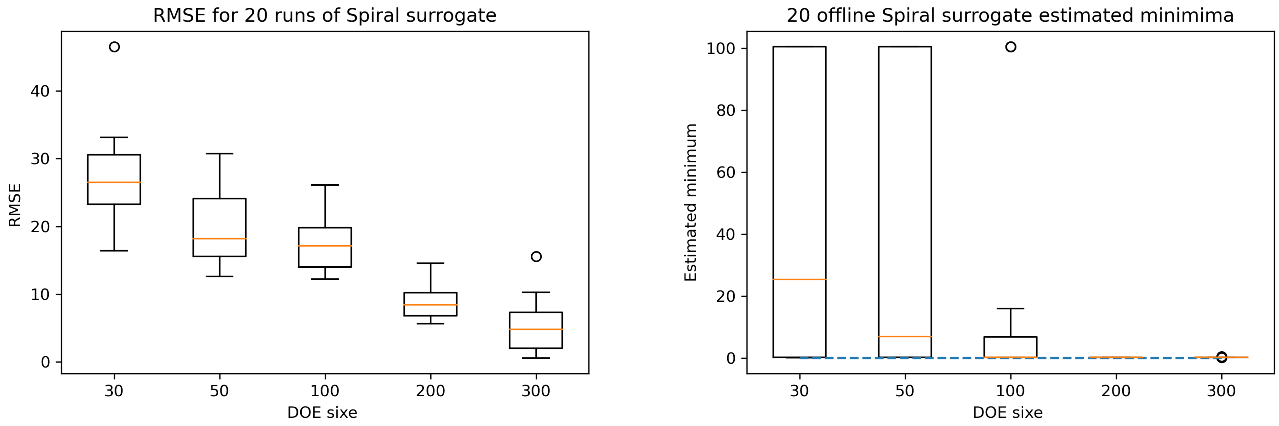


Figure A.31: Spiral Model

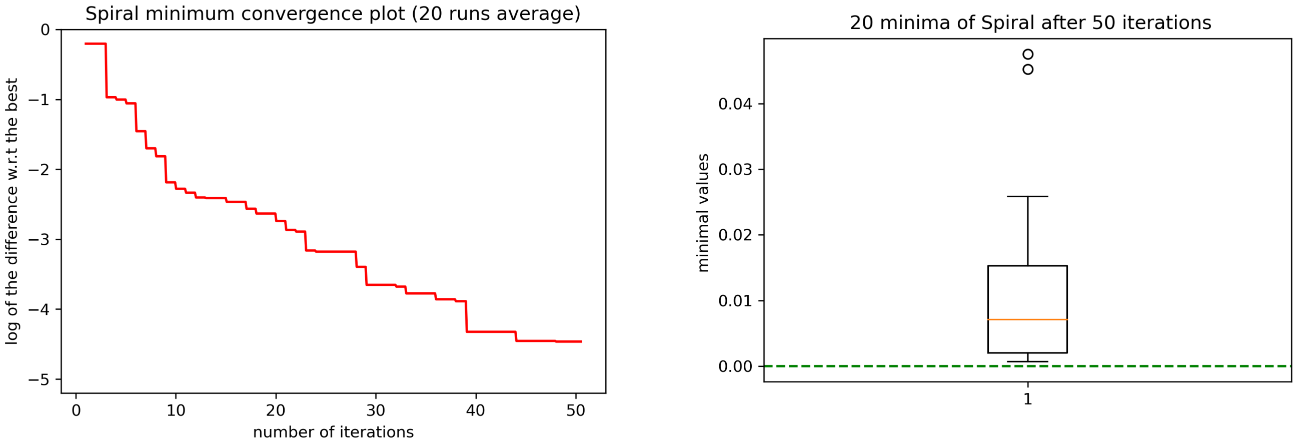


Figure A.32: Spiral Optimization

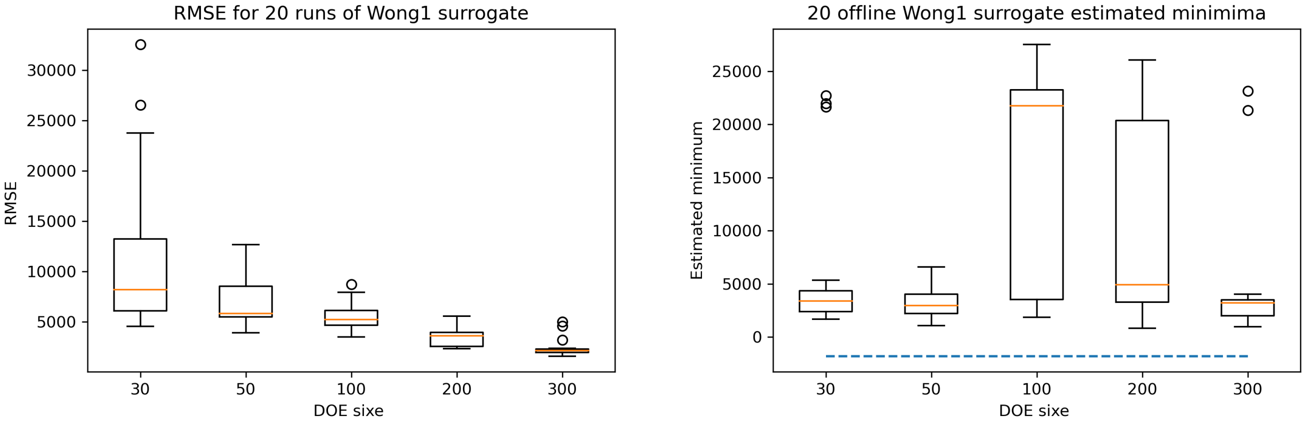


Figure A.33: Wong Model

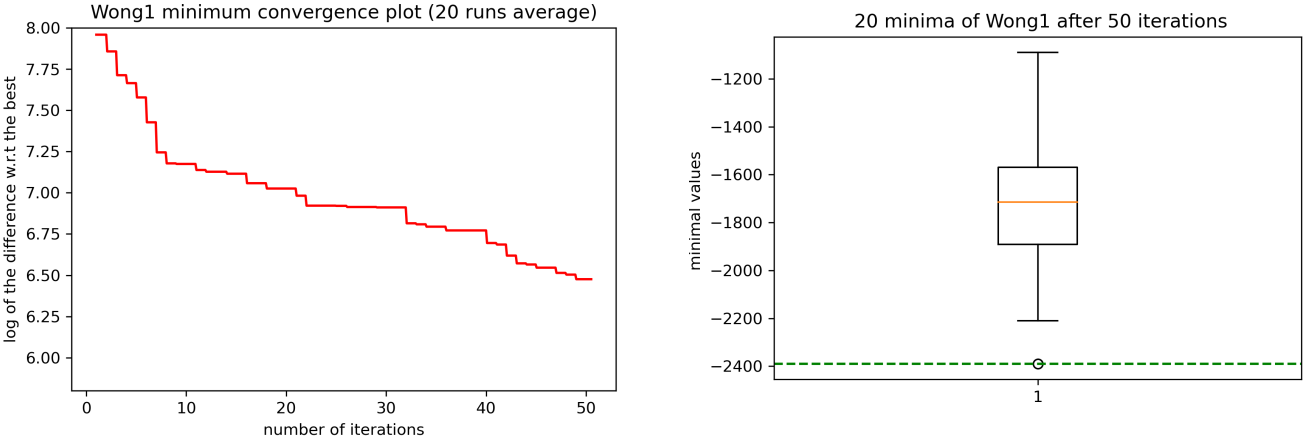


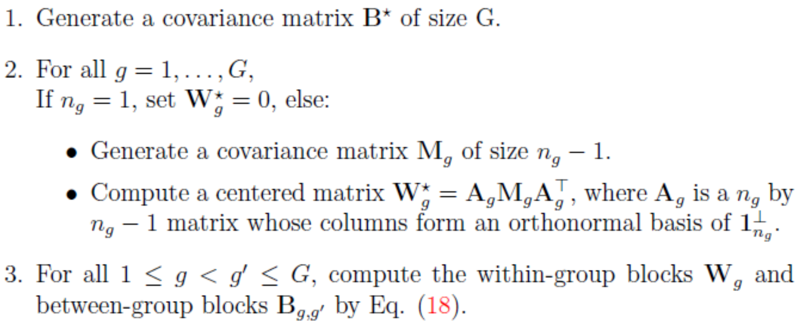
Figure A.34: Wong Optimization

Appendix B

Roustant et al. - Theorems about how to construct matrices

Below are the two results that one's used to generate Generalized Compound Symmetry matrices under the centered (zero mean) assumption.

1. Assume that $\lambda_{g/.}$ are of zero mean, the covariance matrix of η is a GCS block matrix with $\forall(g,g') \in \{1..G\}^2 : W_g = B^*_{g,g'} J_{n_g} + W^*_g$ *and* $B_{g,g'} = B^*_{g,g'} J_{n_g,n_{g'}}$ *where* $W^*_g = Cov(\lambda_{g/.}|\bar{\lambda}_{g/.} = 0)$ is a centered PSD matrix. Conversely, let T be a PSD CGS block matrix, T is the covariance of η with zero mean error if $B^* = \widetilde{T}$ and $Cov(\lambda_{g/.}|\bar{\lambda}_{g/.} = 0) = W_g - \bar{W}_g J_g$ where $\widetilde{T}$ is the G× G matrix obtained by averaging each block of T.
2. T is PSD iif $\widetilde{T}$ is PSD. T is positive definite iif $\widetilde{T}$ and all the diagonal blocks Wg are positive definite.



1. Generate a covariance matrix $\mathbf{B}^\star$ of size G.
2. For all $g = 1, \ldots, G$,
