



HAL
open science

Categorical-Continuous Bayesian optimization applied to chemical reactions

Theo Rabut, Hamamache Kheddouci, Thomas Galeandro-Diamant

► **To cite this version:**

Theo Rabut, Hamamache Kheddouci, Thomas Galeandro-Diamant. Categorical-Continuous Bayesian optimization applied to chemical reactions. *Communications in Computer and Information Science*, 2022, 1684, pp 226-239. hal-04720486

HAL Id: hal-04720486

<https://hal.science/hal-04720486v1>

Submitted on 8 Oct 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Categorical-Continuous Bayesian optimization applied to chemical reactions

Theo Rabut¹, Hamamache Kheddouci¹, and Thomas Galeandro-Diamant^{2,3}

¹ Université de Lyon, Université Lyon 1, LIRIS UMR CNRS 5205, F-69621, Lyon, France

`theo.rabut@univ-lyon1.fr`

² ChemIntelligence, Lyon, France

³ DeepMatter, Glasgow, United Kingdom

Abstract. Chemical reaction optimization is a challenging task for the industry. Its purpose is to experimentally find reaction parameters (e.g. temperature, concentration, pressure) that maximize or minimize a set of objectives (e.g. yield or selectivity of the chemical reaction). These experiments are often expensive and long (up to several days), making the use of modern optimization methods more and more attractive for chemistry scientists.

Recently, Bayesian optimization has been shown to outperform human decision-making for the optimization of chemical reactions [16]. It is well-suited for chemical reaction optimization problems, for which the evaluation is expensive and noisy.

In this paper we address the problem of chemical reaction optimization with continuous and categorical variables.

We propose a Bayesian optimization method that uses a covariance function specifically designed for categorical and continuous variables and initially proposed by Ru *et al.* in the COCABO method [14].

We also experimentally compare different methods to optimize the acquisition function. We measure their performances in the optimization of multiple chemical reaction (or formulation) simulators.

We find that a brute-force approach for the optimization of the acquisition function offers the best results but is too slow when there are many categorical variables or categories. However we show that an ant colony optimization technique for the optimization of the acquisition function is a well-suited alternative when the brute-force approach cannot be (reasonably) used.

We show that the proposed Bayesian optimization algorithm finds optimal reaction parameters in fewer experiments than state of the art algorithms on our simulators.

Keywords: Mixed Bayesian optimization · chemical reaction optimization · categorical variables.

1 Introduction

Every chemical reaction is optimized before being industrialized. The goal is to find, by carrying out experiments, input parameters (e.g. temperature, pressure,

residence time, etc.) that offer optimal values for a set of objectives (e.g maximize the yield, minimize the production of an impurity, etc.).

The pursuit of high-performance optimization methods is driven by the high cost of chemical experiments. The performances of optimization methods applied to chemical reactions are measured against the quality of the solution (i.e. how close the solution is to the optimization objectives) and how many experiments are needed to find this solution.

One-Variable-At-a-Time (OVAT) and Design of Experiments (DoE) [1, 18] methods are the most used approaches to optimize chemical reactions. The OVAT method iterates by performing experiments and modifying only one parameter at a time. DoE methods consist in planning a series of experiments following a design matrix, running these experiments and building a statistical model (usually linear or polynomial) with the resulting dataset. An optimum is then computed from the model. OVAT and DoE methods tend to need a large number of experiments to be effective. In addition, OVAT can be very slow (because only one variable is changed at a time) and can get stuck in local optima. Simplex-based methods are also sometimes used to optimize chemical reactions [11, 21]. They consists of building a simplex in the search space, then evaluating the objective function at each of the vertices of the simplex and iteratively displacing one vertex at a time following heuristics. Simplex-based methods tend to be easily stuck in local optima [20].

Zhou *et al.* [23] proposed a deep reinforcement learning (DRL) based method to optimize chemical reactions. The authors combined DRL and pre-training to be able to start working with very small amounts of data. This leads to satisfactory results on problems containing only continuous variables but hasn't been tested with categorical variables (without descriptors).

Bayesian optimization (BO) is a powerful approach to optimize problems for which the evaluations are expensive and noisy. It has shown a variety of successful applications [15]. BO concepts are described in Figure 1. First, an initialisation is done with a small number of experiments. Then, a surrogate model (e.g. Gaussian process) is trained using these experiments. An acquisition function, that balances the predicted improvement (exploitation strategy) and the uncertainty of the predictions (exploration strategy), is applied to the model. An optimization algorithm is applied to find the maximum of this acquisition function. The set of parameters that gives this maximal value for the acquisition function determines the next experiment (chemical reaction) to run. This experiment is run, its result is added to the dataset, and the algorithm starts a new iteration. The algorithm stops when the objectives are attained or when the experiments budget is spent.

Categorical variables are often present in the optimization of chemical reactions [13]. We can cite as an example the choice of a catalyst or additives, the choice of the solvent or the order of addition of the reactants. Categorical variables have two important particularities. The first one is the non-continuity constraint, since categorical variables are not defined on a continuous space. The second one is the non-ordinality constraint: they can only be compared with the

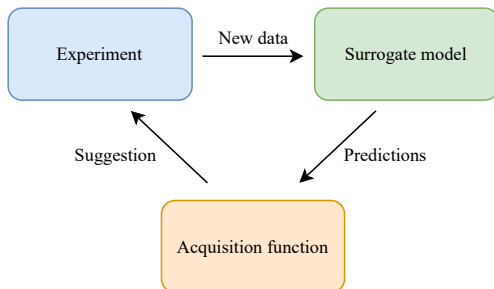


Fig. 1: Simplified Bayesian optimization algorithm applied to chemical reactions.

equality operator. For example, with a categorical variable representing a choice between the solvents *water*, *ethanol*, *toluene*, asserting that *water* > *toluene* is meaningless.

Mixed-variable optimization can be handled with one-hot encoding: a categorical variable with n categories is encoded as a vector of n corresponding bits, with all bits being equal to 0 except the bit corresponding to the selected category, that is equal to 1. However, in the BO algorithm, treating one-hot dimensions as continuous without any supplementary treatment misleads the acquisition function optimizer and often results in a sub-optimal solution. Indeed, the experiment proposed by the acquisition function optimizer is a real-valued vector and has to be decoded to the closest category. Hence, most of the time, there will be a gap between the experiment suggested by the acquisition function optimizer and the experiment that will actually be performed, leading to a mediocre optimization performance.

The work presented by Garrido-Merchán *et al.* [4] brings an improvement to the basic one-hot encoding approach. During the optimization of the acquisition function, real-valued encoded vectors are transformed to the nearest one-hot vectors before being used as inputs of the model. It follows that the acquisition function optimizer considers real-valued vectors as having the same acquisition values as the associated transformed vectors. Thus, the acquisition optimizer suggests an experiment that can be performed as is, which ensures the convergence to optimal solutions.

Häse *et al.* [5] have developed an augmented Bayesian optimization algorithm called Gryffin that uses a Bayesian neural network as surrogate model. It estimates kernel densities, based on previously evaluated experiments, that are used to approximate the objective function. Gryffin is able to use expert knowledge (descriptors) to guide the optimization, which drastically improves the performances of their method. Its "naive" version doesn't use descriptors, which enabled us to use it in our benchmarks.

COCABO [14] is a Bayesian optimization method designed for mixed-variable optimization. At each iteration, COCABO first selects categories with a multi-

armed bandit algorithm and then separately optimizes the numerical variables (after modelling them using a mixed covariance function).

Random forests can be used as surrogate model in Bayesian optimization instead of Gaussian processes [6]. A ready-to-use implementation of this approach is provided in a package called SMAC [8] which handles categorical variables.

In this study, we aim at improving the performances (i.e. reducing the number of experiments necessary to reach an optimum) of the Bayesian optimization method for the optimization of chemical reactions with continuous and categorical variables. Our approach is based on Gaussian processes as surrogate models with the COCABO covariance function [14]. We propose different techniques for the optimization of the acquisition function. Next, we compare the different acquisition function optimizer on the optimization of simulated chemical reactions. And finally, we compare our optimization algorithm (using the COCABO covariance function and the highest-performing acquisition function optimizer) with other state-of-the-art algorithms.

2 Problem definition

Our work is applied to problems with a form given by:

$$\text{Minimize } f(\mathbf{z}) \text{ with the smallest possible number of evaluations} \quad (1)$$

where :

- $\mathbf{z} = (\mathbf{x}, \mathbf{h})$
- $\mathbf{x} = x_0, \dots, x_n$ and $x_i \in [A_i, B_i]$ with $A_i, B_i \in \mathbb{R}$
- $\mathbf{h} = h_0, \dots, h_n$ and $h_i \in C_i$ with C_i denotes the categorical space of the i th categorical variable.

This work is restricted to single objective optimization. Moreover, only continuous and categorical variables are used.