   If $n_g = 1$, set $\mathbf{W}^\star_g = 0$, else:
   - Generate a covariance matrix $\mathbf{M}_g$ of size $n_g - 1$.
   - Compute a centered matrix $\mathbf{W}^\star_g = \mathbf{A}_g \mathbf{M}_g \mathbf{A}_g^\top$, where $\mathbf{A}_g$ is a n_g by $n_g - 1$ matrix whose columns form an orthonormal basis of $\mathbf{1}^\perp_{n_g}$.
3. For all $1 \leq g < g' \leq G$, compute the within-group blocks $\mathbf{W}_g$ and between-group blocks $\mathbf{B}_{g,g'}$ by Eq. (18).

Figure B.1: Pratical usage to build a GCS matrix [31]

where "Eq. 18" is the theorem 1.

Appendix C

qEI in SMT

For the optimization to run faster, one's could seek to execute parallel optimizations. The Expected improvement (EI) is extended to proposed q new sampling points instead of one, Ginsbourger et al. [49] called this criterion the qEI criterion.
As explained in the documentation of SMT, the basic idea is to run q iterations of the EGO algorithm and to set temporally the response $\hat{y_q}$ of the q new sampling points to a virtual value.

To do so, 4 strategies have been implemented:

- The minimum constant liar (CLmin) strategy: $\hat{y_q} = min(Y)$
- The Kriging believer (KB) strategy: $\hat{y_q} = \mu(x_q)$
- The KB Upper Bound (KBUB): $\hat{y_q} = \mu(x_q) + 3.\sigma(x_q)$
- The KB Lower Bound (KBLB): $\hat{y_q} = \mu(x_q) - 3.\sigma(x_q)$