The "No-Free Lunch Theorem" [19] stipulates that the performances of every optimization methods are equal when averaged on all possible problems. It implies that in order to increase the performances on a specific optimization problem (e.g. chemical reaction optimization), we must evaluate the optimization method on similar problems without any regards on the performances of unrelated ones. The underlying functions of chemical reactions have some particularities: they are smooth and have few local optima [10, 17]. So, in order to be specific to the chemical reaction optimization problem, we measure the performances of our approach using chemical reaction (or formulation) simulators. We have built these simulations by training machine learning models with publicly available chemical data (see Table 1). Each data set has been harvested from patents or academic articles and are produced by the optimization of chemical reactions (or chemical formulations). The individuals of these data sets corresponds to experiments where input parameters (e.g. temperature, choice of catalyst) have been tested and scores have been calculated following

the observations made at the end of these experiments. In the case of the first Suzuki-Miyaura reaction simulation (second row of the Table 1), there are two objectives (turnover number and yield). We choose to optimize the mean of the two scores.

This benchmarking strategy was initially introduced by Felton *et al.* [3] for measuring performances on chemical reactions with continuous and categorical variables. It allows us to establish optimization performances on chemical reactions without having to run experiments in a chemistry lab.

Table 1: Details of the data used to train the simulators

Reaction type	Number of experiments	Source
Pd-catalysed direct arylation	1728	[16]
Suzuki-Miyaura cross-coupling	4 cases of 96	[13, 3]
Stereoselective Suzuki-Miyaura cross-coupling	192	[2]
Polycarbonate resin formulation	100	[9]

3 Propositions

In a first part, we describe the surrogate model including the COCABO kernel and its hyperparameters. In a second part, we present different approaches for the optimization of the acquisition function.

3.1 Gaussian process kernel

We use Gaussian processes (GP) to approximate the underlying functions of chemical reactions. It is the most commonly used model since it can inherently predict both a value and an associated uncertainty. Gaussian processes are mainly defined by their covariance function. Since the underlying functions of chemical reactions are smooth, we use a smooth covariance function, Matérn_{5/2} [12], for the continuous dimensions.

The smoothness of the GP on continuous variables is kept with the use of the one-hot encoding. However, the Euclidean distance used for the calculation of the Matérn_{5/2} kernel is based on all dimensions (continuous and encoded). We believe that, in order to capture complex relationships between categorical and continuous variables, the covariance function should use the Euclidean distance only on continuous variables and incorporate categorical knowledge later in its calculation. The COCABO method [14] uses such a covariance function (see equation 2). It combines two sub-functions: one for continuous variables, K_{cont} , and one for categorical variables, K_{cat} .

$$K(\mathbf{z}, \mathbf{z}') = (1 - \lambda) \times (K_{cont}(\mathbf{x}, \mathbf{x}') \times K_{cat}(\mathbf{h}, \mathbf{h}')) + \lambda \times (K_{cont}(\mathbf{x}, \mathbf{x}') + K_{cat}(\mathbf{h}, \mathbf{h}')) \quad (2)$$

where :

- $\mathbf{z} = (\mathbf{x}, \mathbf{h})$
- \mathbf{x} is the set of continuous variables
- \mathbf{h} is the set of categorical variables

K_{cont} is the Matérn_{5/2} function. It is a standard covariance function for smooth Gaussian processes regressions with continuous inputs. K_{cat} , the kernel for categorical inputs (see equation 3), measures similarity between categorical vectors with the equality operator (which is the only permitted operation for categorical variables).

$$K_{cat}(\mathbf{h}, \mathbf{h}') = \sigma \times \frac{1}{D} \sum_1^D \alpha(h_d, h'_d) \quad (3)$$

where:

- $\alpha(a, b)$ equals 1 if $a = b$ and 0 if $a \neq b$
- D is the number of categorical variables
- σ is the variance hyperparameter.

The proposition made by Ru *et al.* in COCABO [14] revolves around the hyperparameter λ , which is a trade-off between the two terms of the equation 2: the sum and the product of K_{cont} and K_{cat} . Both of these terms capture different relationships between continuous and categorical variables. The sum of the two sub-kernels produces a learning of a single trend on the continuous variables and shift this trend depending on the categories whereas the product is able to produce a learning of complex relationships with highly different trends depending on the categories. The sum is especially necessary when the amount of training data is low (beginning of the optimization) because the product is able to capture knowledge only if the evaluations have categories in common. For example, if two evaluations have the same continuous features but different categorical ones, the product will be equal to 0 which prevent the model to learn even on continuous variables. Nonetheless, the product is essential because, as the optimization goes on, more evaluations are added to the training dataset and a single trend with a simple shift will not be sufficient to model the complexity offered by the data. In other words the sum alone will not be able to capture all the knowledge available to guide the optimization. With the hyperparameter λ , the authors ensure that the relationships that can be captured either by the sum or by the product are taken into account into the covariance $K(\mathbf{z}, \mathbf{z}')$, because λ is tuned during the fitting of the Gaussian process.

In order to avoid underfitting/overfitting the data while training the Gaussian process (tuning its hyperparameters to minimize its negative log marginal likelihood [12]), we confined hyperparameter values within a range. $\sigma_K, \sigma_{K_{cont}}$

and $\sigma_{K_{cat}}$ were bounded in $[10^{-2}, 20]$ while the lengthscale parameter of K_{cont} and λ were respectively bounded in $[10^{-2}, 20]$ and $[0.1, 0.9]$. We used the L-BFGS optimizer to tune the GP hyperparameters.

3.2 Acquisition function optimization

We chose to use the Expected Improvement (EI) acquisition function because it has shown good results on diverse applications and has a strong theoretical support [22]. The equation of Expected Improvement is given by:

$$\text{EI}(\mathbf{x}) = \mathbb{E}[\max(f(\mathbf{x}) - f(\mathbf{x}^+), 0)] \quad (4)$$

with $f(\mathbf{x}^+)$ the value of the evaluation that have yielded the best result so far. The analytical form of EI is the following:

$$\text{EI}(\mathbf{x}) = \begin{cases} \sigma(\mathbf{x})Z\Phi(Z) + \sigma(\mathbf{x})\phi(Z) & \text{if } \sigma(\mathbf{x}) \neq 0 \\ 0 & \text{if } \sigma(\mathbf{x}) = 0 \end{cases} \quad (5)$$

where

$$Z = \frac{\mu(\mathbf{x}) - f(\mathbf{x}^+) - \xi}{\sigma(\mathbf{x})} \quad (6)$$

$\Phi(Z)$ and $\phi(Z)$ denotes respectively the cumulative distribution function (CDF) and the probability density function (PDF) of the variable Z . Z denotes the predicted improvement divided by the standard deviation (uncertainty) and the parameter ξ determines the weight of the exploration strategy in the equation. This analytical form of EI is cheap to evaluate and can be optimized without sparing on the number of evaluations. Therefore, we propose several approaches for the optimization of the acquisition function with mixed variables.

The first approach (denoted as L-BFGS-OHE) involves the one-hot encoding of the categorical variables and a multi-started gradient descent for the optimization of the acquisition function. However, since the COCABO model does not accept one-hot vectors, one-hot dimensions are systematically decoded before any predictions. In other words, predictions are asked for by the acquisition function optimizer with encoded inputs but they are decoded before they pass through the model. The multi-started gradient descent is performed as follows: 1000 configurations are randomly drawn and the 5 configurations with the highest acquisition function value are kept and a gradient descent (L-BFGS) is performed on each of these 5 configurations.

We also propose an approach based on a "brute-force" optimization of the categorical space and a multi-started gradient descent on the continuous space (see Algorithm 1). First, all the combinations of the categorical parameters are constructed. Then, for each combination, a multi-started gradient descent (previously described) is performed on the continuous parameters. Finally, after determining the maximal acquisition values for each categorical combination, the configuration with the highest acquisition value is suggested as the next experiment. This algorithm reduces the difficulty of the optimization of the acquisition

function because instead of dealing with different types of variables (or with supplementary dimensions from the encoding), the acquisition optimizer only works on the continuous dimensions. Still, it can be heavy in terms of computational cost if the number of categories and categorical variables is large.

Algorithm 1 Categorical brute-force and multi-started gradient descent

- 1: Construct all categorical combinations
 - 2: Multi-started gradient descent optimization of continuous parameters for each combination
 - 3: Choose as suggestion the configuration (continuous and categorical) with the highest acquisition
-

While the brute-force approach evaluates thoroughly the search space, with five optimizations of the continuous space for each categorical combination, this method becomes prohibitively long to run when the number of categorical combinations is higher than a few hundreds.

Lastly, we implemented an evolutionary algorithm based the behaviour of ant colonies (ACO) that scales better with the number of categorical variables than brute-force [7]. In our experiments, we used the colony hyperparameters proposed by the authors without any restart allowed. This algorithm is a multi-agent method inspired by the behaviour of ants. An ant represent an evaluation at a given set of parameters. At each generation, each ant randomly moves towards previously evaluated points with good results (exploitation strategy). The presence of multiple ants in the colony and the randomness of their movements enable the mandatory exploration of the search space. It allows the ants to not only moves around promising areas but also randomly explore areas that may have not been explored so far.

4 Results

This section presents the optimization of four simulators. We did 25 runs with 50 suggestions for each optimization algorithm. At the beginning of each run, we randomly drew 5 initial evaluations and, for a fairness purpose, these 5 evaluations were used to initialize all the optimizers. As shown in the Table 2, the number of input variables (16) of the polycarbonate resin formulation is higher than the other simulations (less than 6), so we allowed 100 suggestions for each optimizer on this simulation.

Table 2: Number of variables for each simulations

Reaction type	cont. var.	cat. var.
Pd-catalysed direct arylation	2	3
Suzuki-Miyaura cross-coupling	3	1
Stereoselective Suzuki-Miyaura cross-coupling	5	1
Polycarbonate resin formulation	11	5

4.1 Acquisition function optimizer

The four plots presented in Figure 2 are used to compare the performances of different acquisition optimizers, using the COCABO kernel and the Expected Improvement acquisition function. The average (and standard deviation) of the best score obtained at the end of each optimization are summarized in Table 3. On each benchmark we provide, as a baseline, the results of a random strategy that suggests input parameters randomly.

On the first three simulators (Figures 2a, 2b and 2c), brute-force outperforms the other methods. On the last simulator (see Figure 2d), the brute-force approach cannot be used due to the high number of categorical variables and categories. Indeed, there are 1728 combinations of categorical variables, so the brute-force approach consists of 8640 optimization routines upon continuous variables, which is too long to run.

The approach based on categorical relaxation (L-BFGS-OHE) with the one-hot encoding and a rounding to the closest categorical variable before passing through the model, is the second best performing method on every simulators. Even if the optimizer (L-BFGS) evolves in a continuous space with a large number of encoded dimensions and a large number of flat regions (depending on the number of categorical variables and categories), the optimizer manages to optimize the acquisition function.

The behaviour of the ant colony optimization method (ACO) highly depends on its parameters, notably the exploration hyperparameter q which is set to 0.05 and the *restart* parameter. The ACO performs poorly, compared to the other optimization methods, on the simulators with a small number of dimensions but performs slightly better on the polycarbonate resin formulation simulator which have more dimensions.

As consequence of the results presented above and for the rest of our study, we choose the brute-force approach to be the acquisition function optimizer when the number of categorical variables is lower than 4. ACO is used otherwise.

4.2 Comparison with other methods

The next results (Figure 3) present a comparison between our method (composed of the COCABO kernel, Expected Improvement and the brute-force optimizer

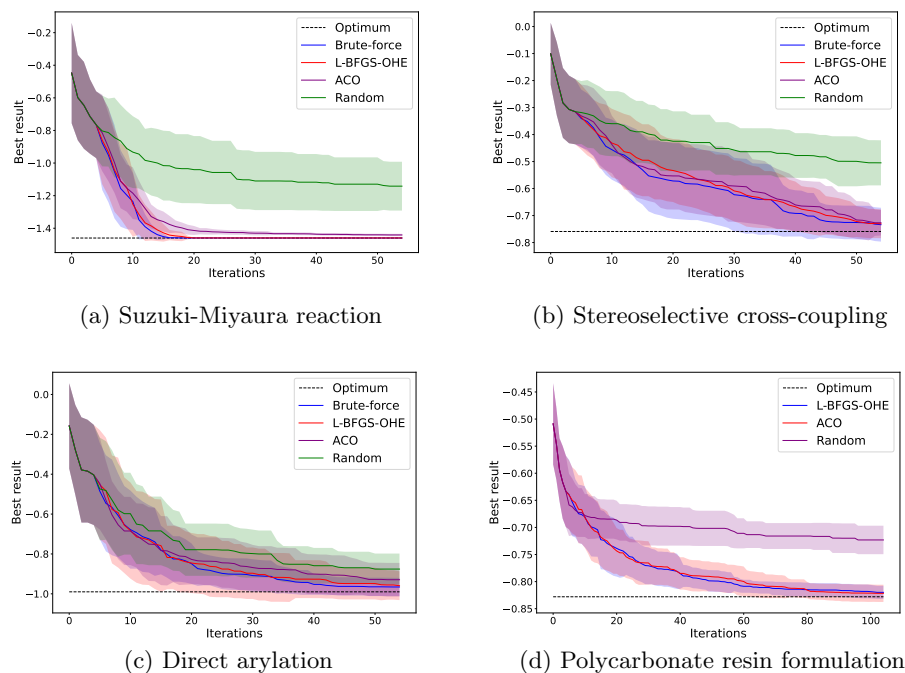


Fig. 2: Best score evolution on simulations with the use of different acquisition function optimizers (Brute-force, L-BFGS-OHE, ACO, Random).

Method	Suzuki-Miyaura	Stereoselective Cross coupling	Direct arylation	Polycarbonate resin formulation
Brute-force	-1.460 \pm 0.00	-0.733 \pm 0.06	-0.965 \pm 0.05	–
L-BFGS-OHE	-1.460 \pm 0.00	-0.730 \pm 0.05	-0.959 \pm 0.07	-0.820 \pm 0.01
ACO	-1.441 \pm 0.01	-0.726 \pm 0.04	-0.929 \pm 0.08	-0.821 \pm 0.02
Random	-1.141 \pm 0.15	-0.504 \pm 0.08	-0.875 \pm 0.08	-0.723 \pm 0.03

Table 3: Average scores for different acquisition optimizers at 5 random initial evaluations and 50 suggestions (100 for the polycarbonate resin formulation). Best scores on each simulators are marked in bold.

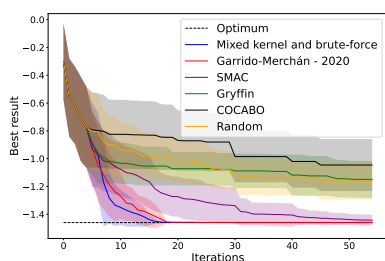
or the ant colony optimizer when the brute-force method can't be used), Gryffin [5], COCABO [14], SMAC [6], and the work of Garrido-Merchán *et al.* [4].

We used the "naive" version of Gryffin in its authors' implementation, which does not use any chemical descriptor to guide the optimization. The computation time of Gryffin is long when the number of dimensions is high, so we used the "boosted" version of Gryffin for the optimization of the polycarbonate resin formulation simulation.

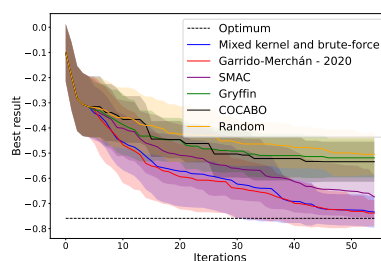
We used COCABO in its authors' implementation with its default settings and a starting $\lambda = 0.5$.

SMAC denotes an optimization algorithm based on Random Forest [6] and the Expected Improvement acquisition function. We used an implementation proposed by Lindauer *et al.* [8].

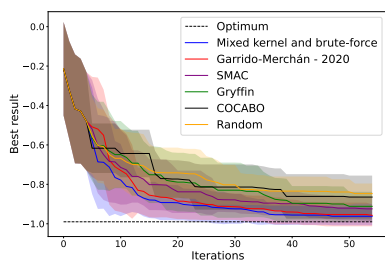
"Garrido-Merchán - 2020" is a Bayesian optimization method which involves a Matérn_{5/2} kernel and the one-hot encoding of the categorical variables.



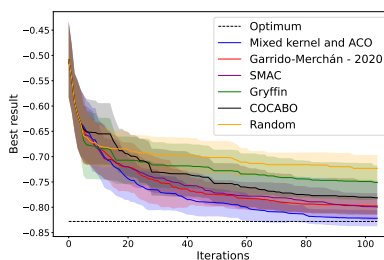
(a) Suzuki-Miyaura reaction



(b) Stereoselective cross-coupling



(c) Direct arylation



(d) Polycarbonate resin formulation

Fig. 3: Best score evolution on simulators with the use of different optimization methods: Mixed kernel and brute-force (or ACO), Garrido-Merchán - 2020, SMAC, Gryffin, COCABO, Random

Method	Suzuki-Miyaura	Stereoselective	Direct arylation	Polycarbonate resin formulation
		Cross coupling		
Mixed kernel and brute-force	-1.460 \pm 0.01	-0.733 \pm 0.06	-0.965 \pm 0.05	–
Mixed kernel and ACO	-1.441 \pm 0.01	-0.726 \pm 0.04	-0.929 \pm 0.08	-0.821 \pm 0.02
Garrido-Merchán - 2020	-1.459 \pm 0.01	-0.738 \pm 0.05	-0.957 \pm 0.06	-0.797 \pm 0.02
SMAC	-1.443 \pm 0.04	-0.674 \pm 0.09	-0.924 \pm 0.06	-0.799 \pm 0.03
Gryffin	-1.149 \pm 0.13	-0.517 \pm 0.08	-0.912 \pm 0.07	-0.751 \pm 0.03
COCABO	-1.046 \pm 0.18	0.534 \pm 0.08	-0.865 \pm 0.11	-0.781 \pm 0.03
Random	-1.160 \pm 0.13	0.504 \pm 0.08	-0.847 \pm 0.05	-0.723 \pm 0.03

Table 4: Average scores for different optimization methods at 5 random initial evaluations and 50 suggestions (100 for the polycarbonate resin formulation). Best scores on each simulators are marked in bold.

The results on the simulator denoted as "Suzuki-Miyaura" (which corresponds to a simulation built upon the first data set proposed by Reizman *et al.* [13]), shows that the Mixed kernel combined with a categorical brute-force approach for the optimization of the acquisition function is the method that offers the best performances.

The method denoted as "Garrido-Merchán - 2020" is the method that provides the best performances on the second simulator ("stereoselective cross-coupling").

On the Pd-catalysed direct arylation simulator, the COCABO kernel combined with the brute-force approach gives the best results. Gryffin and COCABO do not provide satisfactory scores on the two first simulators ("Suzuki-Miyaura", "stereoselective cross-coupling"). However these optimization methods provide better results on the other two simulators.

The method composed of the COCABO kernel and the ant colony optimization technique gives the best averaged final score on the polycarbonate resin formulation simulator (-0.821).

SMAC performs poorly on the first three simulators compared to Bayesian optimization with Gaussian process as surrogate model. However on the last simulator, the random forest based Bayesian optimization performances are similar to the Bayesian optimization with one-hot encoding and the Matérn_{5/2} kernel.

We show that our method composed of the COCABO kernel and either the categorical brute-force or the ant colony optimizer as the acquisition function optimization technique, generally converges faster to the optimum than the other methods.

4.3 Kernel influence on performances

The Table 5 summarises the averaged final scores of two Bayesian optimization methods that only differ from each other by their covariance functions and the use of the one-hot encoding. "Garrido-Merchán-2020" is using the one-hot encoding and a standard continuous kernel (Matérn_{5/2}) whereas the method denoted as "L-BFGS-OHE" is using a kernel specially designed for continuous

and categorical variables without encoding (the COCABO kernel). Their results are similar on the first simulator but on the second one ("stereoselective cross coupling"), the method denoted as "Garrido-Merchán-2020" achieves better performances (-0.738) than "L-BFGS-OHE" method (-0.730). On the polycarbonate resin formulation simulator, the COCABO kernel based method ("L-BFGS-OHE") performs better (-0.820) than "Garrido-Merchán-2020" (-0.797). This comparison exposes that the COCABO kernel should be used instead of a Matérn_{5/2} kernel and the one-hot encoding when the chemical reaction (or formulation) optimization problem involves multiple continuous and categorical variables. Indeed, this covariance function allows a more efficient Bayesian optimization on our simulators. The authors of COCABO (Ru *et al.*) attribute this phenomenon to a stronger modeling power of the COCABO kernel.

Method	Suzuki-Miyaura	Stereoselective Cross coupling	Direct arylation	Polycarbonate resin formulation
L-BFGS-OHE	-1.460 ± 0.00	-0.730 ± 0.05	-0.959 ± 0.07	-0.820 ± 0.01
Garrido-Merchán - 2020	-1.459 ± 0.01	-0.738 ± 0.05	-0.957 ± 0.06	-0.797 ± 0.02

Table 5: Average final scores of "Garrido-Merchán-2020" and "L-BFGS-OHE" extracted from Table 3 and Table 4.

5 Conclusion

This paper presents a study of different optimization techniques for the optimization of chemical reaction (or formulation) simulators with mixed variables (continuous and categorical).

We expose a Bayesian optimization algorithm based on a Gaussian process with a covariance function specifically designed for continuous and categorical variables [14]. Also, we evaluate different methods for the optimization of the acquisition function. We show that when facing a small number of categorical variables, a categorical brute-force approach associated with a multi-started gradient descent performs best for the optimization of the acquisition function. The ant colony optimization method is the method the most suited (in our study) for the optimization of the acquisition function when the number of categorical combinations is too high to use the brute-force based approach. A more in-depth study of these evolutionary methods will be the subject of further works. Our method globally performs better than other state-of-the-art methods [5, 4, 14, 8] on our simulators.

Since the use of a kernel specifically designed for continuous and categorical inputs shows the best performances, we are working on further increasing the quality of the model by modifying the COCABO covariance function. We believe that more relationships between variables and evaluations can be captured with a new set of parameters in the covariance function.

Nevertheless, in order to fully establish the performance of the presented method, we are currently working on an experimental validation in chemistry labs.

Acknowledgement: This work was supported by the R&D Booster SMAPI project 2020 of the Auvergne-Rhône-Alpes Region.

References

1. Carlson, R., Carlson, J.E.: Design and optimization in organic synthesis. Elsevier (2005)
2. Christensen, M., Yunker, L.P., Adedeji, F., Häse, F., Roch, L.M., Gensch, T., dos Passos Gomes, G., Zepel, T., Sigman, M.S., Aspuru-Guzik, A., et al.: Data-science driven autonomous process optimization. *Communications Chemistry* **4**(1), 1–12 (2021)
3. Felton, K., Rittig, J., Lapkin, A.: Summit: Benchmarking machine learning methods for reaction optimisation (2020)
4. Garrido-Merchán, E.C., Hernández-Lobato, D.: Dealing with categorical and integer-valued variables in bayesian optimization with gaussian processes. *Neurocomputing* **380**, 20–35 (2020)
5. Häse, F., Aldeghi, M., Hickman, R.J., Roch, L.M., Aspuru-Guzik, A.: Gryffin: An algorithm for bayesian optimization of categorical variables informed by expert knowledge. *Applied Physics Reviews* **8**(3), 031406 (2021)
6. Hutter, F., Hoos, H.H., Leyton-Brown, K.: Sequential model-based optimization for general algorithm configuration. In: International conference on learning and intelligent optimization. pp. 507–523. Springer (2011)
7. Liao, T., Socha, K., de Oca, M.A.M., Stützle, T., Dorigo, M.: Ant colony optimization for mixed-variable optimization problems. *IEEE Transactions on Evolutionary Computation* **18**(4), 503–518 (2013)
8. Lindauer, M., Eggenesperger, K., Feurer, M., Biedenkapp, A., Deng, D., Benjamins, C., Ruhkopf, T., Sass, R., Hutter, F.: Smac3: A versatile bayesian optimization package for hyperparameter optimization. In: ArXiv: 2109.09831 (2021), <https://arxiv.org/abs/2109.09831>
9. Monden, T.: Polycarbonate resin composition (2019), EP2810989B1
10. Moore, K.W., Pechen, A., Feng, X.J., Dominy, J., Beltrani, V.J., Rabitz, H.: Why is chemical synthesis and property optimization easier than expected? *Physical Chemistry Chemical Physics* **13**(21), 10048–10070 (2011)
11. Morgan, S.L., Deming, S.N.: Simplex optimization of analytical chemical methods. *Analytical chemistry* **46**(9), 1170–1181 (1974)
12. Rasmussen, C.E., Nickisch, H.: Gaussian processes for machine learning (gpml) toolbox. *The Journal of Machine Learning Research* **11**, 3011–3015 (2010)
13. Reizman, B.J., Wang, Y.M., Buchwald, S.L., Jensen, K.F.: Suzuki–miyaura cross-coupling optimization enabled by automated feedback. *Reaction chemistry & engineering* **1**(6), 658–666 (2016)
14. Ru, B., Alvi, A., Nguyen, V., Osborne, M.A., Roberts, S.: Bayesian optimisation over multiple continuous and categorical inputs. In: International Conference on Machine Learning. pp. 8276–8285. PMLR (2020)

15. Shahriari, B., Swersky, K., Wang, Z., Adams, R.P., De Freitas, N.: Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE* **104**(1), 148–175 (2015)
16. Shields, B.J., Stevens, J., Li, J., Parasram, M., Damani, F., Alvarado, J.I.M., Janey, J.M., Adams, R.P., Doyle, A.G.: Bayesian reaction optimization as a tool for chemical synthesis. *Nature* **590**(7844), 89–96 (2021)
17. Tibbetts, K.M., Feng, X.J., Rabitz, H.: Exploring experimental fitness landscapes for chemical synthesis and property optimization. *Physical Chemistry Chemical Physics* **19**(6), 4266–4287 (2017)
18. Weissman, S.A., Anderson, N.G.: Design of experiments (doe) and process optimization. a review of recent publications. *Organic Process Research & Development* **19**(11), 1605–1633 (2015)
19. Wolpert, D.H., Macready, W.G., et al.: No free lunch theorems for search. Tech. rep., Technical Report SFI-TR-95-02-010, Santa Fe Institute (1995)
20. Wright, M.H., et al.: Nelder, mead, and the other simplex method. *Documenta Mathematica* **7**, 271–276 (2010)
21. Xiong, Q., Jutan, A.: Continuous optimization using a dynamic simplex method. *Chemical Engineering Science* **58**(16), 3817–3828 (2003)
22. Zhan, D., Xing, H.: Expected improvement for expensive optimization: a review. *Journal of Global Optimization* **78**(3), 507–544 (2020)
23. Zhou, Z., Li, X., Zare, R.N.: Optimizing chemical reactions with deep reinforcement learning. *ACS central science* **3**(12), 1337–1344 (2017)